aws

User Guide

AWS Systems Manager

AWS Systems Manager: User Guide

Table of Contents

Was ist AWS Systems Manager? 1
Funktionsweise 1
Funktionen 2
Application Management 2
Änderungsmanagement 3
Node Management 4
Verfahrensmanagement 7
Quick Setup 8
Gemeinsam genutzte -Ressourcen 9
Zugriff auf Systems Manager 9
Systems Manager Service-Namengeschichte 10
Unterstützt AWS-Regionen 11
Unterstützte Betriebssysteme und Maschinentypen 11
Unterstützte Betriebssysteme für Systems Manager 11
Unterstützte Maschinentypen in Hybrid- und Multi-Cloud-Umgebungen 18
Mit AWS SDKs arbeiten 18
Systems Manager einrichten 20
Systems Manager mit EC2-Instances verwenden 20
Konfigurieren Sie die für Systems Manager erforderlichen Instanzberechtigungen 21
Verbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager 33
Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen 39
Erstellen Sie die für Systems Manager in Hybrid- und Multicloud-Umgebungen erforderliche IAM-Servicerolle 42
Erstellen Sie eine Hybridaktivierung, um Knoten bei Systems Manager zu registrieren 50
So installieren Sie das SSM Agent auf Hybrid-Linux-Knoten 57
So installieren Sie den SSM Agent auf Windows Hybridknoten 66
Verwaltung von Edge-Geräten mit Systems Manager 71
Erstellen Sie eine IAM-Servicerolle für Ihre Edge-Geräte 72
Konfigurieren Sie Ihre Edge-Geräte für AWS IoT Greengrass 79
Aktualisieren Sie die AWS IoT Greengrass Token-Exchange-Rolle und installieren Sie sie SSM Agent auf Ihren Edge-Geräten 79
Einen AWS Organizations delegierten Administrator für Systems Manager erstellen 80
Verwenden eines delegierten Administrators mit Change Manager 80

Verwenden Sie einen delegierten Administrator mit Explorer 81
Verwenden Sie einen delegierten Administrator mit OpsCenter 81
Allgemeine Einrichtung 82
Melden Sie sich an für ein AWS-Konto 82
Erstellen Sie einen Benutzer mit Administratorzugriff 82
Führen Sie eine Verwaltungsaufgabe mit Systems Manager aus 85
Voraussetzungen 85
Starten einer Instance mit einer AMI mit vorinstalliertem SSM Agent 85
Stellen Sie mithilfe von Systems Manager eine Connect zu Ihrer verwalteten Instanz her 87
Bereinigen Ihrer Instance 87
Arbeiten mit SSM Agent 88
Erfahren Sie technische Details über die SSM Agent 88
Verhalten der Anmeldeinformationen von SSM Agent-Version 3.2.x.x 89
Priorität der SSM Agent-Anmeldeinformationen 89
Über das lokale ssm-user-Konto 91
SSM Agent und die Instance Metadata Service (IMDS) 92
Behalten SSM Agent up-to-date 92
Sicherstellen, dass das SSM Agent-Installationsverzeichnis nicht geändert, verschoben oder gelöscht wird 93
SSM Agentfortlaufende Updates von AWS-Regionen 93
SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets 94
Finden Sie AMIs mit dem SSM Agent vorinstallierten 103
Arbeiten mit SSM Agent auf EC2-Instances für Linux 109
Arbeiten mit SSM Agent auf EC2-Instances für macOS 185
Arbeiten mit SSM Agent auf EC2-Instances für Windows Server 188
Prüfen des SSM Agent-Status und Starten des Agenten 196
Überprüfen der SSM Agent-Versionsnummer 198
Anzeigen von SSM Agent-Protokollen 203
Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent 206
Automatisieren von Updates für SSM Agent 207
Abonnieren von SSM Agent-Benachrichtigungen 211
Fehlerbehebung für SSM Agent 212
SSM Agent ist veraltet 212
Probleme mithilfe von SSM Agent-Protokolldateien beheben 213
Agent-Protokolldateien werden nicht gedreht (Windows) 213
Keine Verbindung mit SSM-Endpunkten möglich 214

Verwenden Sie `ssm-cli`, um die Verfügbarkeit von verwalteten Knoten zu überprüfen 215
Quick Setup 216
Was sind die Vorteile von Quick Setup? 216
An wen richtet sich Quick Setup? 217
Verfügbarkeit von Quick Setup in AWS-Regionen 217
Erste Schritte mit Quick Setup 218
Konfigurieren der Heimat- AWS-Region 218
IAM-Rollen und -Berechtigungen für das Quick Setup-Onboarding 219
Verwenden von Quick Setup 222
Konfigurationsdetails 223
Bearbeiten und Löschen Ihrer Konfiguration 224
Compliance von Konfigurationen 225
Unterstützte Quick Setup-Konfigurationstypen 225
Amazon-EC2-Host-Verwaltung 225
Standard-Host-Verwaltung für eine Organisation 233
AWS Config Configuration Recorder 234
AWS Config Bereitstellung von Konformitätspaketen 237
Patch Manager Patching-Konfiguration der Organisation 239
DevOpsGuru-Konfiguration 250
Distributor-Paket-Bereitstellung 253
Amazon-EC2-Instance-Resource-Scheduler 254
AWS Ressourcen Explorer Konfiguration 256
Fehlerbehebung von Quick Setup-Ergebnissen 258
Verfahrensmanagement 261
Incident Manager 261
Explorer 261
Über welche Features verfügt Explorer? 262
In welcher Verbindung steht Explorer mit OpsCenter? 264
Was ist OpsData? 264
Entstehen Kosten für die Verwendung von Explorer? 266
Erste Schritte 266
Verwenden von Explorer 284
Exportieren OpsData 294
Fehlerbehebung 299
OpsCenter 301
OpsCenter-Workflow 302

Einrichten von OpsCenter ... 302
Integrieren von OpsCenter in anderen AWS-Services ... 326
Geben Sie einen Namen für den Benutzer ein und klicken Sie dann auf OpsItems ... 335
Verwalten von OpsItems ... 357
Löschen Sie OpsItems ... 380
Beheben von OpsItem-Problemen ... 381
Anzeigen von OpsCenter-Zusammenfassungsberichten ... 385
Beheben von Problemen mit OpsCenter ... 386
CloudWatch Armaturenbrett ... 388
Application Management ... 2
Application Manager ... 390
Was sind die Vorteile der Nutzung von Application Manager? ... 392
Über welche Features verfügt Application Manager? ... 392
Entstehen Kosten für die Verwendung von Application Manager? ... 395
Was sind die Ressourcenkontingente für Application Manager? ... 395
Erste Schritte ... 395
Arbeiten mit Application Manager ... 411
AWS AppConfig ... 441
Parameter Store ... 441
Welche Vorteile bietet Parameter Store meiner Organisation? ... 442
An wen richtet sich Parameter Store? ... 442
Über welche Features verfügt Parameter Store? ... 443
Was ist ein Parameter? ... 445
Einrichten von Parameter Store ... 448
Arbeiten mit Parameter Store ... 479
Arbeiten mit öffentlichen Parametern ... 562
Walkthroughs für Parameter Store ... 592
Prüfen und Protokollieren von Parameter Store-Aktivitäten ... 604
Fehlerbehebung für Parameter Store ... 605
Änderungsmanagement ... 607
Change Manager ... 607
Funktionsweise von Change Manager ... 608
Welche Vorteile bietet Change Manager meinen Vorgängen? ... 610
An wen richtet sich Change Manager? ... 611
Was sind die Hauptfeatures von Change Manager? ... 611
Entstehen Kosten für die Verwendung von Change Manager? ... 613

Was sind die primären Komponenten von Change Manager? ... 613
Einrichten von Change Manager ... 616
Arbeiten mit Change Manager ... 643
Prüfen und Protokollieren von Change Manager-Aktivitäten ... 697
Fehlerbehebung für Change Manager ... 698
Automatisierung ... 699
Wie kann meine Organisation von Automation profitieren? ... 699
Wer sollte Automation nutzen? ... 702
Was ist eine Automatisierung? ... 702
Einrichten der Automatisierung ... 706
Ausführen von Automatisierungen ... 718
Planung von Automatisierungen ... 790
Referenz zu Automation-Aktionen ... 815
Erstellen Ihrer eigenen Runbooks ... 923
Referenz zu Automation-Runbooks ... 1111
Tutorials ... 1111
Grundlegendes zu Automatisierungsstatus ... 1173
Fehlerbehebung für Systems Manager Automation. ... 1176
Change Calendar ... 1181
An wen richtet sich Change Calendar? ... 1182
Vorteile von Change Calendar ... 1182
Einrichten von Change Calendar ... 1183
Arbeiten mit Change Calendar ... 1186
Hinzufügen von Change Calendar-Abhängigkeiten zu Automation-Runbooks ... 1200
Fehlerbehebung für Change Calendar ... 1201
Maintenance Windows ... 1202
Einrichten von Maintenance Windows ... 1205
Arbeiten mit Wartungsfenstern (Konsole) ... 1217
Maintenance Windows Tutorials (AWS CLI) ... 1236
Anleitungen zu Wartungsfenstern ... 1301
Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben ... 1326
Wartungsfenster-Optionen für Planung und aktive Zeiträume ... 1332
Wartungsfenster-Tasks ohne Ziele registrieren ... 1337
Fehlerbehebung bei Wartungsfenstern ... 1339
Knotenverwaltung ... 1345

Fleet Manager 1345
An wen richtet sich Fleet Manager? 1345
Welche Vorteile bietet Fleet Manager meiner Organisation? 1346
Über welche Features verfügt Fleet Manager? 1346
Erste Schritte mit Fleet Manager 1347
Arbeiten mit Fleet Manager 1354
Problembehandlung bei der Verfügbarkeit verwalteter Knoten 1418
-Compliance 1433
Erste Schritte mit Compliance 1435
Erstellen einer Ressource Data Sync für Compliance 1436
Arbeiten mit Compliance 1439
Löschen einer Ressource Data Sync für Compliance 1444
Beheben von Compliance-Problemen mithilfe von EventBridge 1444
Compliance-Walkthrough (AWS CLI) 1447
-Bestand 1452
Weitere Informationen über Inventory 1457
Einrichten von Inventory 1468
Konfigurieren der Bestandserfassung 1482
Arbeiten mit Bestandsdaten 1489
Arbeiten mit benutzerdefiniertem Bestand 1512
Anzeigen von Bestandsverlauf und Änderungsnachverfolgung 1528
Anhalten der Datenerfassung und Löschen von Bestandsdaten 1530
Anleitungen zu Inventory 1532
Beheben von Inventory-Problemen 1551
Hybride Aktivierungen 1556
Session Manager 1557
Welche Vorteile bietet Session Manager meiner Organisation? 1558
An wen richtet sich Session Manager? 1560
Was sind die Hauptfeatures von Session Manager? 1561
Was ist eine Sitzung? 1563
Einrichten von Session Manager 1564
Arbeiten mit Session Manager 1647
Prüfen von Sitzungsaktivitäten 1674
Protokollierung von Sitzungsaktivitäten aktivieren und deaktivieren 1675
Schema des Sitzungsdokuments 1683
Fehlerbehebung für Session Manager 1692

Run Command 1701
Einrichten von Run Command 1703
Ausführen von Befehlen auf verwalteten Knoten 1708
Verwendung von Beendigungscodes in Befehlen 1726
Grundlegendes zu Befehlsstatus 1729
Walkthroughs für Run Command 1742
Fehlerbehebung für Run Command 1771
State Manager 1772
Welche Vorteile bietet State Manager meiner Organisation? 1772
An wen richtet sich State Manager? 1773
Über welche Features verfügt State Manager? 1773
Entstehen Kosten für die Verwendung von State Manager? 1775
Erste Schritte mit State Manager 1775
Informationen zu State Manager 1776
Arbeiten mit Zuordnungen 1780
Walkthroughs für State Manager 1825
Patch Manager 1875
Verwenden von Quick Setup-Patch-Richtlinien 1879
Patch Manager-Voraussetzungen 1882
Funktionsweise 1889
Über SSM-Dokumente für das Patchen von verwalteten Knoten 1947
Über Patch-Baselines 2004
Verwenden von Kernel Live Patching auf von Amazon Linux 2 verwalteten Knoten 2027
Arbeiten mit Patch Manager (Konsole) 2036
Arbeiten mit Patch Manager (AWS CLI) 2111
Patch Manager-Anleitungen 2146
Fehlerbehebung für Patch Manager 2162
Distributor 2183
Welche Vorteile bietet Distributor meiner Organisation? 2183
An wen richtet sich Distributor? 2184
Über welche Features verfügt Distributor? 2184
Was ist ein Paket? 2186
Einrichten von Distributor 2188
Arbeiten mit Distributor 2191
Prüfen und Protokollieren von Distributor-Aktivitäten 2237
Fehlerbehebung für Distributor 2238

Freigegebene Ressourcen 2241
-Documents 2241
Wie kann meine Organisation von der Documents-Funktion profitieren? 2241
Wer sollte Documents verwenden? 2242
Welche Typen von SSM-Dokumenten gibt es? 2243
Dokument-Komponenten 2253
Erstellen von SSM-Dokumentinhalten 2344
Arbeiten mit Dokumenten 2350
Sicherheit 2383
Datenschutz 2384
Datenverschlüsselung 2385
Richtlinie für den Datenverkehr zwischen Netzwerken 2388
Identity and Access Management 2388
Zielgruppe 2388
Authentifizierung mit Identitäten 2389
Verwalten des Zugriffs mit Richtlinien 2393
Funktionsweise von AWS Systems Manager mit IAM 2395
Beispiele für identitätsbasierte Richtlinien 2407
AWS verwaltete Richtlinien 2419
Fehlerbehebung 2431
Verwenden von serviceverknüpften Rollen 2433
Inventar und Explorer-Datenrollen 2435
OpsCenter- und Explorer-Rolle der Kontoerkennung 2438
OpsData und OpsItems Erstellungsrolle 2441
Rolle für die Erstellung operativer Einblicke 2445
OpsData Servicerolle exportieren 2449
Protokollierung und Überwachung 2452
Compliance-Validierung 2455
Ausfallsicherheit 2456
Sicherheit der Infrastruktur 2456
Konfigurations- und Schwachstellenanalyse 2457
Bewährte Methoden für die Gewährleistung der Sicherheit 2457
Bewährte Methoden für vorbeugende Systems Manager-Sicherheitsmaßnahmen 2457
Bewährte Methoden zur Überwachung und Prüfung von Systems Manager 2462
Codebeispiele 2465
Aktionen 2470

AddTagsToResource 2473
CancelCommand 2475
CreateActivation 2477
CreateAssociation 2478
CreateAssociationBatch 2483
CreateDocument 2486
CreateMaintenanceWindow 2490
CreateOpsItem 2494
CreatePatchBaseline 2496
DeleteActivation 2500
DeleteAssociation 2501
DeleteDocument 2503
DeleteMaintenanceWindow 2504
DeleteParameter 2506
DeletePatchBaseline 2507
DeregisterManagedInstance 2509
DeregisterPatchBaselineForPatchGroup 2510
DeregisterTargetFromMaintenanceWindow 2511
DeregisterTaskFromMaintenanceWindow 2513
DescribeActivations 2514
DescribeAssociation 2516
DescribeAssociationExecutionTargets 2519
DescribeAssociationExecutions 2522
DescribeAutomationExecutions 2525
DescribeAutomationStepExecutions 2527
DescribeAvailablePatches 2530
DescribeDocument 2534
DescribeDocumentPermission 2536
DescribeEffectiveInstanceAssociations 2538
DescribeEffectivePatchesForPatchBaseline 2541
DescribeInstanceAssociationsStatus 2544
DescribeInstanceInformation 2546
DescribeInstancePatchStates 2552
DescribeInstancePatchStatesForPatchGroup 2554
DescribeInstancePatches 2558
DescribeMaintenanceWindowExecutionTaskInvocations 2561

DescribeMaintenanceWindowExecutionTasks 2563
DescribeMaintenanceWindowExecutions 2564
DescribeMaintenanceWindowTargets 2568
DescribeMaintenanceWindowTasks 2571
DescribeMaintenanceWindows 2576
DescribeOpsItems 2579
DescribeParameters 2582
DescribePatchBaselines 2587
DescribePatchGroupState 2591
DescribePatchGroups 2592
GetAutomationExecution 2594
GetCommandInvocation 2597
GetConnectionStatus 2600
GetDefaultPatchBaseline 2601
GetDeployablePatchSnapshotForInstance 2602
GetDocument 2605
GetInventory 2607
GetInventorySchema 2609
GetMaintenanceWindow 2611
GetMaintenanceWindowExecution 2613
GetMaintenanceWindowExecutionTask 2614
GetParameterHistory 2617
GetParameters 2619
GetPatchBaseline 2623
GetPatchBaselineForPatchGroup 2625
ListAssociationVersions 2626
ListAssociations 2628
ListCommandInvocations 2633
ListCommands 2637
ListComplianceItems 2643
ListComplianceSummaries 2646
ListDocumentVersions 2649
ListDocuments 2650
ListInventoryEntries 2653
ListResourceComplianceSummaries 2656
ListTagsForResource 2659

ModifyDocumentPermission ... 2660
PutComplianceItems ... 2662
PutInventory ... 2663
PutParameter ... 2664
RegisterDefaultPatchBaseline ... 2671
RegisterPatchBaselineForPatchGroup ... 2672
RegisterTargetWithMaintenanceWindow ... 2674
RegisterTaskWithMaintenanceWindow ... 2677
RemoveTagsFromResource ... 2684
SendCommand ... 2685
StartAutomationExecution ... 2692
StopAutomationExecution ... 2694
UpdateAssociation ... 2695
UpdateAssociationStatus ... 2698
UpdateDocument ... 2700
UpdateDocumentDefaultVersion ... 2702
UpdateMaintenanceWindow ... 2704
UpdateManagedInstanceRole ... 2707
UpdateOpsItem ... 2708
UpdatePatchBaseline ... 2710
Szenarien ... 2712
Erste Schritte mit Systems Manager ... 2713
Überwachen ... 2728
Überwachungstools ... 2729
Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent) ... 2729
Migrieren Sie die Erfassung von Windows Server-Knotenprotokollen auf den CloudWatch Agenten ... 2731
Speichern Sie die CloudWatch Agentenkonfigurationseinstellungen in Parameter Store ... 2743
Rollback zur Protokollerfassung mit SSM Agent ... 2744
Senden von SSM Agent-Protokollen an CloudWatch Logs ... 2748
Überwachung der Ereignisse Ihrer Änderungsanfragen ... 2751
Überwachung Ihrer Automatisierungen ... 2754
Automation-Metriken ... 2755
Überwachen von Run Command-Metriken mit Amazon CloudWatch ... 2755
Systems Manager Run Command-Metriken und -Dimensionen ... 2756
AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail ... 2757

Systems Manager Manager-Datenereignisse in CloudTrail ... 2759
Systems Manager Manager-Verwaltungsereignisse in CloudTrail ... 2761
Beispiele Systems Manager Manager-Ereignisse ... 2761
Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs ... 2767
Konfiguration von Amazon CloudWatch Logs für Run Command ... 2771
CloudWatch Logs angeben, wenn Sie Befehle senden ... 2772
Befehlsausgabe in CloudWatch Logs anzeigen ... 2773
Überwachung mit Amazon EventBridge ... 2773
Konfigurieren von EventBridge für Systems Manager-Ereignisse ... 2775
EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager ... 2779
Beispielszenarien: Systems-Manager-Ziele in Amazon-EventBridge-Regeln ... 2794
Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen ... 2795
Konfigurieren von Amazon SNS-Benachrichtigungen für AWS Systems Manager ... 2796
Beispiele für Amazon SNS-Benachrichtigungen für AWS Systems Manager ... 2807
Verwenden von Run Command zum Senden eines Befehls, der Statusbenachrichtigungen zurückgibt ... 2808
Verwenden eines Wartungsfensters zum Senden eines Befehls, der Statusbenachrichtigungen zurückgibt ... 2812
Produkt- und Service-Integrationen ... 2819
Integration mit AWS-Services ... 2819
Datenverarbeitung ... 2819
Internet of Things (IoT) ... 2822
Speicher ... 2823
Entwicklertools ... 2824
Sicherheit, Identität und Compliance ... 2825
Kryptografie und PKI ... 2828
Verwaltung und Governance ... 2828
Netzwerk und Bereitstellung von Inhalten ... 2835
Analysen ... 2836
Anwendungsintegration ... 2838
AWS Management Console ... 2839
Ausführen von Skripts von Amazon S3 ... 2840
Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter ... 2845
Verwenden von Parameter Store-Parametern in AWS Lambda -Funktionen ... 2851

Integration in andere Produkte und Services ... 2870
Ausführen von Skripts von GitHub ... 2873
Verwenden von Chef InSpec Profilen mit Systems Manager Compliance ... 2882
Integration mit ServiceNow ... 2888
Markieren von Systems Manager-Ressourcen ... 2890
Systems-Manager-Ressourcen, die Sie mit Tags versehen können ... 2891
Markieren von Systems-Manager-Zuordnungen ... 2892
Erstellen von Zuordnungen mit Tags ... 2893
Hinzufügen von Tags zu einer vorhandenen Zuordnung ... 2893
Entfernen von Tags aus einer Zuordnung ... 2894
Markieren von Automatisierungen ... 2896
Hinzufügen von Tags zu Automatisierungen (Konsole) ... 2896
Hinzufügen von Tags zu Automatisierungen (Befehlszeile) ... 2897
Entfernen von Tags aus Automatisierungen ... 2899
Markierungen von Systems Manager-Dokumenten ... 2900
Erstellen von Dokumenten mit Tags ... 2901
Hinzufügen von Tags zu vorhandenen Dokumenten ... 2901
Entfernen von Tags aus SSM-Dokumenten ... 2904
Markieren von Wartungsfenstern ... 2906
Erstellen von Wartungsfenstern mit Tags ... 2906
Hinzufügen von Tags zu vorhandenen Wartungsfenstern ... 2906
Entfernen von Tags aus Wartungsfenstern ... 2909
Markieren verwalteter Knoten ... 2911
Erstellen oder Aktivieren verwalteter Knoten mit Tags ... 2912
Hinzufügen von Tags zu vorhandenen verwalteten Knoten ... 2912
Entfernen von Tags aus verwalteten Knoten ... 2915
Markieren von OpsItems ... 2918
Erstellen von OpsItems mit Tags ... 2918
Hinzufügen von Tags zu vorhandenen OpsItems ... 2918
Entfernen von Tags aus Systems Manager OpsItems ... 2920
Markieren von Systems Manager-Parametern ... 2922
Erstellen von Parametern mit Tags ... 2922
Hinzufügen von Tags zu vorhandenen Parametern ... 2923
Entfernen von Tags aus SSM-Parametern ... 2925
Markieren von Patch-Baselines ... 2927
Erstellen von Patch-Baselines mit Tags ... 2927

Hinzufügen von Tags zu vorhandenen Patch-Baselines ... 2928
Entfernen von Tags aus Patch-Baselines ... 2930
AWS Systems Manager Referenz ... 2933
Amazon EventBridge Ereignismuster und -typen für Systems Manager ... 2934
Ereignistyp: Automation ... 2935
Ereignistyp: Change Calendar ... 2936
Ereignistyp: Change Manager ... 2936
Ereignistyp: Configuration Compliance ... 2937
Ereignistyp: Inventory ... 2937
Ereignistyp: Wartungsfenster ... 2938
Ereignistyp: OpsCenter ... 2941
Ereignistyp: Parameter Store ... 2941
Ereignistyp: Run Command ... 2942
Ereignistyp: State Manager ... 2943
Cron- und Rate-Ausdrücke ... 2944
Allgemeine Informationen zu Cron- und Rate-Ausdrücken ... 2944
Cron- und Rate-Ausdrücke für Zuordnungen ... 2950
Cron- und Rate-Ausdrücke für Wartungsfenster ... 2953
ec2messages, ssmmessages und andere API-Operationen ... 2955
API-Operationen (ssmmessagesund ec2messages Endpunkte) im Zusammenhang mit Agenten ... 2956
ssm:*API-Operationen im Zusammenhang mit Namespace-Instanzen ... 2958
Erstellen formatierter Datums- und Uhrzeitzeichenfolgen für Systems Manager ... 2959
Formatieren von Datums- und Uhrzeitzeichenfolgen für Systems Manager ... 2960
Erstellen benutzerdefinierter Datums- und Uhrzeitzeichenfolgen für Systems Manager ... 2960
Anwendungsfälle und bewährte Methoden ... 2964
Löschen von Systems Manager Ressourcen und Artefakten ... 2967
Auswahl zwischen State Manager und Maintenance Windows ... 2972
State Manager und Maintenance Windows: Hauptanwendungsfälle ... 2972
Ähnliche Informationen ... 2981
Dokumentverlauf ... 2983
Updates vor Juni 2018 ... 3185
Dokumentkonventionen ... 3207
AWS Glossar ... 3209
... mmmccx

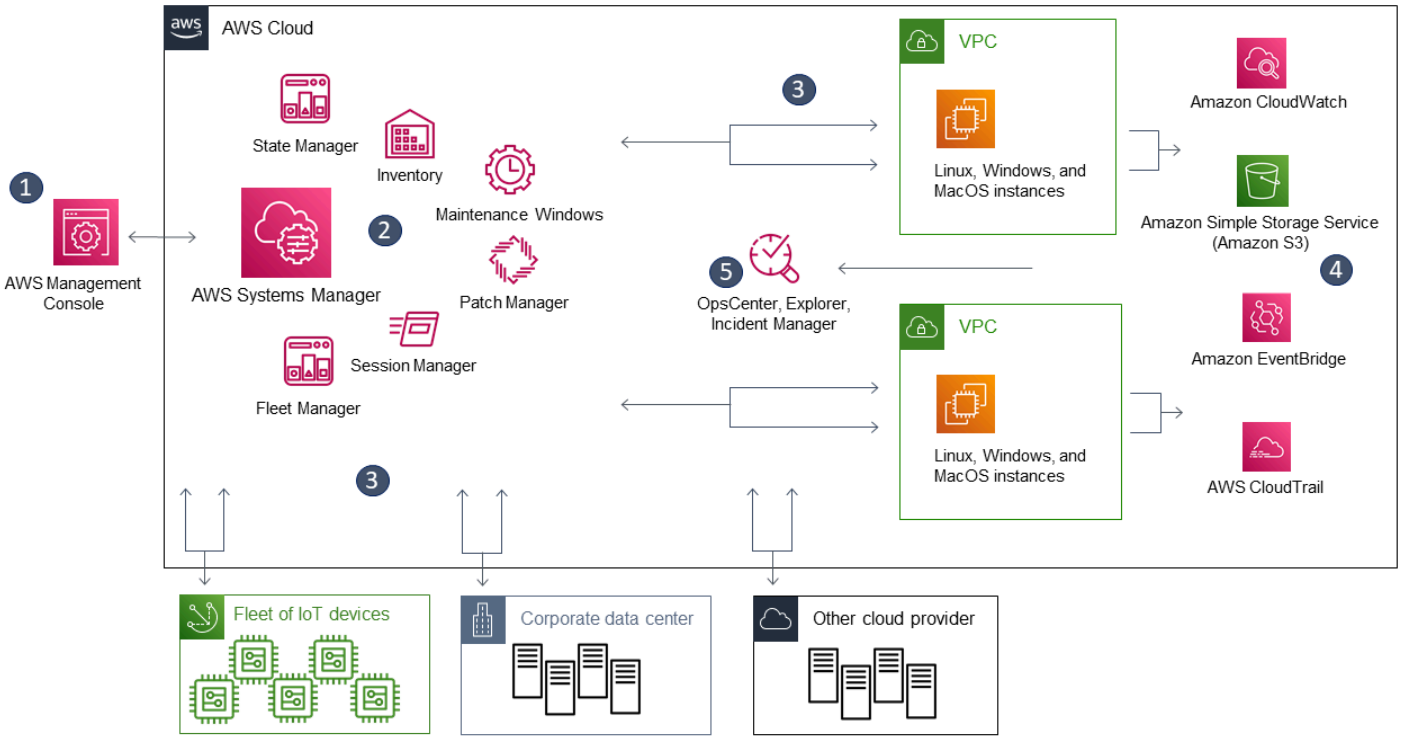
Was ist AWS Systems Manager?

AWS Systems Manager ist die zentrale Anlaufstelle für Ihre AWS Anwendungen und Ressourcen und eine sichere end-to-end Verwaltungslösung für Hybrid- und Multi-Cloud-Umgebungen, die sichere Betriebsabläufe in großem Maßstab ermöglicht.

Funktionsweise von Systems Manager

Das folgende Diagramm zeigt, wie einige System-Manager-Funktionen Aktionen für Ihre Ressourcen durchführen. Das Diagramm deckt nicht alle Funktionen ab. Jede aufgezählte Interaktion wird vor dem Diagramm beschrieben.

1. Zugriff auf Systems Manager – Verwenden Sie eine der verfügbaren Optionen für den Zugriff auf Systems Manager.
2. Wählen Sie eine System-Manager-Funktion – Bestimmen Sie, welche Funktion Ihnen helfen kann, die Aktion auszuführen, die Sie für Ihre Ressourcen ausführen möchten. Das Diagramm zeigt nur einige der Funktionen, die IT-Administratoren und DevOps Mitarbeiter zur Verwaltung ihrer Anwendungen und Ressourcen nutzen.
3. Überprüfung und Verarbeitung — Systems Manager überprüft, ob Ihr Benutzer, Ihre Gruppe oder Rolle über die erforderlichen AWS Identity and Access Management (IAM-) Berechtigungen verfügt, um die von Ihnen angegebene Aktion auszuführen. Wenn das Ziel Ihrer Aktion ein verwalteter Knoten ist, wird der auf dem Knoten aufgeführte Systems Manager Agent (SSM Agent) die Aktion ausführen. Bei anderen Ressourcentypen führt Systems Manager die angegebene Aktion aus oder kommuniziert mit anderen AWS-Services , um die Aktion im Namen von Systems Manager auszuführen.
4. Berichterstellung – Systems Manager, SSM Agent und andere AWS-Services , die eine Aktion im Auftrag des Berichtsstatus von Systems Manager durchgeführt haben. Systems Manager kann Statusdetails an andere senden AWS-Services, sofern dies konfiguriert ist.
5. Funktionalitäten für das Betriebsmanagement von Systems Manager – Wenn diese Option aktiviert ist, können System-Manager-Betriebsverwaltungs-Funktionen wie Explorer OpsCenter und Incident Manager Betriebsdaten aggregieren oder Artefakte als Reaktion auf Ereignisse oder Fehler mit Ihren Ressourcen erstellen. Zu diesen Artefakten gehören operative Arbeitselemente (OpsItems) und Vorfälle. Die Betriebsverwaltungsfunktionen von Systems Manager bieten betrieblichen Einblick in Ihre Anwendungen und Ressourcen sowie automatisierte Behebungslösungen zur Behebung von Problemen.



Systems Manager-Funktionen

Systems Manager gruppiert Funktionen in die folgenden Kategorien. Wählen Sie die Registerkarten unter jeder Kategorie aus, um mehr über jede Funktion zu erfahren.

Themen

- Application Management
- Änderungsmanagement
- Node Management
- Verfahrensmanagement
- Quick Setup
- Gemeinsam genutzte -Ressourcen

Application Management

Application Manager

Application Managerunterstützt DevOps Techniker bei der Untersuchung und Behebung von Problemen mit ihren AWS Ressourcen im Kontext ihrer Anwendungen und Cluster. In Application

Manager ist ein Anwendung eine logische Gruppierung von AWS Ressourcen, die Sie als Einheit betreiben möchten. Diese logische Gruppe kann verschiedene Versionen einer Anwendung, Eigentumsgrenzen für Betreiber oder Entwicklerumgebungen repräsentieren, um nur einige zu nennen. Application ManagerDie Unterstützung für Container-Cluster umfasst sowohl Amazon Elastic Kubernetes Service (Amazon EKS) als auch Amazon Elastic Container Service (Amazon ECS) -Cluster. Application Managerfasst Betriebsinformationen aus mehreren Funktionen AWS-Services und Systems Manager Manager-Funktionen in einer einzigen AWS Management Console zusammen.

AppConfig

Mit AppConfig können Sie Anwendungskonfigurationen und Feature-Flags erstellen, verwalten und bereitstellen. AppConfig unterstützt kontrollierte Bereitstellungen für Anwendungen jeder Größe. Sie können AppConfig mit Anwendungen verwenden, die auf Amazon-EC2-Instances, in AWS Lambda -Containern, mobilen Anwendungen oder auf Edge-Geräten gehostet werden. Um Fehler bei der Bereitstellung von Anwendungskonfigurationen zu vermeiden, enthält AppConfig Validierungen. Eine Validierung stellt durch eine syntaktische oder semantische Prüfung sicher, dass die Konfiguration, die Sie bereitstellen möchten, wie beabsichtigt funktioniert. Während einer Konfigurationsbereitstellung überwacht AppConfig die Anwendung, um sicherzustellen, dass die Bereitstellung erfolgreich ist. Falls im System ein Fehler auftritt oder die Bereitstellung einen Alarm auslöst, setzt AppConfig die Änderung zurück, um die Auswirkungen auf die Benutzer Ihrer Anwendung zu minimieren.

Parameter Store

Parameter Store ermöglicht eine sichere, hierarchische Speicherung für die Konfigurationsdatenverwaltung und das Verschlüsselungsmanagement. Sie können Daten wie Passwörter, Datenbankzeichenfolgen, Amazon Elastic Compute Cloud (Amazon EC2)-Instance-IDs und Amazon Machine Image (AMI)-IDs sowie Lizenzcodes als Parameterwerte speichern. Sie können Werte als Klartext oder als verschlüsselte Daten speichern. Anschließend können Sie die Werte anhand des eindeutigen Namens, den Sie beim Erstellen des Parameters angegeben haben, referenzieren.

Änderungsmanagement

Change Manager

Change Manager ist ein Change-Management-Framework für Unternehmen zum Anfordern, Genehmigen, Implementieren und Melden von Betriebsänderungen an Ihrer

Anwendungskonfiguration und Infrastruktur. Wenn Sie ein einziges delegiertes Administratorkonto verwenden AWS Organizations, können Sie Änderungen an mehreren oder mehreren AWS-Konten verwalten. AWS-Regionen Alternativ können Sie mit einem lokalen Konto Änderungen für einen einzigen AWS-Konto verwalten. Wird Change Manager für die Verwaltung von Änderungen sowohl an AWS Ressourcen als auch an lokalen Ressourcen verwendet.

Automation

Mit Automation werden häufig durchzuführende Wartungs- und Bereitstellungsaufgaben automatisiert. Sie können Automation verwenden, um Amazon Machine Images (AMIs) zu erstellen und zu aktualisieren, Treiber- und Agent-Updates anzuwenden, Passwörter auf Windows Server-Instances zurückzusetzen, SSH-Schlüssel auf Linux-Instances zurückzusetzen oder OS-Patches oder Anwendungs-Updates anzuwenden.

Kalender ändern

Change Calendar hilft Ihnen, Datums- und Uhrzeitbereiche einzurichten, in denen von Ihnen angegebene Aktionen (z. B. in Systems-Manager-Automatisierungs-Runbooks) in Ihrem AWS-Konto ausgeführt bzw. nicht ausgeführt werden können. In Change Calendar werden diese Bereiche Ereignisse genannt. Wenn Sie einen Change Calendar-Eintrag erstellen, erstellen Sie ein Systems Manager-Dokument vom Typ `ChangeCalendar`. In Change Calendar wird das Dokument als iCalendar 2.0-Daten im Klartextformat gespeichert. Ereignisse, die Sie dem Change Calendar-Eintrag hinzufügen, werden Teil des Dokuments. Sie können Ereignisse manuell in der Change Calendar-Schnittstelle hinzufügen oder Ereignisse aus einem unterstützten Kalender eines Drittanbieters mithilfe einer `.ics`-Datei importieren.

Wartungsfenster

Mit Maintenance Windows können Sie wiederkehrende Zeitpläne für verwaltete Instances einrichten, um Verwaltungsaufgaben wie etwa die Installation von Patches und Updates ohne Unterbrechung der geschäftskritischen Vorgänge auszuführen.

Node Management

Ein verwalteter Knoten ist jede Maschine, die für die Verwendung mit Systems Manager in Hybrid- und Multi-Cloud-Umgebungen konfiguriert ist.

Compliance

Mit Compliance können Sie Ihre Flotte verwalteter Knoten auf Patch-Compliance und Konfigurations-Inkonsistenzen hin scannen. Sie können Daten aus mehreren AWS-Konten

Bereichen sammeln und aggregieren und dann nach bestimmten Ressourcen suchen, die nicht den Vorschriften entsprechen. AWS-Regionen Standardmäßig werden von Compliance die Compliance-Daten zu Patch Manager-Patches und State Manager-Zuordnungen angezeigt. Sie können auch den Service anpassen und Ihre eigenen Compliance-Typen auf Grundlage Ihrer IT- oder Business-Anforderungen erstellen.

Fleet Manager

Fleet Manager ist eine einheitliche Benutzeroberfläche (UI), mit der Sie Ihre Knoten remote verwalten können. Mit Fleet Manager können Sie sich den Zustand und den Leistungsstatus Ihrer gesamten Flotte von einer Konsole aus ansehen. Sie können auch Daten aus einzelnen Geräten und Instances sammeln, um allgemeine Problembehandlungs- und Verwaltungsaufgaben über die Konsole auszuführen. Dazu gehören das Anzeigen von Verzeichnis- und Dateiinhalten, die Windows-Registry-Verwaltung, die Betriebssystembenutzerverwaltung und vieles mehr.

Inventory

Inventory (Bestand) automatisiert die Erfassung des Software-Bestands auf Ihren verwalteten Instances. Sie können mit Inventory Metadaten über Anwendungen, Dateien, Komponenten, Patches und vieles mehr erfassen.

Session Manager

Verwenden Sie diese Option Session Manager, um Ihre Edge-Geräte und Amazon Elastic Compute Cloud (Amazon EC2) -Instances über eine interaktive browserbasierte Shell mit einem Klick oder über die zu verwalten. AWS CLISession Managerbietet sicheres und überprüfbares Edge-Geräte- und Instance-Management, ohne dass eingehende Ports geöffnet, Bastion-Hosts verwaltet oder SSH-Schlüssel verwaltet werden müssen. Session Managerermöglicht Ihnen außerdem die Einhaltung von Unternehmensrichtlinien, die einen kontrollierten Zugriff auf Edge-Geräte und -Instances, strenge Sicherheitsvorkehrungen und vollständig überprüfbare Protokolle mit Edge-Geräten und Instanzzugriffsdetails vorschreiben, und bietet Endbenutzern gleichzeitig einen einfachen plattformübergreifenden Zugriff auf Ihre Edge-Geräte und EC2-Instances mit nur einem Klick. Um Session Manager zu verwenden, müssen Sie das Advanced-Instances-Kontingent aktivieren. Weitere Informationen finden Sie unter Aktivieren des Kontingents für erweiterte Instances.

Run Command

Mit Run Command können Sie die Konfiguration Ihrer verwalteten Instances in großem Umfang remote und sicher verwalten. Verwenden Sie Run Command zum Ausführen von On-Demand-Änderungen, wie das Aktualisieren von Anwendungen oder das Ausführen von Linux-Shell-

Skripts und Windows PowerShell-Befehlen auf einem Zielsatz mit Dutzenden oder Hunderten von verwalteten Knoten.

State Manager

State Manager ermöglicht die Automatisierung des Prozesses, mit dem Ihre verwalteten Knoten in einem definierten Zustand gehalten werden. Mit State Manager können Sie bei Ihren verwalteten Knoten sicherstellen, dass ihnen per Bootstrapping bestimmte Software beim Startup angefügt wird, dass sie mit einer Windows-Domain verbunden werden (nur Windows Server-Knoten) oder dass sie mit bestimmten Software-Updates gepatcht werden.

Patchmanager

Verwenden Sie Patch Manager zum Automatisieren des Patch-Vorgangs für Ihre verwalteten Knoten, sowohl mit Sicherheits-Updates als auch anderen Arten von Updates. Sie können Patch Manager verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows Server ist der Anwendungssupport auf Updates für Microsoft-Anwendungen beschränkt.)

Mit dieser Funktion können Sie verwaltete Knoten auf fehlende Patches scannen und diese mit Tags einzeln oder bei großen Gruppen von verwalteten Knoten anwenden. Patch Manager nutzt Patch-Baselines, die Regeln für die automatische Genehmigung von Patches innerhalb einer festgelegten Anzahl von Tagen nach ihrer Veröffentlichung enthalten können, sowie Listen von genehmigten und abgelehnten Patches. Sie können sicherheitsrelevante Patches regelmäßig installieren, indem Sie das Einspielen von Patches als Systems-Manager-Wartungsfenster-Aufgabe planen, oder Sie können Ihre verwalteten Knoten jederzeit bei Bedarf patchen.

Für Linux-Betriebssysteme können Sie als Teil Ihrer Patch-Baseline die Repositorys definieren, die für Patching-Operationen verwendet werden sollen. Auf diese Weise können Sie sicherstellen, dass Updates nur aus vertrauenswürdigen Repositorys installiert werden, unabhängig davon, welche Repositorys auf dem verwalteten Knoten konfiguriert sind. Für Linux haben Sie auch die Möglichkeit, ein beliebiges Paket auf dem verwalteten Knoten zu aktualisieren, nicht nur diejenigen, die als Sicherheits-Updates für Betriebssysteme eingestuft sind. Sie können Patchberichte auch generieren, die an einen S3-Bucket Ihrer Wahl gesendet werden. Für einen einzelnen verwalteten Knoten enthalten Berichte Details aller Patches für die Maschine. Für einen Bericht über alle verwaltete Knoten wird nur eine Zusammenfassung der fehlenden Patches bereitgestellt.

Distributor

Distributor ermöglicht das Erstellen von Paketen und deren Bereitstellung auf verwalteten Knoten. Mit Distributor können Sie Ihre eigene Software paketieren oder nach AWS bereitgestellten Agent-Softwarepaketen suchen, z. B. für die Installation auf verwalteten Systems AmazonCloudWatchAgentManager Manager-Knoten. Nachdem Sie ein Paket zum ersten Mal installiert haben, können Sie Distributor verwenden, um eine neue Paketversion zu deinstallieren und neu zu installieren oder um eine direkte Aktualisierung durchzuführen, bei der neue oder geänderte Dateien hinzufügt werden. Distributor veröffentlicht Ressourcen (z. B. Softwarepakete) auf von Systems Manager verwalteten Knoten.

Hybrid Activations

Um Nicht-EC2-Maschinen in Ihrer Hybrid- und Multi-Cloud-Umgebung als verwaltete Knoten einzurichten, erstellen Sie eine Hybrid-Aktivierung. Nach Abschluss der Aktivierung erhalten Sie einen Aktivierungscode und eine ID. Diese Code- und ID-Kombination funktioniert wie eine Amazon Elastic Compute Cloud (Amazon EC2)-Zugriffs-ID und ein geheimer Schlüssel, um einen sicheren Zugriff auf den Systems-Manager-Service von Ihren verwalteten Instances aus zu ermöglichen.

Sie können auch eine Aktivierung für Edge-Geräte erstellen, wenn Sie diese mithilfe von Systems Manager verwalten möchten.

Verfahrensmanagement

Incident Manager

Incident Manager ist eine Incident-Management-Konsole, die Benutzern hilft, Vorfälle, die sich auf ihre gehosteten Anwendungen auswirken, zu minimieren und diese zu beheben. AWS

Incident Manager beschleunigt die Behebung von Vorfällen, indem es die zuständigen Mitarbeiter über die Auswirkungen informiert, relevante Daten zur Fehlerbehebung hervorhebt und Tools für die Zusammenarbeit bereitstellt, um die Dienste wieder zum Laufen zu bringen. Incident Manager automatisiert auch Antwortpläne und ermöglicht die Eskalation des Responder-Teams.

Explorer

Explorerist ein anpassbares Operations-Dashboard, das Informationen über Ihre AWS Ressourcen enthält. Explorerzeigt eine aggregierte Ansicht der Betriebsdaten (OpsData) für Sie AWS-Konten und Across AWS-Regionen an. OpsData Enthält Metadaten zu Ihren Amazon

EC2 EC2-Instances, Details zur Patch-Konformität und betriebliche Arbeitsaufgaben (OpsItems). Explorer Explorerbietet einen Überblick darüber, wie sie auf Ihre Geschäftsbereiche oder Anwendungen verteilt OpsItems sind, wie sie sich im Laufe der Zeit entwickeln und wie sie sich je nach Kategorie unterscheiden. Sie können Informationen in Explorer gruppieren und filtern, um sich auf die Elemente zu konzentrieren, die für Sie relevant sind und eine Aktion erfordern. Wenn Sie Probleme mit hoher Priorität erkennen, können Sie OpsCenter, eine Funktion von Systems Manager verwenden, um Automation-Runbooks ausführen und die Probleme beheben.

OpsCenter

OpsCenterbietet einen zentralen Ort, an dem Betriebsingenieure und IT Fachkräfte betriebliche Arbeitsaufgaben (OpsItems) im Zusammenhang mit AWS Ressourcen einsehen, untersuchen und lösen können. OpsCenterwurde entwickelt, um die durchschnittliche Zeit bis zur Lösung von Problemen zu reduzieren, die sich auf AWS Ressourcen auswirken. Diese Systems Manager-Funktion aggregiert und standardisiert OpsItems über Services hinweg und bietet gleichzeitig kontextbezogene Untersuchungsdaten über jedes OpsItem, verwandte OpsItems und verwandte Ressourcen. OpsCenter bietet außerdem Systems-Automatisierungsdokumente (Runbooks), die Sie verwenden können, um Probleme schnell zu lösen. Sie können durchsuchbare, benutzerdefinierte Daten für jedes OpsItem angeben. Sie können automatisch generierte Zusammenfassungsberichte über OpsItems nach Status und Quelle anzeigen.

CloudWatch Dashboards

CloudWatch Amazon-Dashboards sind anpassbare Seiten in der CloudWatch Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können, auch die Ressourcen, die über verschiedene Regionen verteilt sind. Sie können CloudWatch Dashboards verwenden, um benutzerdefinierte Ansichten der Kennzahlen und Alarme für Ihre AWS Ressourcen zu erstellen.

Quick Setup

Verwenden Sie diese Quick SetupOption, um häufig verwendete Funktionen AWS-Services und Funktionen mit empfohlenen Best Practices zu konfigurieren. Sie können es einzeln AWS-Konto oder Quick Setup in mehreren verwenden AWS-Konten und AWS-Regionen durch Integration mit AWS Organizations. Quick Setupvereinfacht die Einrichtung von Diensten, einschließlich Systems Manager, durch die Automatisierung häufiger oder empfohlener Aufgaben. Zu diesen Aufgaben gehören beispielsweise die Erstellung der erforderlichen Instanzprofilrollen AWS Identity and Access Management (IAM) und die Einrichtung betrieblicher Best Practices wie regelmäßige Patchscans und Inventarerfassung.

Gemeinsam genutzte -Ressourcen

Documents

Ein Systems Manager-Dokument (SSM-Dokument) definiert die Aktionen, die Systems Manager ausführt. Zu den SSM-Dokumenttypen gehören Command-Dokumente, die von State Manager und Run Command verwendet werden, und Automation-Runbooks, die von Systems Manager Automation verwendet werden. Systems Manager umfasst Dutzende vorkonfigurierter Dokumente, die Sie verwenden können, indem Sie zur Laufzeit Parameter angeben. Die Dokumente können im JSON- oder YAML-Format vorliegen und enthalten die von Ihnen angegeben Schritte und Parameter.

Zugriff auf Systems Manager

Sie können folgendermaßen mit Systems Manager arbeiten:

Systems Manager-Konsole

Die Systems-Manager-Konsole ist eine browserbasierte Schnittstelle für den Zugriff auf und die Verwendung von Systems Manager.

AWS IoT Greengrass V2 Konsole

Sie können Edge-Geräte, für die sie konfiguriert sind, AWS IoT Greengrass in der Greengrass-Konsole anzeigen und verwalten.

AWS Befehlszeilentools

Mithilfe der AWS Befehlszeilentools können Sie Befehle an der Befehlszeile Ihres Systems ausgeben, um Systems Manager und andere AWS Aufgaben auszuführen. Die Tools werden unter Linux, macOS und Windows unterstützt. Die AWS Command Line Interface (AWS CLI) kann schneller und bequemer zu verwenden sein als die Konsole. Die Befehlszeilen-Tools sind auch hilfreich, wenn Sie Skripts erstellen möchten, die AWS -Aufgaben ausführen.

AWS stellt zwei Gruppen von Befehlszeilentools bereit: das AWS Command Line Interfaceund das AWS Tools for Windows PowerShell. Informationen zur Installation und Verwendung von finden Sie im AWS Command Line Interface Benutzerhandbuch. AWS CLI Informationen zur Installation und Verwendung der Tools für Windows PowerShell finden Sie im AWS Tools for Windows PowerShell Benutzerhandbuch.

> **Note**
>
> Auf Ihren Windows Server-Instances wird Windows PowerShell 3.0 oder höher benötigt, um bestimmte SSM-Dokumente ausführen zu können (z. B. das Legacy–`AWS-ApplyPatchBaseline`Dokument). Überprüfen Sie, ob Ihre Windows Server-Instances auf Windows Management Framework 3.0 oder höher ausgeführt werden. Das Framework umfasst Windows PowerShell.

AWS SDKs

AWS bietet Software Development Kits (SDKs), die aus Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen bestehen (z. B. Java, Python, Ruby, .NET, iOS und Android und andere). Die SDKs bieten eine bequeme Möglichkeit, programmgesteuerten Zugriff auf Systems Manager zu gewähren. Informationen zu den AWS SDKs, einschließlich deren Download und Installation, finden Sie unter Tools für Amazon Web Services.

Systems Manager Service-Namengeschichte

AWS Systems Manager (Systems Manager) war früher bekannt als Amazon Simple Systems Manager (SSM) "und"Amazon EC2 Systems Manager (SSM)“. Der ursprüngliche abgekürzte Name des Dienstes, "SSM„, findet sich immer noch in verschiedenen AWS Ressourcen wieder, darunter auch in einigen anderen Servicekonsolen. Hier einige Beispiele:

- Systems Manager Agent: SSM Agent
- Systems Manager Parameter: SSM-Parameter
- Systems Manager-Service-Endpunkte: `ssm.region.amazonaws.com`
- AWS CloudFormation Ressourcentypen: `AWS::SSM::Document`
- AWS Config Regel-ID: `EC2_INSTANCE_MANAGED_BY_SSM`
- AWS Command Line Interface (AWS CLI) Befehle: `aws ssm describe-patch-baselines`
- AWS Identity and Access Management Namen der (IAM) verwalteten Richtlinien: `AmazonSSMReadOnlyAccess`
- Systems Manager Ressourcen-ARNs: `arn:aws:ssm:region:account-id:patchbaseline/pb-07d8884178EXAMPLE`

Unterstützt AWS-Regionen

Systems Manager ist auf den in Systems Manager AWS-Regionen aufgeführten Dienstendpunkten in der Allgemeine Amazon Web Services-Referenzverfügbar. Bevor Sie mit dem Systems Manager Manager-Konfigurationsprozess beginnen, empfehlen wir Ihnen, zu überprüfen, ob der Dienst in allen Ländern verfügbar ist, in denen AWS-Regionen Sie ihn verwenden möchten.

Für Nicht-EC2-Maschinen in Ihrer Hybrid- und Multi-Cloud-Umgebung empfehlen wir Ihnen, die Region zu wählen, die Ihrem Rechenzentrum oder Ihrer Computerumgebung am nächsten liegt.

Unterstützte Betriebssysteme und Maschinentypen

Bevor Sie mit Systems Manager arbeiten, stellen Sie sicher, dass Ihr Betriebssystem (OS), Ihre Betriebssystemversion und Ihr Maschinentyp als verwaltete Knoten unterstützt werden.

Themen

- Unterstützte Betriebssysteme für Systems Manager
- Unterstützte Maschinentypen in Hybrid- und Multi-Cloud-Umgebungen

Unterstützte Betriebssysteme für Systems Manager

In den folgenden Abschnitten sind die Betriebssysteme und Betriebssystemversionen aufgeführt, die von Systems Manager unterstützt werden.

> **Note**
>
> Wenn Sie planen, AWS IoT Greengrass Kerngeräte mithilfe von Systems Manager zu verwalten und zu konfigurieren, müssen diese Geräte die Anforderungen für erfüllen AWS IoT Greengrass. Weitere Informationen finden Sie im AWS IoT Greengrass Version 2 Entwicklerhandbuch unter Einrichten von AWS IoT Greengrass Kerngeräten.
> Wenn Sie Geräte verwalten AWS IoT und konfigurieren möchten, die keine AWS Edge-Geräte sind, müssen diese Geräte die hier aufgeführten Anforderungen erfüllen und als lokal verwaltete Knoten für Systems Manager konfiguriert sein. Weitere Informationen finden Sie unter Verwaltung von Edge-Geräten mit Systems Manager.

> **Important**
>
> Patch Manager, eine Funktion von Systems Manager, unterstützt möglicherweise nicht alle in diesem Thema aufgeführten OS-Versionen. Eine Liste der von Patch Manager unterstützten Betriebssystemversionen finden Sie unter Patch Manager-Voraussetzungen.

Betriebssystemtypen

- Linux
- macOS (nur Amazon-EC2-Instances)
- Raspberry Pi OS (früher Raspbian)
- Windows Server

Linux

AlmaLinux

Versionen	x86	x86_64	ARM64
8.3—8.9		✓	✓
9.0–9.2		✓	✓

Amazon Linux 1

Versionen	x86	x86_64	ARM64
2012.03–2018.03	✓	✓	

> **Note**
>
> Ab Version 2015.03 wurde Amazon Linux 1 in x86_64 Versionen veröffentlicht. Amazon Linux 1 hat am 31. Dezember 2020 das Ende seines Standardsupports erreicht und am 31. Dezember 2023 das Ende seiner Lebensdauer erreicht, wie im Update zu AMI end-of-life Amazon Linux im AWS News-Blog angekündigt. AWS bietet Amazon Machine

Images (AMIs) für dieses Betriebssystem nicht mehr an. AWS Systems Manager bietet jedoch weiterhin Support für bestehende Amazon Linux 1-Instances.

Amazon Linux 2

Versionen	x86	x86_64	ARM64
2.0 und alle späteren Versionen		✓	✓

Amazon Linux 2023

Versionen	x86	x86_64	ARM64
2023.0.20230315.0 und alle späteren Versionen		✓	✓

Bottlerocket

Versionen	x86_64	ARM64
1.0.0 und alle späteren Versionen	✓	✓

CentOS

Versionen	x86	x86_64	ARM64
6.x[1]	✓	✓	
7.1 und neuere 7.x-Versionen		✓	✓
8.0–8.5		✓	✓

¹ Damit Sie diese Versionen verwenden können, benötigen Sie eine 3.0.x-Version der SSM Agent. Wir empfehlen, die letzte verfügbare Version 3.0.x von SSM Agent zu verwenden. Ältere SSM Agent-Versionen (3.1 oder höher) werden nicht unterstützt.

CentOS Stream

Versionen	x86	x86_64	ARM64
8		✓	✓

Debian Server

Versionen	x86	x86_64	ARM64
Jessie (8)		✓	
Stretch (9)		✓	✓
Buster (10)		✓	✓
Bullseye (11)		✓	✓
Bookworm(12)		✓	✓

Oracle Linux

Versionen	x86	x86_64	ARM64
7.5–7.8		✓	
8,1—8,9		✓	
9.0–9.2		✓	

Red Hat Enterprise Linux (RHEL)

Versionen	x86	x86_64	ARM64
6.x[1]	✓	✓	

Versionen	x86	x86_64	ARM64
7.0–7.5		✓	
7,6—8,9		✓	✓
9,0—9,3		✓	✓

[1] Damit Sie diese Versionen verwenden können, benötigen Sie eine 3.0.x-Version der SSM Agent. Wir empfehlen, die letzte verfügbare Version 3.0.x von SSM Agent zu verwenden. Ältere SSM Agent-Versionen (3.1 oder höher) werden nicht unterstützt.

Rocky Linux

Versionen	x86	x86_64	ARM64
8,4—8,9		✓	✓
9.0–9.2		✓	✓

SUSE Linux Enterprise Server (SLES)

Versionen	x86	x86_64	ARM64
12 und neuere 12.x-Versionen		✓	
15 und neuere 15.x-Versionen		✓	✓

Ubuntu Server

Versionen	x86	x86_64	ARM64
12.04 LTS und 14.04 LTS	✓	✓	

Versionen	x86	x86_64	ARM64
16.04 LTS und 18.04 LTS		✓	✓
20.04 LTS und 20.10 STR		✓	✓
22.04 LTS		✓	✓
23,04		✓	✓

macOS (nur Amazon-EC2-Instances)

Version	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)		✓	✓

> **Note**
>
> macOSwird nicht in allen AWS-Regionen unterstützt. Weitere Informationen zur Amazon EC2-Unterstützung für macOS finden Sie unter Amazon EC2 Mac-Instances im Amazon EC2 EC2-Benutzerhandbuch.

Raspberry Pi OS (früher Raspbian)

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

Weitere Informationen

- Verwalten von Raspberry Pi-Geräte mit AWS Systems Manager

Windows Server

SSM Agenterfordert Windows PowerShell 3.0 oder später die Ausführung bestimmter AWS Systems Manager Dokumente (SSM-Dokumente) auf Windows Server Instances (z. B. das ältere Dokument). `AWS-ApplyPatchBaseline` Überprüfen Sie, ob Ihre Windows Server-Instances auf Windows Management Framework 3.0 oder höher ausgeführt werden. Dieses Framework umfasst Windows PowerShell. Weitere Informationen finden Sie unter Windows Management Framework 3.0.

Version	x86	x86_64	ARM64
2008[1]	✓	✓	
2008 R2[1]		✓	
2012 und 2012 R2		✓	
2016		✓	
2019		✓	
2022		✓	

[1] Ab dem 14. Januar 2020 wird Windows Server 2008 für Feature- oder Sicherheitsupdates von Microsoft nicht mehr unterstützt. Legacy Amazon Machine Images (AMIs) für Windows Server 2008 und 2008 R2 enthalten immer noch die Version 2 vom vorinstallierten SSM Agent, aber Systems Manager unterstützt offiziell nicht mehr die 2008-Versionen und aktualisiert den Agenten für diese

Versionen von Windows Server. Darüber hinaus ist SSM Agent Version 3 möglicherweise nicht mit allen Operationen auf Windows Server 2008 und 2008 R2 kompatibel. Die endgültige offiziell unterstützte Version von SSM Agent für Windows Server 2008 Versionen ist 2.3.1644.0.

Unterstützte Maschinentypen in Hybrid- und Multi-Cloud-Umgebungen

Systems Manager unterstützt eine Reihe von Maschinentypen als verwaltete Knoten. Ein verwalteter Knoten ist eine Maschine, die für die Arbeit mit Systems Manager konfiguriert ist.

In diesem Benutzerhandbuch werden die Begriffe Hybrid- und Multi-Cloud verwendet, um sich auf eine Umgebung zu beziehen, die eine beliebige Kombination der folgenden Maschinentypen enthält:

- Instances von Amazon Elastic Compute Cloud (Amazon EC2)
- Server in Ihren eigenen Räumlichkeiten (On-Premises-Server)
- AWS IoT Greengrass Kerngeräte
- AWS IoT und Geräte, die nicht zu den AWS Edge-Geräten gehören
- Virtuelle Maschinen (VMs), einschließlich VMs in anderen Cloud-Umgebungen

Informationen zur AWS Unterstützung von Hybrid- und Multicloud-Umgebungen finden Sie unter AWS Lösungen für Hybrid- und Multicloud-Umgebungen.

Systems Manager mit einem AWS SDK verwenden

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK for C++	AWS SDK for C++ Code-Beispiele
AWS CLI	AWS CLI Codebeispiele
AWS SDK for Go	AWS SDK for Go Codebeispiele
AWS SDK for Java	AWS SDK for Java Codebeispiele
AWS SDK for JavaScript	AWS SDK for JavaScript Codebeispiele

SDK-Dokumentation	Codebeispiele
AWS SDK for Kotlin	AWS SDK for Kotlin Codebeispiele
AWS SDK for .NET	AWS SDK for .NET Codebeispiele
AWS SDK for PHP	AWS SDK for PHP Codebeispiele
AWS Tools for PowerShell	Tools für PowerShell Codebeispiele
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) Codebeispiele
AWS SDK for Ruby	AWS SDK for Ruby Codebeispiele
AWS SDK for Rust	AWS SDK for Rust Codebeispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Codebeispiele
AWS SDK for Swift	AWS SDK for Swift Codebeispiele

Beispiel für die Verfügbarkeit

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link Provide feedback (Feedback geben) auswählen.

Einrichten AWS Systems Manager

Führen Sie die Aufgaben in diesem Abschnitt aus, um Rollen, Benutzerkonten, Berechtigungen und Erstressourcen für AWS Systems Manager einzurichten und zu konfigurieren. Die in diesem Abschnitt beschriebenen Aufgaben werden in der Regel von AWS-Konto Systemadministratoren ausgeführt. Nachdem Sie diese Schritte abgeschlossen sind, können Benutzer in Ihrer Organisation Systems Manager verwenden, um Ihre verwaltete Knoten zu konfigurieren, zu verwalten und darauf zuzugreifen. Ein verwalteter Knoten ist eine Maschine, die für die Verwendung mit Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung konfiguriert ist.

> **Note**
>
> Wenn Sie sowohl Amazon-EC2-Instances und Ihre eigenen Computing-Ressourcen in einer Hybrid- und Multi-Cloud-Umgebung verwenden möchten, führen Sie zunächst die in Systems Manager mit EC2-Instances verwenden beschriebenen Schritte aus. Dieses Thema stellt die Schritte in der optimalen Reihenfolge vor, um die Einrichtung von Systems Manager für EC2-Instances und Nicht-EC2-Geräte abzuschließen.

Wenn Sie bereits andere verwenden AWS-Services, haben Sie einige dieser Schritte abgeschlossen. Andere Schritte sind jedoch speziell auf den Systems Manager ausgelegt. Aus diesem Grund empfehlen wir, diesen gesamten Abschnitt zu lesen, um sicherzustellen, dass Sie die Voraussetzungen zur Nutzung aller Systems Manager-Funktionen erfüllt haben.

Themen

- Systems Manager mit EC2-Instances verwenden
- Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen
- Verwaltung von Edge-Geräten mit Systems Manager
- Einen AWS Organizations delegierten Administrator für Systems Manager erstellen
- Allgemeine Einrichtung für AWS Systems Manager

Systems Manager mit EC2-Instances verwenden

Führen Sie die Aufgaben in diesem Abschnitt aus, um Rollen, Berechtigungen und erste Ressourcen für AWS Systems Manager einzurichten und zu konfigurieren. Die in diesem Abschnitt beschriebenen

Aufgaben werden in der Regel von AWS-Konto und Systemadministratoren ausgeführt. Nachdem Sie diese Schritte abgeschlossen sind, können Benutzer in Ihrer Organisation Systems Manager verwenden, um Amazon Elastic Compute Cloud (Amazon EC2)-Instances in Ihrem Konto zu konfigurieren, zu verwalten und darauf zuzugreifen.

> **Note**
>
> Wenn Sie mit Systems Manager On-Premises-Maschinen verwalten und konfigurieren möchten, befolgen Sie die Einrichtungsschritte in Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen. Wenn Sie planen, sowohl Amazon EC2 Instances als auch Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung zu verwenden, befolgen Sie zunächst die hier aufgeführten Schritte. Dieser Abschnitt beschreibt die empfohlene Reihenfolge der Schritte zur Konfiguration der Rollen, Benutzer, Berechtigungen und ersten Ressourcen für Ihre Systems-Manager.

Wenn Sie bereits andere verwenden AWS-Services, haben Sie einige dieser Schritte abgeschlossen. Andere Schritte sind jedoch speziell auf den Systems Manager ausgelegt. Aus diesem Grund empfehlen wir, diesen gesamten Abschnitt zu lesen, um sicherzustellen, dass Sie die Voraussetzungen zur Nutzung aller Systems Manager-Funktionen erfüllt haben.

Inhalt

- Konfigurieren Sie die für Systems Manager erforderlichen Instanzberechtigungen
- Verbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager

Konfigurieren Sie die für Systems Manager erforderlichen Instanzberechtigungen

Hat standardmäßig AWS Systems Manager keine Berechtigung, Aktionen auf Ihren Instanzen auszuführen. Sie können Instanzberechtigungen auf Kontoebene mithilfe einer AWS Identity and Access Management (IAM-) Rolle oder auf Instanzebene mithilfe eines Instanzprofils gewähren. Wenn Ihr Anwendungsfall dies zulässt, empfehlen wir, mithilfe der Standardkonfiguration für die Host-Verwaltung Zugriff auf Kontoebene zu gewähren.

Empfohlene Konfiguration für EC2-Instanzberechtigungen

Die Standardkonfiguration für die Host-Verwaltung ermöglicht Systems Manager die automatische Verwaltung Ihrer Amazon-EC2-Instances. Nachdem Sie diese Einstellung aktiviert haben, werden alle Instances, die Instance Metadata Service Version 2 (IMDSv2) in der Version verwenden AWS-Region und AWS-Konto auf der SSM Agent Version 3.2.582.0 oder höher installiert ist, automatisch zu verwalteten Instanzen. Die Standardkonfiguration für die Host-Verwaltung unterstützt die Instance-Metadaten-Service-Version 1 nicht. Informationen zur Umstellung auf IMDSv2 finden Sie unter Umstellung auf die Verwendung von Instance Metadata Service Version 2 im Amazon EC2 EC2-Benutzerhandbuch. Informationen zur Überprüfung der auf Ihrer Instance installierten Version des SSM Agent finden Sie unter Überprüfen der SSM Agent-Versionsnummer. Informationen zur Aktualisierung des SSM Agent finden Sie unter Automatische Aktualisierung von SSM Agent. Zu den Vorteilen verwalteter Instances gehören die folgenden:

- Stellen Sie mit Session Manager eine sichere Verbindung zu Ihren Instances her.
- Führen Sie automatisierte Patch-Scans mit Patch Manager durch.
- Zeigen Sie mit Systems Manager Inventory detaillierte Informationen zu Ihren Instances an.
- Verfolgen und verwalten Sie Instances mithilfe von Fleet Manager.
- Halten Sie den SSM Agent automatisch auf dem neuesten Stand.

Fleet Manager, Inventar und Session Manager sind Patch Manager Funktionen von. AWS Systems Manager

Die Standardkonfiguration für die Host-Verwaltung ermöglicht die Instance-Verwaltung ohne die Verwendung von Instance-Profilen und stellt sicher, dass Systems Manager über Berechtigungen zum Verwalten aller Instances in der Region und im Konto verfügt. Wenn die bereitgestellten Berechtigungen für Ihren Anwendungsfall nicht ausreichen, können Sie auch Richtlinien zur Standard-IAM-Rolle hinzufügen, die von der Standardkonfiguration für die Host-Verwaltung erstellt wird. Wenn Sie keine Berechtigungen für alle Funktionen benötigen, die von der Standard-IAM-Rolle bereitgestellt werden, können Sie alternativ Ihre eigene benutzerdefinierte Rolle und Richtlinien erstellen. Alle Änderungen an der IAM-Rolle, die Sie für die Standardkonfiguration für die Host-Verwaltung auswählen, gelten für alle verwalteten Amazon-EC2-Instances in der Region und im Konto. Weitere Informationen über die von der Standardkonfiguration für die Host-Verwaltung verwendete Richtlinie finden Sie unter AWS verwaltete Richtlinie: InstanceDefault AmazonSSMManagedEC2-Richtlinie. Weitere Informationen zur Standard-Host-

Verwaltungskonfiguration finden Sie unter Verwenden der Standardeinstellung für die Host-Management-Konfiguration.

> **Important**
>
> Instances, die mit der Standardkonfiguration für die Host-Verwaltung registriert wurden, speichern Registrierungsinformationen lokal in den Verzeichnissen `/lib/amazon/ssm` oder `C:\ProgramData\Amazon`. Das Entfernen dieser Verzeichnisse oder der enthaltenen Dateien verhindert, dass die Instance die erforderlichen Anmeldeinformationen für die Verbindung mit Systems Manager über die Standardkonfiguration für die Host-Verwaltung erhält. In diesen Fällen müssen Sie ein Instance-Profil verwenden, um Ihrer Instance die erforderlichen Berechtigungen zu erteilen, oder die Instance neu erstellen.

> **Note**
>
> Dieses Verfahren sollte nur von Administratoren durchgeführt werden. Implementieren Sie den Zugriff mit den geringsten Berechtigungen, wenn Sie Einzelpersonen erlauben, die Standardkonfiguration für die Host-Verwaltung zu konfigurieren oder zu ändern. Sie müssen die Standard-Host-Management-Konfiguration in jeder Instanz aktivieren, in der AWS-Region Sie Ihre Amazon EC2 EC2-Instances automatisch verwalten möchten.

So aktivieren Sie die Einstellung der Standardkonfiguration für die Host-Verwaltung

Sie können die Standardkonfiguration für die Host-Verwaltung über die Fleet Manager-Konsole aktivieren. Um dieses Verfahren mit dem AWS Management Console oder Ihrem bevorzugten Befehlszeilentool erfolgreich abschließen zu können, benötigen Sie Berechtigungen für die API-Operationen GetServiceResetServiceSetting, UpdateService Setting und Setting. Darüber hinaus müssen Sie über Berechtigungen für die `iam:PassRole`-Berechtigung für die `AWSSystemsManagerDefaultEC2InstanceManagementRole`-IAM-Rolle verfügen. Es folgt eine Beispielrichtlinie . Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

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

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

Wenn Sie Instance-Profile an Ihre Amazon-EC2-Instances angefügt haben, entfernen Sie zunächst alle Berechtigungen, die diese `ssm:UpdateInstanceInformation`-Operation zulassen. Der SSM Agent versucht, die Instance-Profilberechtigungen zu verwenden, bevor die Berechtigungen der Standardkonfiguration für die Host-Verwaltung verwendet werden. Wenn Sie die `ssm:UpdateInstanceInformation`-Operation in Ihren Instance-Profilen zulassen, verwendet die Instance nicht die Berechtigungen der Standardkonfiguration für die Host-Verwaltung.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Drop-Down-Menü Kontoverwaltung die Option Standardkonfiguration für die Host-Verwaltung konfigurieren aus.
4. Aktivieren Sie Standardkonfiguration für die Host-Verwaltung aktivieren.

5. Wählen Sie die IAM-Rolle aus, die zum Aktivieren von Systems-Manager-Funktionen für Ihre Instances verwendet wird. Wir empfehlen die Verwendung der Standardrolle, die in der Standardkonfiguration für die Host-Verwaltung bereitgestellt wird. Sie enthält die Mindestberechtigungen für die Verwaltung Ihrer Amazon-EC2-Instances mit Systems Manager. Wenn Sie es vorziehen, eine benutzerdefinierte Rolle zu verwenden, muss die Vertrauensrichtlinie der Rolle Systems Manager als vertrauenswürdige Entität zulassen.
6. Wählen Sie Konfigurieren, um die Einrichtung abzuschließen.

Nach dem Aktivieren der Standardkonfiguration für die Host-Verwaltung kann es 30 Minuten dauern, bis Ihre Instances die Anmeldeinformationen der von Ihnen ausgewählten Rolle verwenden. Sie müssen die Standard-Host-Verwaltungskonfiguration in jeder Region aktivieren, in der Sie Ihre Amazon-EC2-Instances automatisch verwalten möchten.

Alternative Konfiguration für EC2-Instanzberechtigungen

Sie können Zugriff auf der Ebene der einzelnen Instances gewähren, indem Sie ein AWS Identity and Access Management (IAM) Instance-Profil verwenden. Ein Instance-Profil ist ein Container, der Informationen zur IAM-Rolle beim Start an eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance übergibt. Sie können ein Instance-Profil für Systems Manager erstellen, indem Sie eine oder mehrere IAM-Richtlinien anfügen, die die erforderlichen Berechtigungen für eine neue Rolle oder eine bereits von ihnen erstellte Rolle definieren.

> **Note**
>
> Sie könnenQuick Setup, eine Funktion von, verwenden AWS Systems Manager, um schnell ein Instanzprofil für alle Instances in Ihrem AWS-Konto zu konfigurieren. Quick Setuperstellt außerdem eine IAM-Servicerolle (oder übernimmt die Rolle), sodass Systems Manager Befehle auf Ihren Instances in Ihrem Namen sicher ausführen kann. Mit Quick Setup können Sie diesen Schritt (Schritt 3) und Schritt 4 überspringen. Weitere Informationen finden Sie unter AWS Systems Manager Quick Setup.

Beachten Sie die folgenden Details zum Erstellen eines IAM-Instance-Profils:

- Wenn Sie Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung für Systems Manager konfigurieren, müssen Sie für diese kein Instance-Profil erstellen. Konfigurieren Sie Ihre Server und VMs stattdessen zur Verwendung einer IAM-Servicerolle. Weitere Informationen finden Sie

unter Erstellen der für Systems Manager erforderlichen IAM-Servicerolle in Hybrid- und Multicloud-Umgebungen.

- Wenn Sie das IAM-Instance-Profil ändern, kann es einige Zeit dauern, bis die Instance-Anmeldeinformationen aktualisiert werden. SSM Agent verarbeitet Anfragen erst, wenn dies erfolgt ist. Um den Aktualisierungsprozess zu beschleunigen, können Sie SSM Agent oder die Instance erneut starten.

Verwenden Sie eine der folgenden Vorgehensweisen, je nachdem, ob Sie eine neue Rolle für Ihr Instance-Profil erstellen oder die erforderlichen Berechtigungen zu einer vorhandenen Rolle hinzufügen.

Erstellen eines Instance-Profils für von Systems Manager verwaltete Instances (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
3. Wählen Sie für Trusted entity type (Vertrauenswürdige Entität) die Option AWS-Service aus.
4. Wählen Sie direkt unter Use case (Anwendungsfall) die Option EC2 und dann Next (Weiter) aus.
5. Gehen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) wie folgt vor:

   - Verwenden Sie das Suchfeld, um die ManagedInstanceAmazonSSM Core-Richtlinie zu finden. Aktivieren Sie das Kontrollkästchen neben dem Namen.

     Die Konsole behält Ihre Auswahl auch dann bei, wenn Sie nach anderen Richtlinien suchen.

   - Wenn Sie im vorherigen Verfahren, (Optional) Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen S3-Bucket, eine benutzerdefinierte S3-Bucket-Richtlinie erstellt haben, suchen Sie danach und aktivieren Sie das Kontrollkästchen neben ihrem Namen.
   - Wenn Sie Instances einem Active Directory hinzufügen möchten, das von verwaltet wird AWS Directory Service, suchen Sie nach AmazonSSM DirectoryService Access und aktivieren Sie das Kontrollkästchen neben dem Namen.

- Wenn Sie planen, Ihre Instance mithilfe von EventBridge oder CloudWatch Logs zu verwalten oder zu überwachen, suchen Sie nach CloudWatchAgentServerPolicy und aktivieren Sie das Kontrollkästchen neben dem Namen.

6. Wählen Sie Weiter aus.
7. Geben Sie unter Role name (Rollenname) einen Namen für Ihr neues Instance-Profil ein, wie z. B. **`SSMInstanceProfile`**.

> **Note**
>
> Notieren Sie sich den Rollennamen. Sie wählen diese Rolle beim Erstellen neuer Instances, die Sie mit Systems Manager verwalten möchten.

8. (Optional) Aktualisieren Sie in Description (Beschreibung) die Beschreibung für dieses Instance-Profil ein.
9. (Optional) Fügen Sie für Tags ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, nachzuverfolgen oder zu steuern, und wählen Sie dann Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Roles (Rollen) zurück.

So fügen Sie Instance-Profil-Berechtigungen für Systems Manager zu einer vorhandenen Rolle hinzu (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich auf Rollen und wählen Sie die vorhandene Rolle aus, die Sie einem Instance-Profil für Systems Manager-Operationen zuordnen möchten.
3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Optionen Add permissions, Attach policies (Berechtigungen hinzufügen, Richtlinien anfügen) aus.
4. Führen Sie auf der Seite Attach Policy (Richtlinie anfügen) die folgenden Schritte aus:

   - Verwenden Sie das Suchfeld, um die AmazonSSM ManagedInstance Core-Richtlinie zu finden. Aktivieren Sie das Kontrollkästchen neben dem Namen.
   - Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie erstellt haben, suchen Sie danach und aktivieren Sie das Kontrollkästchen neben ihrem Namen. Weitere Informationen zu benutzerdefinierten S3-Bucket-Richtlinien für ein Instance-Profil finden Sie unter (Optional) Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen S3-Bucket.

- Wenn Sie Instances einem Active Directory hinzufügen möchten, das von verwaltet wird AWS Directory Service, suchen Sie nach AmazonSSM DirectoryService Access und aktivieren Sie das Kontrollkästchen neben dem Namen.
- Wenn Sie planen, Ihre Instance mithilfe von EventBridge oder CloudWatch Logs zu verwalten oder zu überwachen, suchen Sie nach CloudWatchAgentServerPolicy und aktivieren Sie das Kontrollkästchen neben dem Namen.

5. Wählen Sie Richtlinien anfügen.

Informationen zum Aktualisieren einer Rolle mit einer vertrauenswürdigen juristischen Stelle oder zur weiteren Einschränkung des Zugriffs finden Sie unter Ändern einer Rolle im IAM-Benutzerhandbuch.

(Optional) Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen S3-Bucket

Es muss nur dann eine benutzerdefinierte Richtlinie für Amazon S3-Zugriff erstellt werden, wenn Sie einen VPC-Endpunkt oder einen eigenen S3 Bucket in Ihren Systems Manager-Operationen verwenden. Sie können diese Richtlinie an die Standard-IAM-Rolle anfügen, die Sie mit der Standardkonfiguration für die Host-Verwaltung erstellt haben, oder an ein Instance-Profil, das Sie mit dem vorherigen Verfahren erstellt haben.

Informationen zu den AWS verwalteten S3-Buckets, auf die Sie in der folgenden Richtlinie Zugriff gewähren, finden Sie unterSSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets.

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create policy (Richtlinie erstellen).
3. Wählen Sie die Registerkarte JSON und ersetzen Sie den Standard-Text durch den folgenden.

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

  1                                                                    {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::aws-ssm-region/*",
                "arn:aws:s3:::aws-windows-downloads-region/*",
```

```
                "arn:aws:s3:::amazon-ssm-region/*",
                "arn:aws:s3:::amazon-ssm-packages-region/*",
                "arn:aws:s3:::region-birdwatcher-prod/*",
                "arn:aws:s3:::aws-ssm-distributor-file-region/*",
                "arn:aws:s3:::aws-ssm-document-attachments-region/*",
                "arn:aws:s3:::patch-baseline-snapshot-region/*"
            ]
        },

  2                                                                    {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",

 "s3:PutObjectAcl", 3

 "s3:GetEncryptionConfiguration" 4
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
                "arn:aws:s3:::DOC-EXAMPLE-
BUCKET" 5
            ]
        }
    ]
}
```

[1] Das erste `Statement`-Element ist nur erforderlich, wenn Sie einen VPC-Endpunkt verwenden.

[2] Das zweite `Statement`-Element ist nur erforderlich, wenn Sie einen S3 Bucket verwenden, den Sie zur Verwendung bei Ihren Systems Manager-Operationen erstellt haben.

[3] Die `PutObjectAcl`-ACL-Berechtigung ist nur erforderlich, wenn Sie vorhaben, kontoübergreifenden Zugriff auf S3-Buckets in anderen Konten zu unterstützen.

[4] Das `GetEncryptionConfiguration`-Element ist erforderlich, wenn Ihr S3-Bucket für die Verwendung der Verschlüsselung konfiguriert ist.

[5] Wenn Ihr S3 Bucket für die Verwendung der Verschlüsselung konfiguriert ist, muss das S3-Bucket-Stammverzeichnis (z. B. `arn:aws:s3:::DOC-EXAMPLE-BUCKET`) im Abschnitt Resource (Ressource) aufgeführt werden. Ihr Benutzer, Ihre Gruppe oder Ihre Rolle muss mit Zugriff auf den Stamm-Bucket konfiguriert sein.

4. Wenn Sie einen VPC-Endpunkt in Ihren Operationen verwenden, gehen Sie wie folgt vor:

   Ersetzen Sie im ersten `Statement`-Element jeden *`region`*-Platzhalter durch die ID der AWS-Region , in der diese Richtlinie verwendet wird. Verwenden Sie beispielsweise `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

   > **Important**
   >
   > Wir empfehlen, dass Sie keine Platzhalterzeichen (*) für bestimmte Regionen in dieser Richtlinie verwenden. Verwenden Sie beispielsweise `arn:aws:s3:::aws-ssm-us-east-2/*` und nicht `arn:aws:s3:::aws-ssm-*/*`. Bei der Verwendung von Platzhaltern könnte Zugriff auf S3-Buckets erteilt werden, für die Sie keinen Zugriff gewähren möchten. Wenn Sie das Instance-Profil für mehr als eine Region verwenden möchten, empfehlen wir, das erste `Statement`-Element für jede Region zu wiederholen.

   –oder–

   Wenn Sie in Ihren Operationen keinen VPC-Endpunkt verwenden, können Sie das erste `Statement`-Element löschen.

5. Wenn Sie einen eigenen S3-Bucket in Ihren Systems Manager-Operationen verwenden, gehen Sie wie folgt vor:

   Ersetzen Sie im zweiten `Statement`-Element *`DOC-EXAMPLE-BUCKET`* durch den Namen eines S3 Buckets in Ihrem Konto. Dieser Bucket wird nun für Ihre Systems Manager-Operationen verwendet. Er bietet die Berechtigung für Objekte im Bucket, wobei `"arn:aws:s3:::my-bucket-name/*"` als Ressource verwendet wird. Weitere Informationen über die Bereitstellung von Berechtigungen für Buckets oder Objekte in Buckets finden Sie im Thema Amazon-S3-Aktionen im Benutzerhandbuch zu Amazon Simple Storage Service und im AWS -Blog-Beitrag IAM Policies and Bucket Policies and ACLs! Oh, My! (Steuern des Zugriffs auf S3-Ressourcen).

> **Note**
>
> Wenn Sie mehr als einen Bucket verwenden, geben Sie den ARN für jeden davon an. Im folgenden Beispiel finden Sie Berechtigungen für Buckets.
>
> ```
> "Resource": [
> "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
> "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*"
>              ]
> ```

–oder–

Wenn Sie bei Ihren Systems Manager-Operationen keinen eigenen S3 Bucket verwenden, können Sie das zweite `Statement`-Element löschen.

6. Wählen Sie Weiter: Markierungen.
7. (Optional) Fügen Sie Tags hinzu, indem SieAdd tag (Tag hinzufügen) auswählen und die bevorzugten Tags für die Richtlinie eingeben.
8. Wählen Sie Weiter: Prüfen aus.
9. Geben Sie unter Name einen Namen für diese Richtlinie, z. B. **`SSMInstanceProfileS3Policy`**, ein.
10. Wählen Sie Richtlinie erstellen aus.

Zusätzliche Richtlinienüberlegungen für verwaltete Instances

In diesem Abschnitt werden einige der Richtlinien beschrieben, die Sie der Standard-IAM-Rolle hinzufügen können, die von der Standardkonfiguration für die Host-Verwaltung oder Ihren Instance-Profilen für AWS Systems Manager erstellt wurde. Um Berechtigungen für die Kommunikation zwischen Instances und der Systems-Manager-API zu erteilen, empfehlen wir, benutzerdefinierte Richtlinien zu erstellen, die Ihre System- und Sicherheitsanforderungen berücksichtigen. Abhängig von Ihrem Betriebsplan benötigen Sie möglicherweise Berechtigungen, die in einer oder mehreren der anderen Richtlinien dargestellt werden.

Richtlinie: **`AmazonSSMDirectoryServiceAccess`**

Nur erforderlich, wenn Sie vorhaben, Amazon-EC2-Instances für Windows Server einem Microsoft AD-Verzeichnis zuzuteilen.

Diese AWS verwaltete Richtlinie ermöglicht SSM Agent den Zugriff in Ihrem Namen AWS Directory Service auf Anfragen der verwalteten Instanz, der Domäne beizutreten. Weitere Informationen finden Sie unter Nahtloser Beitritt zu einer Windows-EC2-Instance im AWS Directory Service -Administratorhandbuch.

Richtlinie: **`CloudWatchAgentServerPolicy`**

Nur erforderlich, wenn Sie planen, den CloudWatch Agenten auf Ihren Instances zu installieren und auszuführen, um Metrik- und Protokolldaten auf einer Instance zu lesen und in Amazon zu schreiben CloudWatch. Diese helfen Ihnen dabei, Probleme oder Änderungen an Ihren AWS Ressourcen zu überwachen, zu analysieren und schnell darauf zu reagieren.

Ihre Standard-IAM-Rolle, die mit der Standard-Host-Management-Konfiguration oder dem Instance-Profil erstellt wurde, benötigt diese Richtlinie nur, wenn Sie Funktionen wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden. (Sie können auch eine restriktivere Richtlinie erstellen, die beispielsweise den Schreibzugriff auf einen bestimmten CloudWatch Logs-Protokollstream einschränkt.)

> **Note**
>
> Die Verwendung der Funktionen EventBridge und CloudWatch Protokollierung ist optional. Wenn Sie sich jedoch hierzu entschlossen haben, sollte diese zu Beginn des Systems-Manager-Konfigurationsprozesses eingerichtet werden. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch und im Amazon CloudWatch Logs-Benutzerhandbuch.

Informationen zum Erstellen von IAM-Richtlinien mit Berechtigungen für zusätzliche Systems-Manager-Funktionen finden Sie in den folgenden Ressourcen:

- Einschränken des Zugriffs auf Systems Manager-Parameter mithilfe von IAM-Richtlinien
- Einrichten der Automatisierung
- Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager

Anfügen des Systems-Manager-Instance-Profils an eine Instance (Konsole)

1. Melden Sie sich bei der Amazon EC2 EC2-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Instances.
3. Navigieren Sie in der Liste zu Ihrer EC2-Instance und wählen Sie sie aus.
4. Wählen Sie im Menü Actions (Aktionen) die Option Security (Sicherheit), Modify IAM role (IAM-Rolle ändern).
5. Wählen Sie für die IAM-Rolle das Instance-Profil aus, das Sie mithilfe des Verfahrens in Alternative Konfiguration für EC2-Instanzberechtigungen erstellt haben.
6. Wählen Sie Aktualisieren der IAM-Rolle.

Für weitere Informationen zum Anhängen von IAM-Rollen an Instances, wählen Sie, je nach Ihrem ausgewählten Betriebssystem, einen der folgenden Schritte aus:

- Fügen Sie einer Instance im Amazon EC2 EC2-Benutzerhandbuch eine IAM-Rolle hinzu
- Fügen Sie einer Instance im Amazon EC2 EC2-Benutzerhandbuch eine IAM-Rolle hinzu

Fahren Sie fort mit Verbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

Verbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager

Sie können die Sicherheitslage Ihrer verwalteten Knoten (einschließlich Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung) verbessern, indem Sie die Verwendung eines VPC-Schnittstellen-Endpunkts in Amazon Virtual Private Cloud (Amazon VPC) konfigurieren AWS Systems Manager . Mithilfe eines Schnittstellen-VPC-Endpunkts (Schnittstellenendpunkt) können Sie eine Verbindung zu Diensten herstellen, die von bereitgestellt werden AWS PrivateLink. AWS PrivateLink ist eine Technologie, mit der Sie privat auf Amazon Elastic Compute Cloud (Amazon EC2) und Systems Manager Manager-APIs zugreifen können, indem Sie private IP-Adressen verwenden.

AWS PrivateLink schränkt den gesamten Netzwerkverkehr zwischen Ihren verwalteten Instances, Systems Manager und Amazon EC2 auf das Amazon-Netzwerk ein. Dies bedeutet, dass Ihre

verwalteten Instances keinen Zugriff auf das Internet haben. Wenn Sie verwenden AWS PrivateLink, benötigen Sie kein Internet-Gateway, kein NAT-Gerät oder ein virtuelles privates Gateway.

Eine Konfiguration ist nicht erforderlich AWS PrivateLink, wird aber empfohlen. Weitere Informationen zu AWS PrivateLink VPC-Endpunkten finden Sie unter AWS PrivateLink und VPC-Endpoints.

> **Note**
>
> Die Alternative zur Verwendung eines VPC-Endpunkts ist das Aktivieren von ausgehendem Internetzugriff auf Ihren verwalteten Instances. In diesem Fall müssen die verwalteten Instances auch ausgehenden HTTPS-Datenverkehr (Port 443) zu den folgenden Endpunkten zulassen:
>
> - `ssm.region.amazonaws.com`
> - `ssmmessages.region.amazonaws.com`
> - `ec2messages.region.amazonaws.com`
>
> SSM Agent initiiert alle Verbindungen zum Systems Manager-Service in der Cloud. Aus diesem Grund müssen Sie Ihre Firewall nicht so konfigurieren, dass eingehender Datenverkehr zu Ihren Instances für Systems Manager zugelassen wird. Weitere Informationen über Anrufe an diese Endpunkte finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.

Über Amazon VPC

Sie können Amazon Virtual Private Cloud (Amazon VPC) verwenden, um ein virtuelles Netzwerk in Ihrem eigenen logisch isolierten Bereich innerhalb der AWS Cloud sogenannten Virtual Private Cloud (VPC) zu definieren. Sie können Ihre AWS -Ressourcen, z. B. Instances, in Ihrer VPC launchen. Eine VPC ist einem herkömmlichen Netzwerk in einem eigenen Rechenzentrum sehr ähnlich, bietet jedoch die Vorteile durch die Nutzung der skalierbaren Infrastruktur von AWS. Sie können Ihre VPC konfigurieren. Hierzu können Sie den IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen, Netzwerk-Gateways und Sicherheitseinstellungen konfigurieren. Sie können jetzt Instances in der VPC mit dem Internet verbinden. Sie können Ihre VPC mit Ihrem eigenen Unternehmensrechenzentrum verbinden und sie so zu AWS Cloud einer Erweiterung Ihres Rechenzentrums machen. Um die Ressourcen in den einzelnen Subnetzen zu schützen, können Sie mehrere Sicherheitsebenen verwenden, darunter Sicherheitsgruppen und Netzwerk-Zugriffskontrolllisten. Weitere Informationen finden Sie im Amazon-VPC-Benutzerhandbuch.

Themen

- Beschränkungen und Einschränkungen bei VPC-Endpunkten
- Erstellen von VPC-Endpunkten für Systems Manager
- Erstellen einer Schnittstellen-VPC-Endpunkt-Richtlinie

Beschränkungen und Einschränkungen bei VPC-Endpunkten

Seien Sie sich der folgenden Einschränkungen bewusst, bevor Sie VPC-Endpunkte für Systems Manager konfigurieren.

Regionsübergreifende Anforderungen

VPC-Endpunkte unterstützen keine regionsübergreifenden Anfragen. Stellen Sie sicher, dass Sie Ihren Endpunkt im selben Bucket erstellen. AWS-Region Die Region Ihres Buckets können Sie auf der Amazon S3-Konsole oder mit dem Befehl get-bucket-location abrufen. Verwenden Sie einen regionsspezifischen Amazon S3-Endpunkt, um auf Ihren Bucket zuzugreifen, z. B, `DOC-EXAMPLE-BUCKET.s3-us-west-2.amazonaws.com`. Weitere Informationen über regionsspezifische Endpunkte für Amazon S3 finden Sie unter Amazon-S3-Endpunkte im Allgemeine Amazon Web Services-Referenz. Wenn Sie das verwenden, AWS CLI um Anfragen an Amazon S3 zu stellen, setzen Sie Ihre Standardregion auf dieselbe Region wie Ihr Bucket oder verwenden Sie den `--region` Parameter in Ihren Anfragen.

VPC-Peering-Verbindungen

Der Zugriff auf VPC-Schnittstellenendpunkte erfolgt sowohl über intraregionale als auch interregionale VPC-Peering-Verbindungen. Weitere Informationen zu VPC-Peering-Verbindungsanforderungen für VPC-Schnittstellenendpunkte finden Sie unter VPC-Peering-Verbindungen (Kontingente) im Benutzerhandbuch zu Amazon Virtual Private Cloud.

VPC-Gateway-Endpunktverbindungen können nicht auf einen Bereich außerhalb einer VPC erweitert werden. Ressourcen am anderen Ende einer VPC-Peering-Verbindung in Ihrer VPC können nicht über den Gateway-Endpunkt mit Ressourcen im Gateway-Endpunktservice kommunizieren. Weitere Informationen zu VPC-Peering-Verbindungsanforderungen für VPC-Gateway-Endpunkte finden Sie unter VPC-Endpunkte (Kontingente) im Benutzerhandbuch zu Amazon Virtual Private Cloud

Eingehende Verbindungen

Die Sicherheitsgruppe für den VPC-Endpunkt müssen eingehende Verbindungen auf Port 443 aus dem privaten Subnetz der verwalteten Instance zulassen. Wenn keine eingehenden Verbindungen

zulässig sind, kann die verwaltete Instance keine Verbindung mit den SSM- und EC2-Endpunkten einrichten.

DNS-Auflösung

Wenn Sie einen benutzerdefinierten DNS-Server verwenden, müssen Sie dem Amazon-DNS-Server für Ihre VPC einen bedingten Weiterleiter für alle Abfragen an die `amazonaws.com`-Domain hinzufügen.

S3-Buckets

Ihre VPC-Endpunktrichtlinie muss mindestens Zugriff auf die folgenden Amazon S3-Buckets gewähren:

- Die in SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets aufgeführten S3-Buckets.
- Die S3-Buckets, die vom Patch Manager für Patch-Basisoperationen in Ihrer AWS-Region verwendet werden. Diese Buckets enthalten den Code, der vom Patch-Baseline-Service abgerufen und auf Instances und ausgeführt wird. Jeder AWS-Region hat seine eigenen Patch-Baseline-Operations-Buckets, aus denen der Code abgerufen wird, wenn ein Patch-Baseline-Dokument ausgeführt wird. Wenn der Code nicht heruntergeladen werden kann, schlägt der Patch-Baseline-Befehl fehl.

  > **Note**
  >
  > Wenn Sie eine On-Premises-Firewall verwenden und Patch Manager nutzen möchten, muss diese Firewall auch den Zugriff auf den geeigneten Patch-Baseline-Endpunkt zulassen.

  Um Zugriff auf die Buckets in Ihrem zu gewähren AWS-Region, nehmen Sie die folgende Berechtigung in Ihre Endpunktrichtlinie auf.

  ```
  arn:aws:s3:::patch-baseline-snapshot-region/*
  arn:aws:s3:::aws-ssm-region/*
  ```

  `region` steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

Sehen Sie sich das folgende -Beispiel an.

```
arn:aws:s3:::patch-baseline-snapshot-us-east-2/*
arn:aws:s3:::aws-ssm-us-east-2/*
```

> **Note**
>
> Nur in der Region Naher Osten (Bahrain) (me-south-1) werden für diese Buckets andere Namenskonventionen verwendet. Verwenden Sie stattdessen AWS-Region nur für diesen Zweck die folgenden zwei Buckets:
>
> - `patch-baseline-snapshot-me-south-1-uduvl7q8`
> - `aws-patch-manager-me-south-1-a53fc9dce`

CloudWatch Amazon-Protokolle

Wenn Sie Ihren Instances nicht erlauben, auf das Internet zuzugreifen, erstellen Sie einen VPC-Endpunkt für CloudWatch Logs, um Funktionen zu verwenden, die Logs an CloudWatch Logs senden. Weitere Informationen zum Erstellen eines Endpunkts für CloudWatch Logs finden Sie unter Creating a VPC Endpoint for CloudWatch Logs im Amazon CloudWatch Logs-Benutzerhandbuch.

DNS in einer Hybrid- und Multi-Cloud-Umgebung

Informationen zur Konfiguration von DNS für die Verwendung mit AWS PrivateLink Endpunkten in Hybrid- und Multicloud-Umgebungen finden Sie unter Private DNS für Schnittstellenendpunkte im Amazon VPC-Benutzerhandbuch. Wenn Sie Ihre eigene DNS verwenden möchten, können Sie Route 53-Resolver nutzen. Weitere Informationen finden Sie unter Auflösen von DNS-Abfragen zwischen VPCs und Ihrem Netzwerk im Entwicklerhandbuch zu Amazon Route 53.

Erstellen von VPC-Endpunkten für Systems Manager

Verwenden Sie die folgenden Informationen, um VPC-Schnittstelle und Gateway-Endpunkte für AWS Systems Manager zu erstellen. Dieses Thema verweist auf Verfahren im Amazon VPC User Guide.

Erstellen von VPC-Endpunkten für Systems Manager

Im ersten Schritt dieses Verfahrens erstellen Sie drei erforderliche und einen optionalen Schnittstellen-Endpunkt für Systems Manager. Alle drei Endpunkte sind erforderlich, damit Systems

Manager in einer VPC funktioniert. Der vierte, `com.amazonaws.region.ssmmessages`, ist nur erforderlich, wenn Sie Session Manager-Funktionen verwenden.

Im zweiten Schritt erstellen Sie den erforderlichen Gateway-Endpunkt für Systems Manager für den Zugriff auf Amazon S3.

> **Note**
>
> *region* steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

1. Befolgen Sie die Schritte unter Erstellen eines Schnittstellen-Endpunkts zum Erstellen der folgenden Schnittstellen-Endpunkte:

   - **com.amazonaws.*region*.ssm** – Der Endpunkt für den Systems-Manager-Service.
   - **com.amazonaws.*region*.ec2messages** – Systems Manager verwendet diesen Endpunkt, um Anrufe vom SSM Agent an den Systems-Manager-Service zu tätigen.
   - **com.amazonaws.*region*.ec2**: Wenn Sie mithilfe von Systems Manager VSS-fähige Snapshots erstellen, müssen Sie sicherstellen, dass Sie über einen Endpunkt für den EC2-Service verfügen. Ohne den definierten EC2-Endpunkt schlägt der Aufruf zur Auflistung angefügter Amazon-EBS-Volumes fehl. Dies führt dazu, dass der Systems-Manager-Befehl fehlschlägt.
   - **com.amazonaws.*region*.ssmmessages** – Dieser Endpunkt ist nur erforderlich, wenn Sie die Verbindung zu Ihren Instances über einen sicheren Datenkanal mit Session Manager herstellen. Weitere Informationen finden Sie unter AWS Systems Manager Session Manager und Referenz: ec2messages, ssmmessages und andere API-Operationen.
   - **com.amazonaws.*region*.kms** – Dieser Endpunkt ist optional. Sie kann jedoch erstellt werden, wenn Sie die AWS Key Management Service (AWS KMS) -Verschlüsselung für Session Manager Parameter Store Parameter verwenden möchten.
   - **com.amazonaws.*region*.logs** – Dieser Endpunkt ist optional. Es kann jedoch erstellt werden, wenn Sie Amazon CloudWatch Logs (CloudWatch Logs) für Session ManagerRun Command, oder SSM Agent Logs verwenden möchten.
2. Führen Sie die Schritte unter Erstellen eines Gateway-Endpunkts aus, um den folgenden Gateway-Endpunkt für Amazon S3 zu erstellen.

- **`com.amazonaws.region.s3`** – Systems Manager verwendet diesen Endpunkt zum Aktualisieren von SSM Agent und zum Durchführen von Patching-Vorgängen. Systems Manager verwendet diesen Endpunkt für Aufgaben wie das Hochladen von Ausgabeprotokollen, die Sie in S3 Buckets speichern wollen, das Abrufen von Skripten oder anderen Dateien, die Sie in Buckets speichern, und so weiter. Wenn die mit Ihren Instances verknüpfte Sicherheitsgruppe den ausgehenden Datenverkehr einschränkt, müssen Sie eine Regel hinzufügen, um Datenverkehr zur Präfixliste für Amazon S3 zuzulassen. Weitere Informationen finden Sie unter Modify your security group im AWS PrivateLink -Guide.

Informationen zu den AWS verwalteten S3-Buckets, auf die zugegriffen werden SSM Agent muss, finden Sie unterSSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets. Wenn Sie in Ihren Systems-Manager-Operationen einen Virtual Private Cloud (VPC)-Endpunkt verwenden, müssen Sie in einem EC2-Instance-Profil für Systems Manager oder in einer Servicerolle für Nicht-EC2-verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung eine explizite Berechtigung gewähren.

Erstellen einer Schnittstellen-VPC-Endpunkt-Richtlinie

Sie können Richtlinien für VPC-Schnittstellen-Endpoints erstellen, für AWS Systems Manager die Sie Folgendes angeben können:

- Der Prinzipal, der die Aktionen ausführen kann
- Aktionen, die ausgeführt werden können
- Ressourcen, für die Aktionen ausgeführt werden können

Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Benutzerhandbuch zu Amazon VPC.

Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen

Sie können AWS Systems Manager damit sowohl Amazon Elastic Compute Cloud (EC2) -Instances als auch eine Reihe von Nicht-EC2-Maschinentypen verwalten. Dieser Abschnitt beschreibt die Einrichtungsaufgaben, die Konto- und Systemadministratoren ausführen, um Nicht-EC2-Maschinen

mithilfe von Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung zu verwalten. Nach Abschluss dieser Schritte können Benutzer, denen der AWS-Konto Administrator Berechtigungen erteilt hat, Systems Manager verwenden, um die Nicht-EC2-Computer ihrer Organisation zu konfigurieren und zu verwalten.

Jede Maschine, die für die Verwendung mit Systems Manager konfiguriert wurde, wird als verwalteter Knoten bezeichnet.

> **Note**
>
> - Sie können Edge-Geräte als verwaltete Knoten registrieren, indem Sie die gleichen Hybrid-Aktivierungsschritte wie für andere Nicht-EC2-Maschinen verwenden. Zu diesen Arten von Edge-Geräten gehören sowohl Geräte als auch AWS IoT Geräte, bei denen es sich nicht AWS IoT um Geräte handelt. Verwenden Sie den in diesem Abschnitt beschriebenen Prozess, um diese Arten von Edge-Geräten einzurichten.
>
>   Systems Manager unterstützt auch Edge-Geräte, die AWS IoT Greengrass Core-Software verwenden. Der Einrichtungsprozess und die Anforderungen für AWS IoT Greengrass Core-Geräte unterscheiden sich von denen für Edge-Geräte AWS IoT und Edge-Geräte, bei denen es sich nicht um AWS Edge-Geräte handelt. Informationen zur Registrierung von AWS IoT Greengrass Geräten für die Verwendung mit Systems Manager finden Sie unterVerwaltung von Edge-Geräten mit Systems Manager.
> - Nicht-EC2-macOS-Maschinen werden für Hybrid- und Multi-Cloud-Umgebungen von Systems Manager nicht unterstützt.

Wenn Sie Systems Manager verwenden möchten, um Amazon Elastic Compute Cloud (Amazon EC2)-Instances bzw. sowohl Amazon-EC2-Instances und Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung zu verwalten, befolgen Sie zunächst die Schritte unter Systems Manager mit EC2-Instances verwenden.

Nachdem Sie Ihre Hybrid- und Multi-Cloud-Umgebung für Systems Manager konfiguriert haben, können Sie Folgendes tun:

- Erstellen Sie eine konsistente und sichere Möglichkeit, Hybrid- und Multi-Cloud-Workloads mit denselben Tools oder Skripts von einem Remote-Standort aus zu verwalten.
- Zentralisieren Sie die Zugriffskontrolle für Aktionen, die auf Ihren Computern mithilfe von AWS Identity and Access Management (IAM) ausgeführt werden können.

- Zentralisieren Sie die Überwachung der auf Ihren Maschinen durchgeführten Vorgänge, indem Sie die in AWS CloudTrail aufgezeichnete API-Aktivität anzeigen.

  Informationen CloudTrail zur Überwachung von Systems Manager Manager-Aktionen finden Sie unterAWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.
- Zentralisieren Sie die Überwachung, indem Sie Amazon EventBridge und Amazon Simple Notification Service (Amazon SNS) so konfigurieren, dass Benachrichtigungen über die erfolgreiche Ausführung des Dienstes gesendet werden.

  Informationen EventBridge zur Überwachung von Systems Manager Manager-Ereignissen finden Sie unterÜberwachung von Systems Manager-Ereignissen mit Amazon EventBridge.

Informationen zu verwalteten Knoten

Nachdem Sie die Konfiguration Ihrer Nicht-EC2-Computer für Systems Manager wie in diesem Abschnitt beschrieben abgeschlossen haben, werden Ihre hybridaktivierten Maschinen in den Knoten aufgeführt AWS Management Console und als verwaltete Knoten beschrieben. In der Konsole werden die IDs Ihrer hybrid-aktivierten verwalteten Knoten allerdings mit dem Präfix „mi-“ von Amazon-EC2-Instances unterschieden. Amazon-EC2-Instance-IDs verwenden das Präfix „i-“.

Eine verwalteter Knoten ist jede für Systems Manager konfigurierte Maschine. Bisher wurden alle verwalteten Knoten als verwaltete Instances bezeichnet. Der Begriff Instance bezieht sich jetzt nur noch auf EC2-Instances. Der Befehl deregister-managed-instance wurde vor dieser Terminologieänderung benannt.

Weitere Informationen finden Sie unter Mit verwalteten Knoten arbeiten.

Informationen zum Instance-Kontingent

Systems Manager bietet ein Standard-Instances-Kontingent und ein Advanced-Instances-Kontingent für Nicht-EC2-verwaltete Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung an. Mit dem Standard-Instances-Kontingent erlaubt Ihnen maximal 1 000 hybridaktivierte Maschinen pro AWS-Konto pro AWS-Region zu registrieren. Wenn Sie mehr als 1 000 Nicht-EC2-Maschinen in einem einzigen Konto und einer Region anmelden müssen, verwenden Sie das Advanced-Instances-Kontingent. Mit erweiterten Instances können Sie auch eine Verbindung zu Ihren Nicht-EC2-Computern herstellen, indem Sie AWS Systems Manager Session Manager Session Managerbietet interaktiven Shell-Zugriff auf Ihre verwalteten Knoten.

Weitere Informationen finden Sie unter Konfigurieren von Instance-Kontingenten.

Themen

- Erstellen Sie die für Systems Manager in Hybrid- und Multicloud-Umgebungen erforderliche IAM-Servicerolle
- Erstellen Sie eine Hybridaktivierung, um Knoten bei Systems Manager zu registrieren
- So installieren Sie das SSM Agent auf Hybrid-Linux-Knoten
- So installieren Sie den SSM Agent auf Windows Hybridknoten

Erstellen Sie die für Systems Manager in Hybrid- und Multicloud-Umgebungen erforderliche IAM-Servicerolle

Nicht-EC2-Maschinen (Amazon Elastic Compute Cloud) in einer Hybrid- und Multi-Cloud-Umgebung benötigen eine AWS Identity and Access Management (IAM) -Servicerolle, um mit dem Service zu kommunizieren. AWS Systems Manager Die Rolle gewährt AWS Security Token Service (AWS STS) `AssumeRole` Zugriff auf den Systems Manager-Dienst. Sie müssen nur einmal eine Servicerolle für eine Hybrid- und Multi-Cloud-Umgebung erstellen, und zwar für jedes AWS-Konto. Sie können jedoch mehrere Servicerollen für verschiedene Hybrid- und Multi-Cloud-Aktivierungen erstellen, wenn Maschinen in Ihrer Hybrid-Umgebung unterschiedliche Berechtigungen benötigen.

Im Folgenden wird beschrieben, wie Sie die erforderliche Servicerolle mit der Systems-Manager-Konsole oder Ihrem bevorzugten Befehlszeilen-Tool erstellen.

Verwenden der AWS Management Console , um eine IAM-Dienstrolle für Systems Manager Manager-Hybridaktivierungen zu erstellen

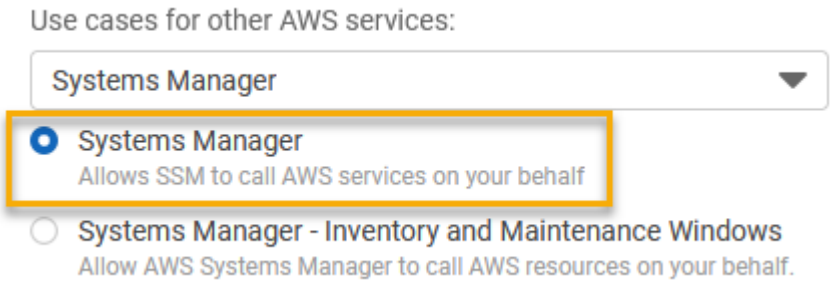
Verwenden Sie das folgende Verfahren zum Erstellen einer Servicerolle für eine Hybrid-Aktivierung. Dieses Verfahren verwendet die `AmazonSSMManagedInstanceCore`-Richtlinie für die Kernfunktionalität von Systems Manager. Abhängig von Ihrem Anwendungsfall müssen Sie Ihrer Servicerolle möglicherweise zusätzliche Richtlinien hinzufügen, damit Ihre Rechner On-Premises auf andere Funktionen oder AWS-Services zugreifen können. Zum Beispiel, ohne Zugriff auf das erforderliche AWS -verwalteten Amazon Simple Storage Service (Amazon S3)-Buckets schlagen Patch Manager-Patch-Operationen fehl.

So erstellen Sie eine -Servicerolle (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).

3. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die folgenden Optionen:

   1. Wählen Sie für Vertrauenswürdige Entität die Option AWS-Service aus.
   2. Wählen Sie Systems Manager für Anwendungsfälle für andere AWS-Services.
   3. Wählen Sie Systems Manager, wie im folgenden Image gezeigt.



4. Wählen Sie Next (Weiter).
5. Gehen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) wie folgt vor:

   - Verwenden Sie das Suchfeld, um die AmazonSSM ManagedInstance Core-Richtlinie zu finden. Aktivieren Sie das Kontrollkästchen neben dem Namen.

   - Die Konsole behält Ihre Auswahl auch dann bei, wenn Sie nach anderen Richtlinien suchen.
   - Wenn Sie im Verfahren (Optional) Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen S3-Bucket, eine benutzerdefinierte S3-Bucket-Richtlinie erstellt haben, suchen Sie danach und aktivieren Sie das Kontrollkästchen neben ihrem Namen.
   - Wenn Sie beabsichtigen, Nicht-EC2-Computer zu einem Active Directory hinzuzufügen, das von verwaltet wird AWS Directory Service, suchen Sie nach AmazonSSM DirectoryService Access und aktivieren Sie das Kontrollkästchen neben dem Namen.
   - Wenn Sie planen, Ihren verwalteten Knoten mithilfe von EventBridge oder CloudWatch Logs zu verwalten oder zu überwachen, suchen Sie nach CloudWatchAgentServerPolicy und aktivieren Sie das Kontrollkästchen neben dem Namen.

6. Wählen Sie Weiter aus.

7. Geben Sie unter Rollenname einen Namen für Ihre neue IAM-Serverrolle ein, z. B. **`SSMServerRole`**.

   > **Note**
   >
   > Notieren Sie sich den Rollennamen. Sie wählen diese Rolle, wenn Sie neue Maschinen registrieren, die Sie mit Systems Manager verwalten möchten.

8. (Optional) Aktualisieren Sie für Beschreibung die Beschreibung für diese IAM-Serverrolle.
9. (Optional) Fügen Sie für Tags ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, nachzuverfolgen oder zu steuern.
10. Wählen Sie Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Roles (Rollen) zurück.

Verwenden der AWS CLI , um eine IAM-Dienstrolle für Systems Manager Manager-Hybridaktivierungen zu erstellen

Verwenden Sie das folgende Verfahren zum Erstellen einer Servicerolle für eine Hybrid-Aktivierung. Dieses Verfahren verwendet die `AmazonSSMManagedInstanceCore`-Richtlinie für die Kernfunktionalität von Systems Manager. Je nach Anwendungsfall müssen Sie Ihrer Servicerolle für Ihre Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung möglicherweise zusätzliche Richtlinien hinzufügen, um auf andere Funktionen oder AWS-Services zugreifen zu können.

S3-Bucket-Richtlinienanforderung

Wenn einer der folgenden Fälle zutrifft, müssen Sie eine benutzerdefinierte IAM-Berechtigungsrichtlinie für Amazon Simple Storage Service (Amazon S3)-Buckets erstellen, bevor Sie dieses Verfahren durchführen:

- Fall 1 — Sie verwenden einen VPC-Endpunkt, um Ihre VPC privat mit unterstützten AWS-Services und VPC-Endpunktdiensten zu verbinden, die von unterstützt werden. AWS PrivateLink
- Fall 2: Sie beabsichtigen, einen Amazon-äS3-Bucket zu verwenden, den Sie als Teil Ihrer Systems-Manager-Operationen erstellen, z. B. zum Speichern der Ausgabe für Run Command-Befehle oder Session Manager-Sitzungen in einem S3-Bucket. Bevor Sie fortfahren, befolgen Sie die Schritte unter Erstellen einer benutzerdefinierten S3-Bucket-Richtlinie für ein Instance-Profil. Die Informationen über S3-Bucket-Richtlinien in diesem Thema gelten auch für Ihre Service-Rolle.

AWS CLI

So erstellen Sie eine IAM-Servicerolle für eine Hybrid- und Multi-Cloud-Umgebung (AWS CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Erstellen Sie eine Textdatei mit einem Namen wie z. B. `SSMService-Trust.json` mit der folgenden Vertrauensrichtlinie auf Ihrer lokalen Maschine. Stellen Sie sicher, dass Sie die Datei mit der Erweiterung `.json` speichern. Stellen Sie sicher, dass Sie Ihre AWS-Konto und die AWS-Region in der ARN angeben, in der Sie Ihre Hybrid-Aktivierung erstellt haben.

```
{
   "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. Öffnen Sie das und führen Sie in dem Verzeichnis AWS CLI, in dem Sie die JSON-Datei erstellt haben, den Befehl create-role aus, um die Servicerolle zu erstellen. In diesem Beispiel wird eine Rolle mit dem Namen `SSMServiceRole` erstellt. Sie können auf Wunsch einen anderen Namen wählen.

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. Führen Sie den attach-role-policy-Befehl wie folgt, um es der gerade von Ihnen erstellten Servicerolle zu ermöglichen, ein Sitzungs-Token zu erstellen. Das Sitzungstoken erteilt Ihrem verwalteten Knoten die Berechtigung, Befehle mit Systems Manager auszuführen.

> **Note**
>
> Die Richtlinien, die Sie für ein Serviceprofil für verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung hinzufügen, sind die gleichen Richtlinien, die zum Erstellen eines Instance-Profils für Amazon Elastic Compute Cloud (Amazon EC2)-Instances verwendet werden. Weitere Informationen zu den in den folgenden Befehlen verwendeten AWS Richtlinien finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

(Erforderlich) Führen Sie den folgenden Befehl aus, damit ein verwalteter Knoten die Kernfunktionen des AWS Systems Manager Service nutzen kann.

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
```

Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie für Ihre Servicerolle erstellt haben, führen Sie den folgenden Befehl aus, damit AWS Systems Manager Agent (SSM Agent) auf die Buckets zugreifen kann, die Sie in der Richtlinie angegeben haben. Ersetzen Sie *account-id* und *DOC-EXAMPLE-BUCKET durch Ihre ID und Ihren Bucket-Namen*. AWS-Konto

Linux & macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::account-id:policy/DOC-EXAMPLE-BUCKET
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::account-id:policy/DOC-EXAMPLE-BUCKET
```

(Optional) Führen Sie den folgenden Befehl aus, um in Ihrem Namen Zugriff AWS Directory Service auf Anfragen zum Beitritt SSM Agent zur Domäne durch den verwalteten Knoten zu gewähren. Ihre Servicerolle benötigt diese Richtlinie nur, wenn Sie Ihre Knoten mit einem Microsoft-AD-Verzeichnis verbinden.

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
```

(Optional) Führen Sie den folgenden Befehl aus, damit der CloudWatch Agent auf Ihren verwalteten Knoten ausgeführt werden kann. Dieser Befehl ermöglicht es, Informationen auf einem Knoten zu lesen und darauf zu schreiben CloudWatch. Ihr Serviceprofil benötigt diese Richtlinie nur, wenn Sie Dienste wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden.

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

Tools for PowerShell

So erstellen Sie eine IAM-Servicerolle für eine Hybrid- und Multi-Cloud-Umgebung (AWS Tools for Windows PowerShell)

1. Installieren und konfigurieren Sie die AWS Tools for PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS Tools for PowerShell.

2. Erstellen Sie eine Textdatei mit einem Namen wie z. B. `SSMService-Trust.json` mit der folgenden Vertrauensrichtlinie auf Ihrer lokalen Maschine. Stellen Sie sicher, dass Sie die Datei mit der Erweiterung `.json` speichern. Stellen Sie sicher, dass Sie Ihre AWS-Konto und die AWS-Region in der ARN angeben, in der Sie Ihre Hybrid-Aktivierung erstellt haben.

```
{
   "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:region:123456789012:*"
            }
         }
      }
   ]
}
```

3. Öffnen Sie PowerShell im Administratormodus und führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, New-IAMRole wie folgt aus, um eine Servicerolle zu erstellen. In diesem Beispiel wird eine Rolle mit dem Namen `SSMServiceRole` erstellt. Sie können auf Wunsch einen anderen Namen wählen.

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

4. Verwenden Sie Register-IAM RolePolicy wie folgt, damit die von Ihnen erstellte Servicerolle ein Sitzungstoken erstellen kann. Das Sitzungstoken erteilt Ihrem verwalteten Knoten die Berechtigung, Befehle mit Systems Manager auszuführen.

> **Note**
>
> Die Richtlinien, die Sie für ein Serviceprofil für verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung hinzufügen, sind dieselben Richtlinien, die zum Erstellen eines Instance-Profils für EC2-Instances verwendet werden. Weitere Informationen zu den in den folgenden Befehlen verwendeten AWS Richtlinien finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

(Erforderlich) Führen Sie den folgenden Befehl aus, damit ein verwalteter Knoten die Kernfunktionen des AWS Systems Manager Service nutzen kann.

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

Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie für Ihre Servicerolle erstellt haben, führen Sie den folgenden Befehl aus, um SSM Agent den Zugriff auf die Buckets zu

ermöglichen, die Sie in der Richtlinie angegeben haben. Ersetzen Sie *account-id* und *my-bucket-policy-name* durch Ihre AWS-Konto -ID und Ihren Bucket-Namen.

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

(Optional) Führen Sie den folgenden Befehl ausSSM Agent, damit der verwaltete Knoten in Ihrem Namen AWS Directory Service auf Anfragen zum Beitritt zur Domäne zugreifen kann. Ihre Servicerolle benötigt diese Richtlinie nur, wenn Sie Ihre Knoten mit einem Microsoft-AD-Verzeichnis verbinden.

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

(Optional) Führen Sie den folgenden Befehl aus, damit der CloudWatch Agent auf Ihren verwalteten Knoten ausgeführt werden kann. Dieser Befehl ermöglicht es, Informationen auf einem Knoten zu lesen und darauf zu schreiben CloudWatch. Ihr Serviceprofil benötigt diese Richtlinie nur, wenn Sie Dienste wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden.

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

Fahren Sie fort mit Erstellen Sie eine Hybridaktivierung, um Knoten bei Systems Manager zu registrieren.

Erstellen Sie eine Hybridaktivierung, um Knoten bei Systems Manager zu registrieren

Um andere Maschinen als Instances der Amazon Elastic Compute Cloud (EC2) als verwaltete Knoten für eine Hybrid- und Multi-Cloud-Umgebung einzurichten, erstellen Sie eine Hybrid-Aktivierung und wenden diese an. Nachdem Sie die Aktivierung erfolgreich abschließen, erhalten Sie sofort einen Aktivierungscode und eine Aktivierungs-ID oben auf der Konsolenseite. Sie geben diese Kombination aus Code und ID bei der Installation AWS Systems Manager SSM Agent auf Nicht-EC2-Computern

für Ihre Hybrid- und Multi-Cloud-Umgebung an. Der Code und die ID bieten einen sicheren Zugriff auf den Systems-Manager-Service von Ihren verwalteten Knoten aus.

> **Important**
>
> Systems Manager übergibt den Aktivierungscode und die ID sofort an die Konsole oder das Befehlsfenster, je nachdem, wie Sie die Aktivierung erstellt haben. Kopieren Sie diese Informationen und speichern Sie sie an einem sicheren Ort. Wenn Sie die Konsole verlassen oder das Befehlsfenster schließen, können diese Informationen verloren gehen. Wenn Sie die Informationen verlieren, müssen Sie eine neue Aktivierung erstellen.

Informationen zu den Aktivierungabläufen

Ein Aktivierungsablauf ist ein Zeitfenster, in dem Sie On-Premises-Maschinen mit Systems Manager registrieren können. Eine abgelaufene Aktivierung hat keine Auswirkungen auf Ihre Server oder VMs, die Sie zuvor bei Systems Manager registriert haben. Wenn eine Aktivierung abgelaufen ist, können Sie anschließend mit dieser bestimmten Aktivierung keine weiteren Server oder VMs mit Systems Manager registrieren. Sie müssen eine neue erstellen.

Jeder On-Premises-Server und jede VM, die Sie zuvor registriert haben, bleibt als verwalteter Systems-Manager-Knoten registriert, bis Sie die Registrierung explizit abmelden. Sie können die Registrierung eines verwalteten Knotens auf der Registerkarte Verwaltete Knoten Fleet Manager in der Systems Manager Manager-Konsole mithilfe des AWS CLI Befehls deregister-managed-instanceoder mithilfe des API-Aufrufs aufheben. DeregisterManagedInstance

Informationen zu verwalteten Knoten

Ein verwalteter Knoten ist ein beliebiger Computer, für den er konfiguriert ist. AWS Systems Manager AWS Systems Manager unterstützt Amazon Elastic Compute Cloud (Amazon EC2) -Instances, Edge-Geräte und lokale Server oder VMs, einschließlich VMs in anderen Cloud-Umgebungen. Bisher wurden alle verwalteten Knoten als verwaltete Instances bezeichnet. Der Begriff Instance bezieht sich jetzt nur noch auf EC2-Instances. Der Befehl deregister-managed-instance wurde vor dieser Terminologieänderung benannt.

Informationen zu Aktivierungs-Tags

Wenn Sie eine Aktivierung entweder mithilfe von AWS Command Line Interface (AWS CLI) oder erstellen AWS Tools for Windows PowerShell, können Sie Tags angeben. Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource

unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Hier ist ein AWS CLI Beispielbefehl zur Ausführung auf einem lokalen Linux-Computer, der optionale Tags enthält.

```
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"
```

Wenn Sie beim Erstellen einer Aktivierung Tags angeben, werden diese Tags automatisch Ihren verwalteten Knoten zugewiesen, wenn Sie diese aktivieren.

Es ist nicht möglich, Tags zu einer vorhandenen Aktivierung hinzuzufügen oder daraus zu löschen. Wenn Sie Ihren On-Premises-Servern und VMs nicht mithilfe einer Aktivierung automatisch Tags zuweisen möchten, können Sie ihnen später Tags hinzufügen. Genauer gesagt können Sie Ihre On-Premises-Server und VMs markieren, nachdem sie sich zum ersten Mal mit Systems Manager verbunden haben. Nachdem diese eine Verbindung hergestellt haben, wird ihnen eine verwaltete Knoten-ID zugewiesen und sie werden in der Systems-Manager-Konsole mit einer ID mit dem Präfix „mi-“ aufgeführt. Informationen zum Hinzufügen von Tags zu Ihren verwalteten Knoten ohne Verwendung des Aktivierungsprozesses finden Sie unter Markieren verwalteter Knoten.

> **Note**
>
> Sie können einer Aktivierung keine Tags zuweisen, wenn Sie sie mithilfe der Systems Manager-Konsole erstellen. Sie müssen ihn entweder mit den Tools AWS CLI oder mit den Tools für Windows erstellen PowerShell.

Wenn Sie einen On-Premises-Server oder eine virtuelle Maschine (VM) nicht mehr mithilfe von Systems Manager verwalten möchten, können Sie die Registrierung aufheben. Weitere Informationen finden Sie unter Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung.

Themen

- Verwenden von AWS Management Console , um eine Aktivierung für die Registrierung verwalteter Knoten bei Systems Manager zu erstellen

- Verwenden der Befehlszeile zum Erstellen einer Aktivierung für die Registrierung verwalteter Knoten bei Systems Manager

Verwenden von AWS Management Console , um eine Aktivierung für die Registrierung verwalteter Knoten bei Systems Manager zu erstellen

So erstellen Sie eine Aktivierung für einen verwalteten Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Hybrid Activations.
3. Wählen Sie Create activation aus.

   –oder–

   Wenn Sie in der aktuellen Version zum ersten Mal auf Hybrid-Aktivierungen zugreifen AWS-Region, wählen Sie Create an Activations.
4. (Optional) Geben Sie für Aktivierungs-Beschreibung eine Beschreibung für diese Aktivierung ein. Wir empfehlen die Eingabe einer Beschreibung, wenn Sie eine große Anzahl von Servern und VMs aktivieren möchten.
5. Geben Sie unter Instanzlimit die Gesamtzahl der Knoten an, bei denen Sie sich im AWS Rahmen dieser Aktivierung registrieren möchten. Der Standardwert ist 1 Instance.
6. Wählen Sie für die IAM-Rolle eine Servicerollenoption aus, mit der Ihre Server und VMs AWS Systems Manager in der Cloud kommunizieren können:

   - Option 1: Wählen Sie Verwenden Sie die vom System erstellte Standardrolle, um eine Rolle und verwaltete Richtlinie zu verwenden, die von AWS bereitgestellt wird.
   - Option 2: Wählen Sie Select an existing custom IAM role that has the required permissions (Vorhandene benutzerdefinierte IAM Rolle auswählen) aus, um die optionale benutzerdefinierte Rolle zu verwenden, die Sie zuvor erstellt haben. Diese Rolle muss über eine Vertrauensbeziehungsrichtlinie verfügen, die `"Service": "ssm.amazonaws.com"` angibt. Wenn Ihre IAM-Rolle dieses Prinzip in einer Vertrauensbeziehungsrichtlinie nicht angibt, wird die folgende Fehlermeldung angezeigt:

     ```
     An error occurred (ValidationException) when calling the CreateActivation
     ```

```
                                        operation: Not existing role:
arn:aws:iam::<accountid>:role/SSMRole
```

Weitere Informationen zum Erstellen dieser Rolle finden Sie unter Erstellen Sie die für Systems Manager in Hybrid- und Multicloud-Umgebungen erforderliche IAM-Servicerolle.

7. Geben Sie im Feld Activation expiry date (Aktivierungsablaufdatum) ein Ablaufdatum für die Aktivierung an. Das Verfallsdatum muss in der Zukunft liegen - jedoch nicht mehr als 30 Tage. Der Standardwert beträgt 24 Stunden.

> **Note**
>
> Wenn Sie nach dem Ablaufdatum weitere verwaltete Knoten registrieren möchten, müssen Sie eine neue Aktivierung erstellen. Das Verfallsdatum wirkt sich nicht auf registrierte und ausgeführte Knoten aus.

8. (Optional) Geben Sie für das Feld Default instance name (Standard-Instance-Name) einen identifizierenden Namenswert an, der für alle verwalteten Knoten angezeigt werden soll, die dieser Aktivierung zugeordnet sind.
9. Wählen Sie Create activation aus. Der Systems Manager gibt den Aktivierungscode und die ID sofort an die Konsole zurück.

Verwenden der Befehlszeile zum Erstellen einer Aktivierung für die Registrierung verwalteter Knoten bei Systems Manager

Das folgende Verfahren beschreibt, wie Sie AWS Command Line Interface (AWS CLI) (unter Linux oderWindows) verwenden oder AWS Tools for PowerShell eine verwaltete Knotenaktivierung erstellen.

So erstellen Sie eine Aktivierung

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS Tools for PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.
2. Führen Sie den folgenden Befehl aus, um eine Aktivierung zu erstellen.

Note

- Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.
- Die Rolle, die Sie für den *iam-role*-Parameter angeben, muss über eine Vertrauensbeziehungsrichtlinie verfügen, die `"Service": "ssm.amazonaws.com"` angibt. Wenn Ihre AWS Identity and Access Management (IAM-) Rolle dieses Prinzip nicht in einer Vertrauensstellungsrichtlinie spezifiziert, erhalten Sie die folgende Fehlermeldung:

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

  Weitere Informationen zum Erstellen dieser Rolle finden Sie unter Erstellen Sie die für Systems Manager in Hybrid- und Multicloud-Umgebungen erforderliche IAM-Servicerolle.
- Geben Sie für `--expiration-date` ein Datum im Zeitstempel-Format an, z. B. `"2021-07-07T00:00:00"`, wenn der Aktivierungscode abläuft. Sie können ein Datum bis zu 30 Tage im Voraus angeben. Wenn Sie kein Ablaufdatum angeben, läuft der Aktivierungscode innerhalb von 24 Stunden ab.

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"}
```

Ein Beispiel.

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"}
```

Wenn die Aktivierung erfolgreich erstellt wurde, gibt das System sofort einen Aktivierungscode und eine Aktivierungs-ID zurück.

So installieren Sie das SSM Agent auf Hybrid-Linux-Knoten

In diesem Thema wird beschrieben, wie die Installation AWS Systems Manager SSM Agent auf Linux-Computern, die nicht EC2 (Amazon Elastic Compute Cloud) sind, in einer Hybrid- und Multi-Cloud-Umgebung durchgeführt wird. Wenn Sie Windows Server-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung verwenden möchten, finden Sie die entsprechende Anleitung im nächsten Schritt So installieren Sie den SSM Agent auf Windows Hybridknoten.

> **Important**
>
> Bei diesem Verfahren handelt es sich um andere Maschinentypen als EC2-Instances für eine Hybrid- und Multi-Cloud-Umgebung. Informationen zum Herunterladen und Installieren von SSM Agent auf einer EC2-Instance für Linux finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.

Bevor Sie beginnen, finden Sie den Aktivierungscode und die Aktivierungs-ID, die Sie nach Abschluss der Hybrid-Aktivierung unter Erstellen Sie eine Hybridaktivierung, um Knoten bei Systems Manager zu registrieren erhalten haben. Sie geben den Code und die ID in den folgenden Schritten an.

`region` steht für den Bezeichner für eine Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten

Region-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

Beispiel: Um den SSM Agent für Amazon Linux, RHEL, CentOS und SLES-64-Bit-Versionen aus der Region USA Ost (Ohio) (us-east-2) herunterzuladen, verwenden Sie folgende URL:

```
https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/linux_amd64/amazon-ssm-agent.rpm
```

Amazon Linux 1,Amazon Linux 2, RHEL, Oracle Linux, CentOS, and SLES

- x86_64

  `https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_amd64/amazon-ssm-agent.rpm`

- x86

  `https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_386/amazon-ssm-agent.rpm`

- ARM64

  `https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_arm64/amazon-ssm-agent.rpm`

RHEL 6.x, CentOS 6.x

- x86_64

  `https://s3.region.amazonaws.com/amazon-ssm-region/3.0.1479.0/linux_amd64/amazon-ssm-agent.rpm`

- x86

  `https://s3.region.amazonaws.com/amazon-ssm-region/3.0.1479.0/linux_386/amazon-ssm-agent.rpm`

Ubuntu Server

- x86_64

```
https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_amd64/
amazon-ssm-agent.deb
```

- ARM64

  ```
  https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_arm64/
  amazon-ssm-agent.deb
  ```

- x86

  ```
  https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_386/
  amazon-ssm-agent.deb
  ```

Debian Server

- x86_64

  ```
  https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_amd64/
  amazon-ssm-agent.deb
  ```

- ARM64

  ```
  https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_arm64/
  amazon-ssm-agent.deb
  ```

Raspberry Pi OS (formerly Raspbian)

- ```
  https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_arm/
  amazon-ssm-agent.deb
  ```

So installieren Sie SSM Agent auf Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung

1. Melden Sie sich bei einem Server oder einer VM in Ihrer Hybrid- und Multi-Cloud-Umgebung an.
2. Wenn Sie einen HTTP- oder HTTPS-Proxy verwenden, müssen Sie die `http_proxy` oder `https_proxy`-Umgebungsvariablen in der aktuellen Shell-Sitzung einstellen. Wenn Sie keinen Proxy verwenden, können Sie diesen Schritt überspringen.

   Geben Sie für einen HTTP-Proxy-Server die folgenden Befehle in der Befehlszeile ein:

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

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

Geben Sie für einen HTTPS-Proxy-Server die folgenden Befehle in der Befehlszeile ein:

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

3. Kopieren Sie einen der folgenden Befehlsblöcke und fügen Sie ihn in SSH ein. Ersetzen Sie die Platzhalterwerte durch den Aktivierungscode und die Aktivierungs-ID, die generiert werden, wenn Sie eine Aktivierung für einen verwalteten Knoten erstellen, und durch die Kennung der AWS-Region , aus der Sie SSM Agent herunterladen möchten. Drücken Sie anschließend `Enter`.

   > **Note**
   >
   > Beachten Sie die folgenden wichtigen Details:
   >
   > - `sudo` ist nicht erforderlich, wenn Sie ein Stammbenutzer sind.
   > - Laden Sie es `ssm-setup-cli` von dem Ort AWS-Region herunter, an dem Ihre Hybrid-Aktivierung erstellt wurde.
   > - `ssm-setup-cli` unterstützt eine `manifest-url`-Option, die die Quelle bestimmt, von der der Agent heruntergeladen wird. Geben Sie für diese Option keinen Wert an, es sei denn, Ihre Organisation verlangt dies.
   > - Verwenden Sie bei der Registrierung von Instances nur den bereitgestellten Download-Link für `ssm-setup-cli`. `ssm-setup-cli`sollte nicht separat für die zukünftige Verwendung aufbewahrt werden.
   > - Sie können das hier bereitgestellte Skript verwenden, um die Signatur von zu überprüfen`ssm-setup-cli`.

   `Region` steht für den Bezeichner für eine Region AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

   `ssm-setup-cli` enthält zusätzlich die folgenden Optionen:

   - `version` – Gültige Werte sind `latest` und `stable`.

- `downgrade` – Erlaubt, dass SSM Agent auf eine ältere Version zurückgesetzt wird. Geben Sie `true` an, um eine frühere Version des Agenten zu installieren.
- `skip-signature-validation` – Überspringt die Signaturvalidierung während des Herunterladens und der Installation des Agenten.

RHEL 6.x, und 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.xOracle Linux, CentOS 7.x und 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 und 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 (früher 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

- Verwenden von .deb-Paketen

  ```
  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"
  ```

- Verwenden von Snap-Paketen

  Sie müssen keine URL für den Download angeben, da der `snap`-Befehl den Agenten automatisch aus dem Snap App Store unter https://snapcraft.io herunterlädt.

  Auf Ubuntu Server 20.10 STR und 20.04, 18.04 und 16.04 LTS werden SSM Agent-Installationsprogrammdateien, einschließlich Agentenbinär- und Konfigurationsdateien, im folgenden Verzeichnis gespeichert: `/snap/amazon-ssm-agent/current/`. Wenn Sie Änderungen an einer Konfigurationsdatei in diesem Verzeichnis vornehmen, müssen Sie dies Datei

aus dem Verzeichnis `/snap` in das Verzeichnis `/etc/amazon/ssm/` kopieren. Protokoll- und Bibliotheksdateien wurden nicht geändert (`/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

Der Kandidat-Kanal im Snap Store enthält die neueste Version von SSM Agent; nicht den stabilen Kanal. Wenn Sie SSM Agent-Versionsinformationen auf dem Kandidatenkanal nachverfolgen möchten, führen Sie den folgenden Befehl auf Ihren von Ubuntu Server verwalteten 18.04- und 16.04-LTS-64-Bit-Knoten aus.

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

Der Befehl lädt SSM Agent herunter und installiert es auf der hybrid-aktivierten Maschine in Ihrer Hybrid- und Multi-Cloud-Umgebung. Der Befehl stoppt den SSM Agent und registriert die Maschine beim Systems Manager-Service. Die Maschine ist nun ein verwalteter Knoten. Für Systems Manager konfigurierte Amazon-EC2-Instances sind ebenfalls verwaltete Knoten. In der Systems-Manager-Konsole werden Ihre hybrid-aktivierten Knoten jedoch von Amazon-EC2-Instances mit dem Präfix „mi-“ unterschieden.

Fahren Sie fort mit So installieren Sie den SSM Agent auf Windows Hybridknoten.

Automatische Drehung des privaten Schlüssels einrichten

Um Ihre Sicherheitslage zu stärken, können Sie AWS Systems Manager Agent (SSM Agent) so konfigurieren, dass der private Schlüssel für Ihre Hybrid- und Multi-Cloud-Umgebung automatisch rotiert wird. Sie können auf dieses Feature zugreifen, indem Sie SSM Agent-Version 3.0.1031.0 oder höher verwenden. Aktivieren Sie dieses Feature wie folgt.

Um SSM Agent zu konfigurieren, um den privaten Schlüssel für eine Hybrid- und Multi-Cloud-Umgebung zu rotieren

1. Navigieren Sie zu `/etc/amazon/ssm/` auf einem Linux-Computer oder zu `C:\Program Files\Amazon\SSM` für einen Windows-Computer.
2. Kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` in eine neue Datei namens `amazon-ssm-agent.json`. Speichern Sie `amazon-ssm-agent.json` in demselben Verzeichnis, in dem sich `amazon-ssm-agent.json.template` befindet.
3. Suchen Sie `Profile`, `KeyAutoRotateDays`. Geben Sie die gewünschte Anzahl der Tage zwischen den automatischen Drehungen des privaten Schlüssels eingeben.
4. Starten Sie SSM Agent neu.

Jedes Mal, wenn Sie die Konfiguration ändern, starten Sie SSM Agent neu.

Sie können andere Features von SSM Agent mit dem gleichen Verfahren personalisieren. Eine up-to-date Liste der verfügbaren Konfigurationseigenschaften und ihrer Standardwerte finden Sie unter Definitionen von Konfigurationseigenschaften.

Deregistrierung und Neuregistrierung eines verwalteten Knotens

Sie können die Registrierung eines hybridaktivierten verwalteten Knotens aufheben, indem Sie den DeregisterManagedInstanz-API-Vorgang entweder über Tools für Windows AWS CLI oder über Tools für Windows aufrufen. PowerShell Hier sehen Sie ein Beispiel für einen CLI-Befehl:

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

Um die verbleibenden Registrierungsinformationen für den Agenten zu entfernen, entfernen Sie den `IdentityConsumptionOrder`-Schlüssel aus der `amazon-ssm-agent.json`-Datei. Führen Sie anschließend den folgenden Befehl aus:

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

Sie können eine Maschine neu registrieren, nachdem Sie sie abgemeldet haben. Gehen Sie wie folgt vor, um eine Maschine neu zu registrieren. Nachdem Sie das Verfahren abgeschlossen haben, wird Ihr verwalteter Knoten erneut in der Liste der verwalteten Knoten angezeigt.

So registrieren Sie einen verwalteten Knoten auf einer Nicht-EC2-Linux-Maschine neu

1. Verbinden Sie sich mit Ihrer Maschine.

2. Führen Sie den folgenden Befehl aus. Stellen Sie sicher, dass Sie die Platzhalterwerte durch den Aktivierungscode und die Aktivierungs-ID ersetzen, die generiert werden, wenn Sie eine Aktivierung für einen verwalteten Knoten erstellen, sowie durch die Kennung der Region, aus der Sie den SSM Agent herunterladen möchten.

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

Problembehebung bei der SSM Agent-Installation auf Nicht-EC2-Linux-Maschinen

Nutzen Sie die folgenden Informationen, um Probleme bei der Installation von SSM Agent auf hybrid-aktivierten Linux-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung zu beheben.

DeliveryTimedOut Sie erhalten eine Fehlermeldung

Problem: Wenn Sie eine Maschine in einer Maschine AWS-Konto als verwalteten Knoten für eine separate Maschine konfigurieren AWS-Konto, erhalten Sie `DeliveryTimedOut` nach der Ausführung die Befehle zur Installation SSM Agent auf dem Zielcomputer.

Lösung: `DeliveryTimedOut` ist der erwartete Antwortcode für dieses Szenario. Der Befehl zum Installieren von SSM Agent auf dem Zielknoten ändert die Knoten-ID des Quellknotens. Da sich die Knoten-ID geändert hat, kann der Quellknoten dem Zielknoten nicht antworten, dass der Befehl bei der Ausführung fehlgeschlagen, abgeschlossen oder abgelaufen ist.

Knotenzuordnungen können nicht geladen werden

Problem: Nach dem Ausführen der Installationsbefehle wird der folgende Fehler im SSM Agent in den Fehlerprotokollen angezeigt:

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

Sie sehen diesen Fehler, wenn die Computer-ID bei einem Neustart nicht beibehalten wird.

Lösung: Um dieses Problem zu lösen, führen Sie den folgenden Befehl aus. Dieser Befehl zwingt, dass die Computer-ID bei einem Neustart beibehalten wird.

```
umount /etc/machine-id
```

```
systemd-machine-id-setup
```

So installieren Sie den SSM Agent auf Windows Hybridknoten

Dieses Thema beschreibt, wie Sie SSM Agent auf Windows Server-Maschinen für eine Hybrid- und Multi-Cloud-Umgebung installieren. Wenn Sie planen, Nicht-EC2-Linux-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung zu verwenden, lesen Sie den vorherigen Schritt, So installieren Sie das SSM Agent auf Hybrid-Linux-Knoten.

> **Important**
>
> Dieses Verfahren gilt für Nicht-EC2-Maschinen (Amazon Elastic Compute Cloud) in Hybrid- und Multi-Cloud-Umgebungen. Informationen zum Herunterladen und Installieren von SSM Agent auf einer EC2-Instance für Windows Server finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Windows Server.

Bevor Sie beginnen, finden Sie den Aktivierungscode und die Aktivierungs-ID, die Sie nach Abschluss der Hybrid-Aktivierung unter Erstellen Sie eine Hybridaktivierung, um Knoten bei Systems Manager zu registrieren erhalten haben. Sie geben den Code und die ID in den folgenden Schritten an.

So installieren Sie SSM Agent auf Nicht-EC2-Windows Server-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung

1. Melden Sie sich bei einem Server oder einer VM in Ihrer Hybrid- und Multi-Cloud-Umgebung an.
2. Wenn Sie einen HTTP- oder HTTPS-Proxy verwenden, müssen Sie die `http_proxy` oder `https_proxy`-Umgebungsvariablen in der aktuellen Shell-Sitzung einstellen. Wenn Sie keinen Proxy verwenden, können Sie diesen Schritt überspringen.

   Legen Sie für einen HTTP-Proxyserver folgende Variable fest:

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

   Legen Sie für einen HTTPS-Proxyserver folgende Variable fest:

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

```
https_proxy=https://hostname:port
```

3. Öffnen Sie Windows PowerShell im erweiterten (Verwaltungs-)Modus.
4. Kopieren Sie den folgenden Befehlsblock in Windows PowerShell. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Zum Beispiel der Aktivierungscode und die Aktivierungs-ID, die generiert wurden, wenn Sie eine Hybrid-Aktivierung erstellen, und zwar mit der ID, SSM Agent von der AWS-Region Sie herunterladen möchten.

> **Note**
>
> Beachten Sie die folgenden wichtigen Details:
>
> - `ssm-setup-cli` unterstützt eine `manifest-url`-Option, die die Quelle bestimmt, von der der Agent heruntergeladen wird. Geben Sie für diese Option keinen Wert an, es sei denn, Ihre Organisation verlangt dies.
> - Sie können das hier bereitgestellte Skript verwenden, um die Signatur von zu überprüfen`ssm-setup-cli`.
> - Verwenden Sie bei der Registrierung von Instances nur den bereitgestellten Download-Link für `ssm-setup-cli`. `ssm-setup-cli`sollte nicht separat für die zukünftige Verwendung aufbewahrt werden.

Region steht für den Bezeichner einer Region AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

`ssm-setup-cli` enthält zusätzlich die folgenden Optionen:

- `version` – Gültige Werte sind `latest` und `stable`.
- `downgrade` – Setzt den Agenten auf eine frühere Version zurück.
- `skip-signature-validation` – Überspringt die Signaturvalidierung während des Herunterladens und der Installation des Agenten.

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. Drücken Sie `Enter`.

> **Note**
>
> Wenn der Befehl fehlschlägt, stellen Sie sicher, dass Sie die neueste Version von ausführen AWS Tools for PowerShell.

Der Befehl hat folgende Auswirkungen:

- Lädt SSM Agent herunter und installiert es auf dem Computer.
- Registriert den Computer beim Systems Manager Service.
- Gibt eine Antwort auf die Anfrage zurück, die der folgenden ähnlich ist:

```
    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
```

Die Maschine ist nun ein verwalteter Knoten. Diese verwalteten Knoten werden jetzt mit dem Präfix "mi-" gekennzeichnet. Sie können verwaltete Knoten auf der Seite Verwaltete Knoten in anzeigenFleet Manager, indem Sie den AWS CLI Befehl describe-instance-informationoder den API-Befehl verwenden DescribeInstanceInformation.

Automatische Drehung des privaten Schlüssels einrichten

Um Ihre Sicherheitslage zu stärken, können Sie AWS Systems Manager Agent (SSM Agent) so konfigurieren, dass der private Schlüssel für eine Hybrid- und Multi-Cloud-Umgebung automatisch rotiert wird. Sie können auf dieses Feature zugreifen, indem Sie SSM Agent-Version 3.0.1031.0 oder höher verwenden. Aktivieren Sie dieses Feature wie folgt.

Um SSM Agent zu konfigurieren, um den privaten Schlüssel für eine Hybrid- und Multi-Cloud-Umgebung zu rotieren

1. Navigieren Sie zu `/etc/amazon/ssm/` auf einem Linux-Computer oder zu `C:\Program Files\Amazon\SSM` für einen Windows Server-Computer.
2. Kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` in eine neue Datei namens `amazon-ssm-agent.json`. Speichern Sie `amazon-ssm-agent.json` in demselben Verzeichnis, in dem sich `amazon-ssm-agent.json.template` befindet.
3. Suchen Sie `Profile`, `KeyAutoRotateDays`. Geben Sie die gewünschte Anzahl der Tage zwischen den automatischen Drehungen des privaten Schlüssels eingeben.
4. Starten Sie SSM Agent neu.

Jedes Mal, wenn Sie die Konfiguration ändern, starten Sie SSM Agent neu.

Sie können andere Features von SSM Agent mit dem gleichen Verfahren personalisieren. Eine up-to-date Liste der verfügbaren Konfigurationseigenschaften und ihrer Standardwerte finden Sie unter Definitionen von Konfigurationseigenschaften.

Deregistrierung und Neuregistrierung eines verwalteten Knotens

Sie können die Registrierung eines verwalteten Knotens aufheben, indem Sie den DeregisterManagedInstanz-API-Vorgang entweder über Tools für Windows AWS CLI oder über Tools für Windows aufrufen. PowerShell Hier sehen Sie ein Beispiel für einen CLI-Befehl:

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

Um die verbleibenden Registrierungsinformationen für den Agenten zu entfernen, entfernen Sie den `IdentityConsumptionOrder`-Schlüssel aus der `amazon-ssm-agent.json`-Datei. Führen Sie anschließend den folgenden Befehl aus:

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

Sie können eine Maschine neu registrieren, nachdem Sie sie abgemeldet haben. Gehen Sie wie folgt vor, um eine Maschine erneut als verwalteten Knoten zu registrieren. Nachdem Sie das Verfahren abgeschlossen haben, wird Ihr verwalteter Knoten erneut in der Liste der verwalteten Knoten angezeigt.

So registrieren Sie einen verwalteten Knoten auf einem Windows-Hybridcomputer neu

1. Verbinden Sie sich mit Ihrer Maschine.

2. Führen Sie den folgenden Befehl aus. Achten Sie darauf, die Platzhalterwerte durch den Aktivierungscode und die Aktivierungs-ID zu ersetzen, die bei der Erstellung einer Hybrid-Aktivierung generiert werden, sowie durch die Kennung der Region, aus der Sie den SSM Agent herunterladen möchten.

```
'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"
```

Verwaltung von Edge-Geräten mit Systems Manager

In diesem Abschnitt werden die Einrichtungsaufgaben beschrieben, die Konto- und Systemadministratoren ausführen, um die Konfiguration und Verwaltung von AWS IoT Greengrass Kerngeräten zu ermöglichen. Nachdem Sie diese Aufgaben abgeschlossen haben, können Benutzer, denen der AWS-Konto Administrator Berechtigungen erteilt hat, sie AWS Systems Manager zur Konfiguration und Verwaltung der AWS IoT Greengrass Kerngeräte ihrer Organisation verwenden.

> **Note**
>
> - SSM Agentfür wird unter macOS Windows 10 AWS IoT Greengrass nicht unterstützt. Sie können keine Systems-Manager-Funktionen verwenden, um Edge-Geräte zu verwalten und zu konfigurieren, die diese Betriebssysteme verwenden.
> - Systems Manager unterstützt auch Edge-Geräte, die nicht als AWS IoT Greengrass Core-Geräte konfiguriert sind. Um Systems Manager zur Verwaltung von AWS IoT Core-Geräten und AWS Nicht-Edge-Geräten zu verwenden, müssen Sie sie mithilfe einer Hybridaktivierung konfigurieren. Weitere Informationen finden Sie unter Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen.
> - Um Session Manager und Microsoft-Anwendungs-Patching mit Ihren Edge-Geräten zu verwenden, müssen Sie das Advanced-Instances-Kontingent aktivieren. Weitere Informationen finden Sie unter Aktivieren des Kontingents für erweiterte Instances.

Bevor Sie beginnen

Stellen Sie sicher, dass Ihre Edge-Geräte die folgenden Anforderungen erfüllen.

- Ihre Edge-Geräte müssen die Anforderungen erfüllen, um als AWS IoT Greengrass Core-Geräte konfiguriert zu werden. Weitere Informationen finden Sie unter Einrichten von AWS IoT Greengrass Kerngeräten im AWS IoT Greengrass Version 2 Entwicklerhandbuch.
- Ihre Edge-Geräte müssen mit AWS Systems Manager Agent (SSM Agent) kompatibel sein. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme für Systems Manager.
- Ihre Edge-Geräte müssen mit dem Systems-Manager-Service in der Cloud kommunizieren können. Systems Manager unterstützt keine getrennten Edge-Geräte.

Informationen über das Einrichten von Edge-Geräten

Das Einrichten von AWS IoT Greengrass Geräten für Systems Manager umfasst die folgenden Prozesse.

> **Note**
>
> Informationen zur Deinstallation SSM Agent von einem Edge-Gerät aus finden Sie unter Deinstallieren des AWS Systems Manager Agenten im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

Erstellen Sie eine IAM-Servicerolle für Ihre Edge-Geräte

AWS IoT Greengrass Für die Kommunikation mit Kerngeräten ist eine AWS Identity and Access Management (IAM) -Servicerolle erforderlich. AWS Systems Manager Die Rolle gewährt dem Systems Manager Manager-Dienst `AssumeRole`Vertrauen AWS Security Token Service (AWS STS). Sie müssen die Servicerolle nur einmal für jedes AWS-Konto erstellen. Sie geben diese Rolle für den `RegistrationRole` Parameter an, wenn Sie die SSM Agent Komponente konfigurieren und auf Ihren AWS IoT Greengrass Geräten bereitstellen. Wenn Sie diese Rolle bereits beim Einrichten von Nicht-EC2-Knoten für eine Hybrid- und Multi-Cloud-Umgebung erstellt haben, können Sie diesen Schritt überspringen.

> **Note**
>
> Benutzer in Ihrem Unternehmen oder Ihrer Organisation, die Systems Manager auf Ihren Edge-Geräten verwenden, müssen in IAM die Berechtigung für den Aufruf der Systems-Manager-API erhalten.

S3-Bucket-Richtlinienanforderung

Wenn einer der folgenden Fälle zutrifft, müssen Sie eine benutzerdefinierte IAM-Berechtigungsrichtlinie für Amazon Simple Storage Service (Amazon S3)-Buckets erstellen, bevor Sie dieses Verfahren durchführen:

- Fall 1: Sie verwenden einen VPC-Endpunkt, um Ihre VPC privat mit unterstützten AWS-Services und VPC-Endpunktdiensten zu verbinden, die von unterstützt werden. AWS PrivateLink
- Fall 2: Sie beabsichtigen, einen S3-Bucket zu verwenden, den Sie im Rahmen Ihrer Systems Manager-Operationen erstellen, z. B. zum Speichern von Ausgaben für Run Command-Befehle oder Session Manager-Sitzungen in einem S3-Bucket. Bevor Sie fortfahren, befolgen Sie die Schritte unter Erstellen einer benutzerdefinierten S3-Bucket-Richtlinie für ein Instance-Profil. Die Informationen über S3-Bucket-Richtlinien in diesem Thema gelten auch für Ihre Service-Rolle.

> **Note**
>
> Wenn Ihre Geräte durch eine Firewall geschützt sind und Sie planen, Patch Managerzu verwenden, muss die Firewall den Zugriff auf den Patch-Baseline-Endpunkt `arn:aws:s3:::patch-baseline-snapshot-`*`region`*`/*` erlauben. *`region`* steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

AWS CLI

So erstellen Sie eine IAM-Dienstrolle für eine AWS IoT Greengrass Umgebung ()AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Erstellen Sie eine Textdatei mit einem Namen wie z. B. `SSMService-Trust.json` mit der folgenden Vertrauensrichtlinie auf Ihrer lokalen Maschine. Stellen Sie sicher, dass Sie die Datei mit der Erweiterung `.json` speichern.

> **Note**
>
> Notieren Sie den Namen. Sie geben es an, wenn Sie es SSM Agent auf Ihren AWS IoT Greengrass Kerngeräten bereitstellen.

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

3. Öffnen Sie die und führen Sie in dem Verzeichnis AWS CLI, in dem Sie die JSON-Datei erstellt haben, den Befehl create-role aus, um die Servicerolle zu erstellen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux und 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. Führen Sie den attach-role-policy-Befehl wie folgt, um es der gerade von Ihnen erstellten Servicerolle zu ermöglichen, ein Sitzungs-Token zu erstellen. Das Sitzungs-Token gewährt Ihren Edge-Geräten die Berechtigung zum Ausführen von Befehlen mit Systems Manager.

> **Note**
>
> Die Richtlinien, die Sie für ein Serviceprofil für Edge-Geräte hinzufügen, sind die gleichen Richtlinien, die zum Erstellen eines Instance-Profils für Amazon Elastic Compute Cloud (Amazon EC2)-Instances verwendet werden. Weitere Informationen zu den in den folgenden Befehlen verwendeten IAM-Richtlinien finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

(Erforderlich) Führen Sie den folgenden Befehl aus, damit ein Edge-Gerät die AWS Systems Manager Service-Core-Funktionalität nutzen kann.

Linux und 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
```

Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie für Ihre Servicerolle erstellt haben, führen Sie den folgenden Befehl aus, damit AWS Systems Manager Agent (SSM Agent) auf die Buckets zugreifen kann, die Sie in der Richtlinie angegeben haben. Ersetzen Sie *account_ID* und *my_bucket_policy_name* durch Ihre AWS-Konto -ID und Ihren Bucket-Namen.

Linux und 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
```

(Optional) Führen Sie den folgenden Befehl aus, um SSM Agent in Ihrem Namen den Zugriff auf AWS Directory Service für Anforderungen zum Beitritt zur Domain von Edge-Geräten zu erlauben. Die Servicerolle benötigt diese Richtlinie nur, wenn Sie Ihre Edge-Geräte einem Microsoft-AD-Verzeichnis zuteilen.

Linux und 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
```

(Optional) Führen Sie den folgenden Befehl aus, damit der CloudWatch Agent auf Ihren Edge-Geräten ausgeführt werden kann. Dieser Befehl ermöglicht es, Informationen auf einem Gerät zu lesen und darauf zu schreiben CloudWatch. Für Ihre Servicerolle ist diese Richtlinie nur erforderlich, wenn Sie Dienste wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden.

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

Tools for PowerShell

Um eine IAM-Servicerolle für eine AWS IoT Greengrass Umgebung zu erstellen ()AWS Tools for Windows PowerShell

1. Installieren und konfigurieren Sie die AWS Tools for PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS Tools for PowerShell.

2. Erstellen Sie eine Textdatei mit einem Namen wie z. B. `SSMService-Trust.json` mit der folgenden Vertrauensrichtlinie auf Ihrer lokalen Maschine. Stellen Sie sicher, dass Sie die Datei mit der Erweiterung `.json` speichern.

   > **Note**
   >
   > Notieren Sie den Namen. Sie geben es bei der Bereitstellung SSM Agent auf Ihren AWS IoT Greengrass Kerngeräten an.

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

3. Öffnen Sie PowerShell im Administratormodus und führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, New-IAMRole wie folgt aus, um eine Servicerolle zu erstellen.

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

4. Verwenden Sie Register-IAM RolePolicy wie folgt, damit die von Ihnen erstellte Servicerolle ein Sitzungstoken erstellen kann. Das Sitzungs-Token gewährt Ihren Edge-Geräten die Berechtigung zum Ausführen von Befehlen mit Systems Manager.

> **Note**
>
> Die Richtlinien, die Sie für eine Servicerolle für Edge-Geräte in einer AWS IoT Greengrass -Umgebung hinzufügen, sind die gleichen Richtlinien, die zum Erstellen eines Instance-Profils für EC2-Instances verwendet werden. Weitere Informationen zu den in den folgenden Befehlen verwendeten AWS Richtlinien finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

(Erforderlich) Führen Sie den folgenden Befehl aus, damit ein Edge-Gerät die Kernfunktionen des AWS Systems Manager Service nutzen kann.

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

Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie für Ihre Servicerolle erstellt haben, führen Sie den folgenden Befehl aus, um SSM Agent den Zugriff auf die Buckets zu ermöglichen, die Sie in der Richtlinie angegeben haben. Ersetzen Sie *account_ID* und *my_bucket_policy_name* durch Ihre AWS-Konto -ID und Ihren Bucket-Namen.

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

(Optional) Führen Sie den folgenden Befehl aus, um SSM Agent in Ihrem Namen den Zugriff auf AWS Directory Service für Anforderungen zum Beitritt zur Domäne von Edge-Geräten zu erlauben. Die Servicerolle benötigt diese Richtlinie nur, wenn Sie Ihre Edge-Geräte einem Microsoft-AD-Verzeichnis zuteilen.

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

(Optional) Führen Sie den folgenden Befehl aus, damit der CloudWatch Agent auf Ihren Edge-Geräten ausgeführt werden kann. Dieser Befehl ermöglicht es, Informationen auf einem Gerät zu lesen und darauf zu schreiben CloudWatch. Für Ihre Servicerolle ist diese Richtlinie

nur erforderlich, wenn Sie Dienste wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden.

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

Konfigurieren Sie Ihre Edge-Geräte für AWS IoT Greengrass

Richten Sie Ihre Edge-Geräte als AWS IoT Greengrass Kerngeräte ein. Der Einrichtungsprozess umfasst die Überprüfung der unterstützten Betriebssysteme und Systemanforderungen sowie die Installation und Konfiguration der AWS IoT Greengrass Core-Software auf Ihren Geräten. Weitere Informationen finden Sie unter Einrichten von AWS IoT Greengrass -Core-Geräten im AWS IoT Greengrass Version 2 -Entwicklerhandbuch.

Aktualisieren Sie die AWS IoT Greengrass Token-Exchange-Rolle und installieren Sie sie SSM Agent auf Ihren Edge-Geräten

Im letzten Schritt zur Einrichtung und Konfiguration Ihrer AWS IoT Greengrass Kerngeräte für Systems Manager müssen Sie die Gerätedienst-Rolle AWS IoT Greengrass AWS Identity and Access Management (IAM), auch Token-Austauschrolle genannt, aktualisieren und AWS Systems Manager Agent (SSM Agent) auf Ihren AWS IoT Greengrass Geräten bereitstellen. Informationen zu diesen Prozessen finden Sie unter Installation des AWS Systems Manager -Agenten im AWS IoT Greengrass Version 2 -Entwicklerhandbuch.

Nach der Bereitstellung SSM Agent auf Ihren Geräten werden Ihre Geräte AWS IoT Greengrass automatisch bei Systems Manager registriert. Es ist keine weitere Registrierung erforderlich. Sie können damit beginnen, die Systems Manager Manager-Funktionen für den Zugriff, die Verwaltung und Konfiguration Ihrer AWS IoT Greengrass Geräte zu nutzen.

> **Note**
>
> Ihre Edge-Geräte müssen mit dem Systems-Manager-Service in der Cloud kommunizieren können. Systems Manager unterstützt keine getrennten Edge-Geräte.

Einen AWS Organizations delegierten Administrator für Systems Manager erstellen

Wenn Sie eine Organisation in einrichten AWS Organizations, weisen Sie ein Verwaltungskonto zu, mit dem alle administrativen Aufgaben für alle AWS-Services ausgeführt werden. Der Benutzer des Verwaltungskontos kann nur Systems Manager ein delegiertes Administratorkonto zuweisen, um Verwaltungsaufgaben für Change ManagerExplorer, und OpsCenter auszuführen. AWS Organizations ist ein Kontoverwaltungsdienst, den Sie verwenden können, um eine Organisation zu erstellen und diese Konten zentral AWS-Konten zu verwalten. Weitere Informationen AWS Organizations dazu finden Sie AWS Organizationsim AWS Organizations Benutzerhandbuch.

Change ManagerExplorer, undOpsCenter, Funktionen von AWS Systems Manager, arbeiten mit, AWS Organizations um Aufgaben für alle Mitgliedskonten Ihrer Organisation auszuführen. Sie können immer nur einen delegierten Administrator für alle Systems-Manager-Funktionen zuweisen. Das delegierte Administratorkonto muss Mitglied der Organisationseinheit sein, der es zugewiesen ist.

Themen

- Verwenden eines delegierten Administrators mit Change Manager
- Verwenden Sie einen delegierten Administrator mit Explorer
- Verwenden Sie einen delegierten Administrator mit OpsCenter

Verwenden eines delegierten Administrators mit Change Manager

Change Manager ist ein Change-Management-Framework für Unternehmen zum Anfordern, Genehmigen, Implementieren und Melden von Betriebsänderungen an Ihrer Anwendungskonfiguration und Infrastruktur.

Wenn Sie Change Manager unternehmensweit verwenden, weisen Sie ein delegiertes Administratorkonto zu, um Änderungsvorlagen, Genehmigungen und Berichte für alle Mitgliedskonten zu verwalten. Mit Quick Setup können Sie Change Manager für die Verwendung mit einer Organisation einrichten und das delegierte Administratorkonto auswählen. Wenn Sie Change Manager mit einem einzigen Konto verwenden AWS-Konto, ist das delegierte Administratorkonto nicht erforderlich.

Change Manager zeigt standardmäßig alle Aufgaben im Zusammenhang mit Änderungen im delegierten Administratorkonto an. Anweisungen zur Konfiguration eines delegierten Administrators

bei der Einrichtung von Change Manager für eine Organisation finden Sie unter Einrichten von Change Manager für eine Organisation (Management-Konto).

> **Important**
>
> Wenn Sie den Change Manager in einer Organisation verwenden, empfehlen wir, Änderungen immer über das delegierte Administratorkonto vorzunehmen. Obwohl Sie Änderungen von anderen Konten in der Organisation vornehmen, werden diese Änderungen nicht im delegierten Administratorkonto gemeldet oder können nicht angezeigt werden.

Verwenden Sie einen delegierten Administrator mit Explorer

Explorerist ein anpassbares Betriebs-Dashboard, das eine aggregierte Ansicht der Betriebsdaten (OpsData) für Ihren AWS-Konten Across bietet. AWS-Regionen

Sie können ein delegiertes Administratorkonto für Systems Manager konfigurieren, um Explorer Daten aus mehreren Regionen und Konten mithilfe der Ressourcendatensynchronisierung mit AWS Organizations zu aggregieren. Ein delegierter Administrator kann Explorer Daten mithilfe von, AWS Command Line Interface (AWS CLI) oder suchen AWS Management Console, filtern und aggregieren. AWS Tools for Windows PowerShell

Wenn Sie ein delegiertes Administratorkonto für Explorer verwenden, begrenzen Sie die Anzahl der Administratoren, die Daten für mehrere Konten und regionale Ressourcen erstellen oder löschen können, auf ein einzelnes AWS-Konto.

Sie können Betriebsdaten AWS-Konten in Ihrer gesamten Organisation synchronisieren, indem Sie Explorer Informationen zum Zuweisen eines delegierten Administrators von Explorer finden Sie unter Konfigurierung eines delegierten Administrators.

Verwenden Sie einen delegierten Administrator mit OpsCenter

OpsCenterbietet einen zentralen Ort, an dem Betriebsingenieure und IT-Experten betriebliche Arbeitselemente (OpsItems) im Zusammenhang mit AWS Ressourcen verwalten können. Wenn Sie OpsCenter verwenden möchten, um OpsItems zentral über Konten hinweg zu verwalten, müssen Sie die Organisation in AWS Organizations einrichten.

Mithilfe von Quick Setup für OpsCenter können Sie ein delegiertes Administratorkonto zuweisen und OpsCenter für die zentrale Verwaltung von OpsItems konfigurieren. Weitere Informationen finden Sie

unter (Optional) Konfigurieren Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup.

Allgemeine Einrichtung für AWS Systems Manager

Falls Sie dies noch nicht getan haben, registrieren Sie sich für einen AWS-Konto und erstellen Sie einen Administratorbenutzer.

Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie im Benutzerhandbuch unter Einen Berechtigungssatz erstellen.AWS IAM Identity Center

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Anweisungen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen.

Führen Sie eine Verwaltungsaufgabe mit Systems Manager aus

Verwenden Sie dieses Tutorial, um damit zu beginnen AWS Systems Manager. Sie erfahren, wie Sie eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance starten, die von Systems Manager verwaltet wird, und wie Sie eine Verbindung mit der verwalteten Instance herstellen.

Da Systems Manager eine Sammlung mehrerer Funktionen ist, kann der gesamte Service nicht durch eine einzelne Anleitung oder ein einzelnes Tutorial vorgestellt werden. Dieses Tutorial enthält eine Einführung in einige der Funktionen.

Voraussetzungen

Bevor Sie beginnen, sollten Sie sicherstellen, dass Sie die in Systems Manager mit EC2-Instances verwenden beschriebenen Schritte ausgeführt haben.

Starten einer Instance mit einer AMI mit vorinstalliertem SSM Agent

Sie können eine Amazon EC2 EC2-Instance AWS Management Console wie im folgenden Verfahren beschrieben starten. Dieses Tutorial soll Sie dabei unterstützen, Ihre erste verwaltete Instance schnell zu starten. Es deckt daher nicht alle möglichen Optionen ab.

So starten Sie eine Instance

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie über das Dashboard der EC2-Konsole in der Ansicht Launch instance (Starten der Instance) die Option Launch instance (Instance starten) aus und klicken Sie unter den angezeigten Optionen auf Launch instance (Instance starten).
3. Geben Sie für Name und Tags unter Name einen beschreibenden Namen für Ihre Instance ein.
4. Führen Sie unter Anwendungs- und Betriebssystem-Images (Amazon Machine Image) die folgenden Schritte aus:

    a. Wählen Sie die Registerkarte Schnellstart und dann Amazon Linux. Dies ist das Betriebssystem für Ihre Instance.

    b. Wählen Sie für Amazon Machine Image (AMI) eine HVM-Version von Amazon Linux 2 aus.

5. Für Instance-Typ können Sie aus der Liste Instance-Typ die Hardware-Konfiguration für Ihre Instance auswählen. Wählen Sie den Instance-Typ `t2.micro` aus (Standardeinstellung). Der `t2.micro` Instance-Typ kommt für das AWS kostenlose Kontingent in Frage. In AWS-Regionen , in denen `t2.micro` nicht verfügbar ist, können Sie eine `t3.micro`-Instance im Rahmen des kostenlosen Kontingents verwenden. Weitere Informationen finden Sie unter Kostenloses Kontingent für AWS.
6. Wählen Sie unter Schlüsselpaar (Anmeldung) für Schlüsselpaar-Name ein Schlüsselpaar aus.
7. Wählen Sie für Netzwerkeinstellungen die Option Bearbeiten aus. Beachten Sie bei Name der Sicherheitsgruppe, dass der Assistent eine Sicherheitsgruppe für Sie erstellt und ausgewählt hat. Sie können diese Sicherheitsgruppe verwenden oder alternativ eine zuvor erstellte Sicherheitsgruppe mit den folgenden Schritten auswählen:
   a. Wählen Sie Select an existing security group (Eine bestehende Sicherheitsgruppe auswählen) aus.
   b. Wählen Sie unter Common security groups (Gemeinsame Sicherheitsgruppen) Ihre Sicherheitsgruppe in der Liste mit den vorhandenen Sicherheitsgruppen aus.
8. Wenn Sie die Standardkonfiguration für die Host-Verwaltung nicht verwenden, erweitern Sie den Abschnitt Erweiterte Details und wählen Sie für das IAM-Instance-Profil das Instance-Profil aus, das Sie bei der Einrichtung in Konfigurieren Sie die für Systems Manager erforderlichen Instanzberechtigungen erstellt haben.
9. Behalten Sie die Standardauswahl für die anderen Konfigurationseinstellungen für Ihre Instance bei.
10. Überprüfen Sie eine Zusammenfassung Ihrer Instance-Konfiguration im Bereich Zusammenfassung. Sobald Sie bereit sind, wählen Sie Instance starten aus.
11. Eine Bestätigungsseite informiert Sie darüber, dass Ihre Instance gestartet wird. Wählen Sie View all Instances (Alle Instances anzeigen) aus, um die Bestätigungsseite zu schließen und zur Konsole zurückzukehren.
12. Auf dem Bildschirm Instances können Sie den Status des Starts anzeigen. Es dauert einige Zeit, bis die Instance startet.
13. Es kann ein paar Minuten dauern, bis die Instance als verwaltet angezeigt wird und Sie eine Verbindung damit herstellen können. Um zu überprüfen, ob Ihre Instance die Statusprüfungen bestanden hat, zeigen Sie diese Informationen in der Spalte Statusprüfung an.

Stellen Sie mithilfe von Systems Manager eine Connect zu Ihrer verwalteten Instanz her

So stellen Sie eine Verbindung zu Ihrer verwalteten Instance her

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben der Instanz, zu der Sie eine Verbindung herstellen möchten.
4. Wählen Sie im Menü Knotenaktionen die Option Terminalsitzung starten aus.
5. Wählen Sie Connect (Verbinden) aus.

Bereinigen Ihrer Instance

Wenn Sie die Arbeit mit der verwalteten Instance, die Sie für dieses Tutorial erstellt haben, abgeschlossen haben, beenden Sie sie. Durch das Beenden einer Instance wird diese effektiv gelöscht.

So beenden Sie Ihre Instance

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus. Wählen Sie in der Liste mit den Instances die gewünschte Instance aus.
3. Wählen Sie Instance state (Instance-Status), Terminate instance (Instance beenden).
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Beenden aus.

   Amazon EC2 fährt Ihre Instance herunter und beendet sie. Nachdem Ihre Instance beendet wurde, bleibt sie noch kurz auf der Konsole sichtbar, danach wird der Eintrag automatisch gelöscht. Sie können die beendete Instance nicht selbst aus der Konsolenanzeige entfernen.

Arbeiten mit SSM Agent

AWS Systems Manager Agent (SSM Agent) ist Amazon-Software, die auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances, Edge-Geräten, lokalen Servern und virtuellen Maschinen (VMs) ausgeführt wird. SSM Agentermöglicht es Systems Manager, diese Ressourcen zu aktualisieren, zu verwalten und zu konfigurieren. Der Agent verarbeitet Anfragen vom Systems Manager Manager-Dienst in der AWS Cloud und führt sie dann wie in der Anfrage angegeben aus. SSM Agentsendet dann mithilfe von Amazon Message Gateway Service(`ssmmessages`) Status- und Ausführungsinformationen zurück an den Systems Manager Manager-Dienst. (Bei einem AWS-Regionen Start vor 2024 können Status- und Ausführungsinformationen auch mit dem Amazon Message Delivery Service(Dienstpräfix:`ec2messages`) zurückgesendet werden.)

Wenn Sie den Datenverkehr überwachen, werden Sie feststellen, dass Ihre verwalteten Knoten mit `ssmmessages.*` Endpunkten und möglicherweise `ec2messages.*` Endpunkten kommunizieren. Weitere Informationen finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen. Informationen zum Portieren von SSM Agent Protokollen nach Amazon CloudWatch Logs finden Sie unterÜberwachung AWS Systems Manager.

Inhalt

- Erfahren Sie technische Details über die SSM Agent
- Fehlerbehebung für SSM Agent

Erfahren Sie technische Details über die SSM Agent

Verwenden Sie die Informationen in diesem Thema, um AWS Systems Manager Agent (SSM Agent) zu implementieren und zu verstehen, wie der Agent funktioniert.

Themen

- Verhalten der Anmeldeinformationen von SSM Agent-Version 3.2.x.x
- Priorität der SSM Agent-Anmeldeinformationen
- Über das lokale ssm-user-Konto
- SSM Agent und die Instance Metadata Service (IMDS)
- Behalten SSM Agent up-to-date
- Sicherstellen, dass das SSM Agent-Installationsverzeichnis nicht geändert, verschoben oder gelöscht wird

- SSM Agentfortlaufende Updates von AWS-Regionen
- SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets
- Finden Sie AMIs mit dem SSM Agent vorinstallierten
- Arbeiten mit SSM Agent auf EC2-Instances für Linux
- Arbeiten mit SSM Agent auf EC2-Instances für macOS
- Arbeiten mit SSM Agent auf EC2-Instances für Windows Server
- Prüfen des SSM Agent-Status und Starten des Agenten
- Überprüfen der SSM Agent-Versionsnummer
- Anzeigen von SSM Agent-Protokollen
- Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent
- Automatisieren von Updates für SSM Agent
- Abonnieren von SSM Agent-Benachrichtigungen

Verhalten der Anmeldeinformationen von SSM Agent-Version 3.2.x.x

SSM Agent speichert einen Satz temporärer Anmeldeinformationen unter `/var/lib/amazon/ssm/credentials` (für Linux und macOS) oder `%PROGRAMFILES%\Amazon\SSM\credentials` (für Windows Server), wenn eine Instance mit der Standardkonfiguration für die Host-Verwaltung in Quick Setup eingebunden wird. Die temporären Anmeldeinformationen haben dieselben Berechtigungen wie die IAM-Rolle, die Sie für die Standardkonfiguration für die Host-Verwaltung ausgewählt haben. Auf Linux kann nur das Root-Konto auf diese Anmeldeinformationen zugreifen. Auf Windows Server, können nur das SYSTEM-Konto und lokale Administratoren auf diese Anmeldeinformationen zugreifen.

Priorität der SSM Agent-Anmeldeinformationen

In diesem Thema werden wichtige Informationen darüber beschrieben, wie SSM Agentdie Berechtigung zum Ausführen von Aktionen auf Ihren Ressourcen erhält.

> **Note**
>
> Die Support für Edge-Geräte unterscheidet sich geringfügig. Sie müssen Ihre Edge-Geräte für die Verwendung der AWS IoT Greengrass Core-Software konfigurieren, eine AWS Identity and Access Management (IAM) -Servicerolle konfigurieren und die Bereitstellung SSM Agent

auf Ihren Geräten mithilfe AWS IoT Greengrass von. Weitere Informationen finden Sie unter Verwaltung von Edge-Geräten mit Systems Manager.

Wenn SSM Agent auf einer Maschine installiert ist, sind Berechtigungen für die Kommunikation mit dem Systems-Manager-Service erforderlich. Bei Amazon Elastic Compute Cloud (Amazon EC2)-Instances werden diese Berechtigungen in einem Instance-Profil bereitgestellt, das der Instance angehängt ist. Auf einer Nicht-EC2-Maschine erhält SSM Agent normalerweise die erforderlichen Berechtigungen aus der Datei für freigegebene Anmeldeinformationen, die sich unter `/root/.aws/credentials` (Linux und macOS) oder `%USERPROFILE%\.aws\credentials` (Windows Server) befinden. Die erforderlichen Berechtigungen werden dieser Datei während des Hybrid-Aktivierungsvorgangs hinzugefügt.

In seltenen Fällen kann es jedoch sein, dass eine Maschine Berechtigungen zu mehr als einem der Speicherorte enthält, in denen SSM Agent prüft, ob Berechtigungen zum Ausführen der Aufgaben vorhanden sind.

Nehmen wir an, Sie haben eine EC2-Instance so konfiguriert, dass sie von Systems Manager verwaltet wird. Diese Konfiguration umfasst das Anhängen eines Instance-Profils. Aber dann entscheiden Sie sich, diese Instance auch für Entwickler- oder Endbenutzer-Aufgaben zu verwenden und installieren die AWS Command Line Interface (AWS CLI) darauf. Diese Installation führt dazu, dass zusätzliche Berechtigungen zu einer Anmeldeinformationsdatei auf der Instance hinzugefügt werden.

Wenn Sie einen Systems Manager-Befehl für die Instance ausführen,versucht SSM Agent möglicherweise, Anmeldeinformationen zu verwenden, die sich von denen unterscheiden, die von ihm erwartet werden, z. B. aus einer Anmeldeinformationsdatei anstelle eines Instance-Profils. Das liegt daran, dass SSM Agent in der Reihenfolge nach Anmeldeinformationen sucht, die für die Standard-Anbieterkette für Anmeldeinformationen vorgeschrieben ist.

> **Note**
>
> Unter Linux und macOS wird SSM Agent als Root-Benutzer ausgeführt. Daher sind die Umgebungsvariablen und die Datei mit Anmeldeinformationen, nach denen SSM Agent bei diesem Prozess sucht, nur die des Stammbenutzers (`/root/.aws/credentials`). SSM Agent berücksichtigt bei der Suche nach Anmeldeinformationen weder die Umgebungsvariablen noch die Datei mit den Anmeldeinformationen anderer Benutzer der Instance.

Die Standard-Anbieterkette sucht in folgender Reihenfolge nach Anmeldeinformationen:

1. Umgebungsvariablen, wenn konfiguriert (`AWS_ACCESS_KEY_ID` und `AWS_SECRET_ACCESS_KEY`)
2. Freigegebene Anmeldeinformationen-Datei (`$HOME/.aws/credentials` für Linux und macOS oder `%USERPROFILE%\.aws\credentials` für Windows Server) mit Berechtigungen, die beispielsweise durch eine Hybrid-Aktivierung oder eine AWS CLI -Installation bereitgestellt wird.
3. Eine AWS Identity and Access Management (IAM) -Rolle für Aufgaben, wenn eine Anwendung vorhanden ist, die eine Amazon Elastic Container Service (Amazon ECS) -Aufgabendefinition oder RunTask API-Operation verwendet.
4. Ein einer Amazon EC2-Instance hinzugefügtes Instance-Profil.
5. Die für die Standardkonfiguration für die Host-Verwaltung gewählte IAM-Rolle.

Verwandte Informationen finden Sie in den folgenden Themen:

- Instanzprofile für EC2-Instances — Konfigurieren Sie die für Systems Manager erforderlichen Instanzberechtigungen
- Hybride Aktivierungen — Erstellen Sie eine Hybridaktivierung, um Knoten bei Systems Manager zu registrieren
- AWS CLI Anmeldeinformationen — Konfiguration und Einstellungen der Anmeldeinformationsdatei im Benutzerhandbuch AWS Command Line Interface
- Standard-Anbieterkette für Anmeldeinformationen – Festlegen von Anmeldeinformationen im AWS SDK for Go -Entwicklerhandbuch

> **Note**
>
> Dieses Thema im AWS SDK for Go -Entwicklerhandbuch beschreibt die Standard-Anbieterkette in Bezug auf das SDK for Go. Die gleichen Prinzipien gelten jedoch für die Auswertung von Anmeldeinformationen für SSM Agent.

Über das lokale ssm-user-Konto

Ab Version 2.3.50.0 von SSM Agent, erstellt der Agent ein lokales Benutzerkonto namens `ssm-user` und fügt es dem `/etc/sudoers.d`-Verzeichnis (Linux und macOS) bzw. der Administratorengruppe (Windows Server) hinzu. Auf Agenten-Versionen vor 2.3.612.0 wird das Konto beim ersten Start bzw. Neustart des SSM Agent nach der Installation erstellt. Auf Version 2.3.612.0 und höher wird das

`ssm-user`-Konto beim ersten Start einer Sitzung auf einer Instance erstellt. Dies `ssm-user` ist der Standard-Betriebssystembenutzer, wenn eine Sitzung gestartet wirdSession Manager, eine Fähigkeit von AWS Systems Manager. Sie können die Berechtigungen ändern, indem Sie `ssm-user` einer Gruppe mit weniger Berechtigungen hinzufügen oder die `sudoers`-Datei entsprechend ändern. Das `ssm-user`-Konto wird bei einer Deinstallation von SSM Agent nicht aus dem System entfernt.

Unter Windows Server legt der SSM Agent beim Start einer jeden Sitzung ein neues Passwort für das `ssm-user`-Konto fest. Auf Linux-verwalteten Instances werden keine Passwörter für `ssm-user` festgelegt.

Ab SSM Agent Version 2.3.612.0 wird das `ssm-user`-Konto nicht automatisch auf Windows Server-Maschinen erstellt, die als Domain-Controller verwendet werden. Um Session Manager auf einem Windows Server-Domain-Controller zu verwenden, erstellen Sie das `ssm-user`-Konto manuell, sofern es noch nicht vorhanden ist, und weisen dem Benutzer Domain-Administratorberechtigungen zu.

> **Important**
>
> Damit das `ssm-user`-Konto erstellt werden kann, muss das Instance-Profil, das der Instance zugewiesen ist, über die erforderlichen Berechtigungen verfügen. Weitere Informationen finden Sie unter Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager.

SSM Agent und die Instance Metadata Service (IMDS)

Systems Manager benötigt EC2-Instance-Metadaten, um korrekt zu funktionieren. Systems Manager kann entweder über Version 1 oder Version 2 desInstance Metadata Service (IMDSv1 und IMDSv2) auf Instance-Metadaten zugreifen. Ihre Instance muss auf die IPv4-Adresse des Instance-Metadatendienstes zugreifen können: 169.254.169.254. Weitere Informationen finden Sie unter Instance-Metadaten und Benutzerdaten im Amazon-EC2-Benutzerhandbuch.

Behalten SSM Agent up-to-date

Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen

automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisenGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.

> **Note**
>
> Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisen unterGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.
> Amazon Machine Images (AMIs), die standardmäßig SSM Agent enthalten, können bis zu zwei Wochen benötigen, bis ein aktualisiertes AMI mit der neuesten Version von SSM Agent veröffentlicht wird. Wir empfehlen, dass Sie sogar noch häufigere automatisierte Aktualisierungen für den SSM Agent konfigurieren

Sicherstellen, dass das SSM Agent-Installationsverzeichnis nicht geändert, verschoben oder gelöscht wird

SSM Agent ist unter `/var/lib/amazon/ssm/` (Linux und macOS) und `%PROGRAMFILES%\Amazon\SSM\` (Windows Server) installiert. Diese Installationsverzeichnisse enthalten wichtige Dateien und Ordner, die von SSM Agent verwendet werden, wie z. B. eine Anmeldeinformationsdatei, Ressourcen für die prozessübergreifende Kommunikation (IPC) und Orchestrierungsordner. Nichts im Installationsverzeichnis sollte geändert, verschoben oder gelöscht werden. Andernfalls funktioniert SSM Agent möglicherweise nicht mehr richtig.

SSM Agentfortlaufende Updates von AWS-Regionen

Nachdem ein SSM Agent Update in seinem GitHub Repository verfügbar gemacht wurde, kann es bis zu zwei Wochen dauern, bis die aktualisierte Version zu unterschiedlichen AWS-Regionen Zeiten für alle verfügbar ist. Aus diesem Grund erhalten Sie möglicherweise die Fehlermeldung „Auf der aktuellen Plattform nicht unterstützt“ oder „Aktualisierung amazon-ssm-agent auf eine ältere Version,

bitte aktivieren Sie die Option Downgrade zulassen, um fortzufahren“, wenn Sie versuchen, eine neue Version von SSM Agent in einer Region bereitzustellen.

Um die für Sie verfügbare Version von SSM Agent zu ermitteln, können Sie einen `curl`-Befehl ausführen.

Um die im globalen Download-Bucket verfügbare Version des Agenten anzuzeigen, führen Sie den folgenden Befehl aus.

```
curl https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/VERSION
```

Um die in einer bestimmten Region verfügbare Version des Agenten anzuzeigen, führen Sie den folgenden Befehl aus und ersetzen Sie dabei *`Region`* mit der Region, in der Sie arbeiten, z. B. `us-east-2` für Region USA Ost (Ohio).

```
curl https://s3.region.amazonaws.com/amazon-ssm-region/latest/VERSION
```

Sie können die `VERSION`-Datei auch direkt in Ihrem Browser ohne einen `curl`-Befehl öffnen.

SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets

Bei der Ausführung verschiedener Systems Manager Manager-Operationen greift AWS Systems Manager Agent (SSM Agent) auf eine Reihe von Amazon Simple Storage Service (Amazon S3) -Buckets zu. Diese S3-Buckets sind öffentlich zugänglich und SSM Agent verbindet sich standardmäßig über HTTP-Aufrufe mit ihnen.

Wenn Sie jedoch einen Virtual Private Cloud (VPC) -Endpunkt in Ihren Systems Manager-Vorgängen verwenden, müssen Sie in einem Amazon Elastic Compute Cloud (Amazon EC2) -Instanzprofil für Systems Manager oder in einer Servicerolle für Nicht-EC2-Maschinen in einer Hybrid - und Multi-Cloud-Umgebung eine ausdrückliche Genehmigung erteilen. Andernfalls haben Ihre Ressourcen keinen Zugriff auf diese öffentlichen Buckets.

Um Ihren verwalteten Knoten den Zugriff auf diese Buckets zu gewähren, wenn Sie einen VPC-Endpunkt verwenden, erstellen Sie eine benutzerdefinierte Richtlinie für Amazon S3-Berechtigungen und fügen diese dann Ihrem Instance-Profil (für EC2-Instances) oder Ihrer Servicerolle (für nicht-EC2-verwaltete Knoten) hinzu.

Informationen zur Verwendung eines VPC-Endpunkts (Virtual Private Cloud) in Ihren Systems Manager-Vorgängen finden Sie unter Verbessern der Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

> **Note**
>
> Diese Berechtigungen ermöglichen nur den Zugriff auf die AWS verwalteten Buckets, die für erforderlich sind. SSM Agent Sie erteilen nicht die erforderlichen Berechtigungen für andere Amazon S3-Operationen. Außerdem gewähren sie auch keine Berechtigung für Ihren eigenen S3-Buckets.

Weitere Informationen finden Sie unter den folgenden Themen:

- Konfigurieren Sie die für Systems Manager erforderlichen Instanzberechtigungen
- Erstellen Sie die für Systems Manager in Hybrid- und Multicloud-Umgebungen erforderliche IAM-Servicerolle

Inhalt

- Erforderliche Bucket-Berechtigungen
- Beispiel
- Validierung von hybrid-aktivierten Maschinen mit einem Hardware-Fingerabdruck
- SSM Agent auf GitHub

Erforderliche Bucket-Berechtigungen

In der folgenden Tabelle werden die einzelnen S3-Buckets beschrieben, auf die SSM Agent möglicherweise für Systems Manager-Operationen zugreifen muss.

> **Note**
>
> *`region`* steht für den Bezeichner für eine Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

Von SSM Agent erforderte Amazon S3-Berechtigungen

S3 Bucket-ARN	Beschreibung
`arn:aws:s3:::aws-windows-downloads-` *region*`/*`	Erforderlich für einige SSM-Dokumente, die nur Windows Server-Betriebssysteme unterstützen, sowie einige für plattformübergreifende Unterstützung, wie `AWSEC2-ConfigureSTIG` .
`arn:aws:s3:::amazon-ssm-` *region*`/*`	Erforderlich zum Aktualisieren von SSM Agent-Installationen. Diese Buckets enthalten die SSM Agent-Installationspakete und die Installationsmanifeste, auf die in dem `AWS-UpdateSSMAgent` -Dokument und Plugin verwiesen wird. Wenn diese Berechtigungen nicht bereitgestellt werden, führt der SSM Agent einen HTTP-Aufruf zum Herunterladen des Updates durch.
`arn:aws:s3:::amazon-ssm-packages-` *region*`/*`	Erforderlich für die Ausführung des SSM-Dokuments `AWS-ConfigureAWSPackage` durch ältere Versionen des SSM Agent als 2.2.45.0.
`arn:aws:s3:::` *region*`-birdwatcher-prod/*`	Bietet Zugriff auf den von Version 2.2.45.0 und höher des SSM Agent verwendeten Verteilungsservice. Mit diesem Service wird das Dokument `AWS-ConfigureAWSPackage` ausgeführt. Diese Genehmigung ist für alle AWS-Regionen außer der Region Afrika (Kapstadt) (af-south-1) und der Region Europa (Mailand) (eu-south-1) erforderlich.
`arn:aws:s3:::aws-ssm-distributor-file-` *region*`/*`	Bietet Zugriff auf den von Version 2.2.45.0 und höher des SSM Agent verwendeten Verteilungsservice. Mit diesem Service wird das SSM-

S3 Bucket-ARN	Beschreibung
	Dokument `AWS-ConfigureAWSPackage` ausgeführt. Diese Berechtigung wird für nur für die Region Afrika (Kapstadt) (af-south-1) und die Region Europa (Mailand) (eu-south-1) benötigt.
`arn:aws:s3:::aws-ssm-document-attachments- region/*`	Ermöglicht den Zugriff auf den S3-Bucket, der die Pakete fürDistributor, eine Fähigkeit von, enthält AWS Systems Manager, die Eigentum von sind. AWS

S3 Bucket-ARN	Beschreibung
`arn:aws:s3:::patch-baseline-snapshot-` *region*`/*`	Bietet Zugriff auf den S3-Bucket, der Patch-Baseline-Snapshots enthält. Dies ist erforderlich, wenn Sie eines der folgenden SSM-Dokumente verwenden: • `AWS-RunPatchBaseline` • `AWS-RunPatchBaselineAssociation` • `AWS-RunPatchBaselineWithHooks` • `AWS-ApplyPatchBaseline` (ein altes SSM-Dokument) **Note** In der Region Naher Osten (Bahrain) (me-south-1) nutzt dieser S3-Bucket eine andere Namenskonvention. Verwenden Sie nur in dieser AWS-Region stattdessen den folgenden Bucket. • `patch-baseline-snapshot-me-south-1-uduvl7q8` In der Region Afrika (Kapstadt) (af-south-1) nutzt dieser S3-Bucket eine andere Namenskonvention. Verwenden Sie nur in dieser AWS-Region stattdessen den folgenden Bucket. • `patch-baseline-snapshot-af-south-1-tbxdb5b9`

S3 Bucket-ARN	Beschreibung
Für von Linux und Windows Server verwaltete Knoten: `arn:aws:s3:::aws-ssm-` *region*`/*` Für Amazon EC2 Instances für macOS: `arn:aws:s3:::aws-patchmanager-macos-` *region*`/*`	Bietet Zugriff auf den S3-Bucket mit erforderlichen Modulen zur Verwendung mit bestimmten Systems Manager-Dokumenten (SSM-Dokumenten). Zum Beispiel: • `arn:aws:s3:::aws-ssm-us-east-2/*` • `aws-patchmanager-macos-us-east-2/*` Ausnahmen In einigen Fällen AWS-Regionen verwenden die S3-Bucket-Namen eine erweiterte Benennungskonvention, wie aus ihren ARNs hervorgeht. Verwenden Sie für diese Regionen stattdessen die folgenden ARNs: • Region Naher Osten (Bahrain) (me-south-1): `aws-patch-manager-me-south-1-a53fc9dce` • Region Afrika (Kapstadt) (af-south-1): `aws-patch-manager-af-south-1-bdd5f65a9` • Region Europa (Mailand) (eu-south-1): `aws-patch-manager-eu-south-1-c52f3f594` • Region Asien-Pazifik (Osaka) (ap-northeast-3): `aws-patch-manager-ap-northeast-3-67373598a` SSM-Dokumente

S3 Bucket-ARN	Beschreibung
	Im Folgenden finden Sie einige häufig verwendete SSM-Dokumente, die in diesen Buckets gespeichert sind. In `arn:aws:s3:::aws-ssm-` *region*/: • `AWS-RunPatchBaseline` • `AWS-RunPatchBaselineAssociation` • `AWS-RunPatchBaselineWithHooks` • `AWS-InstanceRebootWithHooks` • `AWS-ConfigureWindowsUpdate` • `AWS-FindWindowsUpdates` • `AWS-PatchAsgInstance` • `AWS-PatchInstanceWithRollback` • `AWS-UpdateSSMAgent` • `AWS-UpdateEC2Config` In `arn:aws:s3:::aws-patchmanager-macos-` *region*/: • `AWS-RunPatchBaseline` • `AWS-RunPatchBaselineAssociation` • `AWS-RunPatchBaselineWithHooks` • `AWS-InstanceRebootWithHooks` • `AWS-PatchAsgInstance` • `AWS-PatchInstanceWithRollback`

Beispiel

Das folgende Beispiel veranschaulicht, wie Zugriff auf die S3-Buckets erteilt wird, die in der Region USA Ost (Ohio) (us-east-2) für Systems Manager-Operationen benötigt werden. In den meisten Fällen müssen Sie diese Berechtigungen explizit in einem Instance-Profil oder einer Servicerolle nur dann bereitstellen, wenn Sie einen VPC Endpunkt verwenden.

> **Important**
>
> Wir empfehlen, dass Sie keine Platzhalterzeichen (*) für bestimmte Regionen in dieser Richtlinie verwenden. Verwenden Sie beispielsweise `arn:aws:s3:::aws-ssm-us-east-2/*` und nicht `arn:aws:s3:::aws-ssm-*/*`. Bei der Verwendung von Platzhaltern könnte Zugriff auf S3-Buckets erteilt werden, für die Sie keinen Zugriff gewähren möchten. Wenn Sie das Instance-Profil für mehr als eine Region verwenden möchten, empfehlen wir, den ersten `Statement`-Block für jede Region zu wiederholen.

```
{
        "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:::amazon-ssm-packages-us-east-2/*",
                    "arn:aws:s3:::us-east-2-birdwatcher-prod/*",
                    "arn:aws:s3:::aws-ssm-document-attachments-us-east-2/*",
                    "arn:aws:s3:::patch-baseline-snapshot-us-east-2/*",
                    "arn:aws:s3:::aws-ssm-us-east-2/*",
                    "arn:aws:s3:::aws-patchmanager-macos-us-east-2/*"
                ]
            }
        ]
    }
```

Validierung von hybrid-aktivierten Maschinen mit einem Hardware-Fingerabdruck

Bei Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung sammelt SSM Agent eine Reihe von Systemattributen (als Hardware-Hash bezeichnet) und verwendet diese Attribute, um einen Fingerabdruck zu berechnen. Der Fingerabdruck ist eine undurchsichtige Zeichenfolge, die der Agent an bestimmte Systems Manager-APIs weitergibt. Dieser eindeutige Fingerabdruck ordnet den Abrufer einem bestimmten hybrid-aktivierten verwalteten Knoten zu. Der Agent speichert den Fingerabdruck und den Hardware-Hash auf der lokalen Festplatte an einem Speicherort namens Vault.

Der Agent berechnet den Hardware-Hash und den Fingerabdruck, wenn die Maschine für die Verwendung mit Systems Manager registriert wird. Anschließend wird der Fingerabdruck an den Systems Manager-Service zurückgegeben, wenn der Agent einen `RegisterManagedInstance`-Befehl sendet.

Später, wenn ein `RequestManagedInstanceRoleToken`-Befehl gesendet wird, überprüft der Agent den Fingerabdruck und den Hardware-Hash im Vault, um sicherzustellen, dass die aktuellen Computerattribute mit dem gespeicherten Hardware-Hash übereinstimmen. Wenn die aktuellen Computerattribute mit dem im Vault gespeicherten Hardware-Hash übereinstimmen, übergibt der Agent den Fingerabdruck aus dem Vault an `RegisterManagedInstance`, was zu einem erfolgreichen Aufruf führt.

Wenn die aktuellen Computerattribute nicht mit dem gespeicherten Hardware-Hash übereinstimmen, berechnet SSM Agent einen neuen Fingerabdruck, speichert den neuen Hardware-Hash und Fingerabdruck im Vault und übergibt den neuen Fingerabdruck an `RequestManagedInstanceRoleToken`. Dadurch schlägt *`RequestManagedInstanceRoleToken`* fehl und der Agent kann kein Rollen-Token für die Verbindung mit dem Systems Manager-Service abrufen.

Dieser Fehler ist vorgesehen und wird als Verifizierungsschritt verwendet, um zu verhindern, dass mehrere verwaltete Knoten als derselbe verwaltete Knoten mit dem Systems-Manager-Service kommunizieren.

Beim Vergleich der aktuellen Computerattribute mit dem im Vault gespeicherten Hardware-Hash verwendet der Agent die folgende Logik, um festzustellen, ob die alten und neuen Hashes übereinstimmen:

- Wenn die SID (System/Maschinen-ID) anders ist, gibt es keine Übereinstimmung.
- Wenn die IP-Adresse identisch ist, gibt es eine Übereinstimmung.

- Andernfalls wird der Prozentsatz der übereinstimmenden Computerattribute berechnet und mit dem vom Benutzer konfigurierten Ähnlichkeitsschwellenwert verglichen, um festzustellen, ob eine Übereinstimmung vorliegt.

Der Ähnlichkeitsschwellenwert wird im Vault als Teil des Hardware-Hash gespeichert.

Der Ähnlichkeitsschwellenwert kann festgelegt werden, nachdem eine Instance mit einem Befehl wie dem folgenden registriert wurde.

Auf Linux-Maschinen:

```
sudo amazon-ssm-agent -fingerprint -similarityThreshold 1
```

Auf Windows Server Maschinen, die Folgendes verwenden PowerShell:

```
cd "C:\Program Files\Amazon\SSM\" `
    .\amazon-ssm-agent.exe -fingerprint -similarityThreshold 1
```

> **Important**
>
> Wenn sich eine der Komponenten zur Berechnung des Fingerprints ändert, kann dies dazu führen, dass der Agent in den Ruhezustand versetzt wird. Um diesen Ruhezustand zu vermeiden, setzen Sie den Ähnlichkeitsschwellenwert auf einen niedrigen Wert, z.B. **1**.

SSM Agent auf GitHub

Der Quellcode für SSM Agent ist auf verfügbar, GitHubsodass Sie den Agenten an Ihre Bedürfnisse anpassen können. Wir möchten Sie bitten, uns eventuelle Änderungswünsche mitzuteilen. Amazon Web Services bietet jedoch keine Unterstützung für die Ausführung modifizierter Kopien dieser Software.

Finden Sie AMIs mit dem SSM Agent vorinstallierten

AWS Systems Manager Agent (SSM Agent) ist auf einigen Amazon Machine Images (AMIs), die von Drittanbietern bereitgestellt werden AWS und denen sie vertrauen, vorinstalliert.

Wenn Sie beispielsweise eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance starten, die von einem beliebigen AMI mit einem der folgenden Betriebssysteme erstellt wurde, werden Sie wahrscheinlich feststellen, dass das SSM Agent bereits installiert ist:

- AlmaLinux
- Amazon Linux 1 Base AMIs von 2017.09 und höher
- Amazon Linux 2
- Amazon Linux 2 ECS-optimierte Basis-AMIs
- Amazon Linux 2023 (AL2023)
- Amazon-EKS-optimierte Amazon Linux-AMIs
- macOS10.14.x (Mojave), 10.15.x (Catalina), 11.x (Big Sur), 12.x (Monterey), 13.x (Ventura) und 14.x (Sonoma)
- SUSE Linux Enterprise Server (SLES) 12 und 15
- Ubuntu Server 16.04, 18.04, 20.04 und 22.04
- Windows Server 2008- bis 2012 R2-AMIs, die im November 2016 oder später veröffentlicht wurden
- Windows Server 2016, 2019 und 2022

> **Note**
>
> SSM Agentsind möglicherweise auf verwalteten Geräten vorinstalliert, die nicht auf dieser Liste stehen. AWS AMIs Dies weist normalerweise darauf hin, dass das Betriebssystem (OS) nicht vollständig von allen Systems-Manager-Funktionen unterstützt wird.
> SSM Agentist möglicherweise auch in AWS Marketplace oder im AMIs AMIs Community-Repository vorinstalliert, unterstützt diese aber AWS nicht. AMIs

Überprüfen des Status des SSM Agent

Je nachdem, wann diese initialisiert wurde, ist bei einer aus einem AMI in der vorhergehenden Liste erstellten Instance möglicherweise kein SSM Agent vorinstalliert. Es ist auch möglich, dass der Agent auf einer Instance vorinstalliert ist, der Agent jedoch nicht ausgeführt wird. Daher empfehlen wir Ihnen, den Status von SSM Agent zu überprüfen, bevor Sie versuchen, Systems Manager zum ersten Mal auf einer Instance zu verwenden.

Verwenden Sie das folgende Verfahren, um zu überprüfen, ob SSM Agent installiert ist und auf einer Instance ausgeführt wird. Wenn Sie feststellen, dass der Agent nicht installiert ist, können Sie ihn unter Linux, macOS und Windows Server-Instances manuell installieren.

So überprüfen Sie die Installation von SSM Agent auf einer Instance

1. Warten Sie nach dem Start einer neuen Instance einige Minuten, bis diese initialisiert ist.
2. Stellen Sie mit Ihrer bevorzugten Methode eine Verbindung zur Instance her. Sie können beispielsweise SSH verwenden, um eine Verbindung zu Linux-Instances herzustellen, oder Remote Desktop verwenden, um eine Verbindung zu Windows Server-Instances herzustellen.
3. Prüfen Sie den Status von SSM Agent, indem Sie den Befehl für den Betriebssystemtyp Ihrer Instance ausführen.

Betriebssystem	Befehl
Amazon Linux 1	`sudo status amazon-ssm-agent`
Amazon Linux 2 und Amazon Linux 2023	`sudo systemctl status amazon-ssm-agent`
macOS	Es gibt keinen Befehl zum Überprüfen des SSM Agent-Status auf macOS. Sie können den Status überprüfen, indem Sie die Agent-Protokolldatei `/var/log/amazon/ssm/amazon-ssm-agent.log` suchen und auswerten.
SUSE Linux Enterprise Server	`sudo systemctl status amazon-ssm-agent`
Ubuntu Server (32 Bit)	`sudo status amazon-ssm-agent`
Ubuntu Server (64-Bit – Deb)	`sudo systemctl status amazon-ssm-agent`

Betriebssystem	Befehl
Ubuntu Server (64-Bit – Snap)	`sudo systemctl status snap.amazon-ssm-agent.amazon-ssm-agent.service`
Windows Server	`Get-Service AmazonSSMAgent`

> **Tip**
>
> Informationen zu den Befehlen zur Überprüfung des SSM Agent-Status für alle von Systems Manager unterstützten Betriebssystemtypen finden Sie unter Prüfen des SSM Agent-Status und Starten des Agenten.

4. Werten Sie die Befehlsausgabe aus, um den Status von SSM Agent zu erfahren.

   Status: Installiert und ausgeführt

   In den meisten Fällen zeigt die Befehlsausgabe an, dass der Agent installiert ist und ausgeführt wird.

   Das folgende Beispiel zeigt, dass SSM Agent auf einer Amazon-Linux-2-Instance installiert ist und ausgeführt wird.

```
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--
```

   Das folgende Beispiel zeigt, dass SSM Agent auf einer Windows Server-Instance installiert ist und ausgeführt wird.

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

Status: Installiert, aber nicht ausgeführt

In einigen Fällen gibt die Befehlsausgabe an, dass der Agent installiert ist, aber nicht ausgeführt wird.

Das folgende Beispiel zeigt, dass SSM Agent auf einer Amazon-Linux-2-Instance installiert ist, aber nicht ausgeführt wird.

```
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--
```

Das folgende Beispiel zeigt, dass SSM Agent installiert ist, aber nicht auf einer Windows Server-Instance ausgeführt wird.

```
Status   Name               DisplayName
------   ----               -----------
Stopped  AmazonSSMAgent     Amazon SSM Agent
```

Wenn der Agent installiert ist, aber nicht ausgeführt wird, können Sie ihn manuell aktivieren, indem Sie die Befehle für den Betriebssystemtyp Ihrer Instace verwenden.

Betriebssystem	Befehl
Amazon Linux 1	`sudo start amazon-ssm-agent`
Amazon Linux 2 und Amazon Linux 2023	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`

Betriebssystem	Befehl
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 Bit)	`sudo start amazon-ssm-agent`
Ubuntu Server (64-Bit – Deb)	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
Ubuntu Server (64-Bit – Snap)	`sudo snap start amazon-ssm-agent`
Windows Server	Führen Sie den folgenden Befehl in PowerShell aus. `Start-Service AmazonSSMAgent`

Status: Nicht installiert

In einigen Fällen gibt die Befehlsausgabe an, dass der Agent nicht installiert ist.

Das folgende Beispiel zeigt, dass SSM Agent nicht auf einer Amazon-Linux-2-Instance installiert ist.

```
Unit amazon-ssm-agent.service could not be found.
```

Das folgende Beispiel zeigt, dass SSM Agent nicht auf einer Windows Server-Instance installiert ist.

```
Get-Service : Cannot find any service with service name 'AmazonSSMAgent'.
--truncated--
```

Wenn der Agent nicht installiert ist, können Sie ihn manuell installieren, indem Sie das Verfahren für Ihren Betriebssystemtyp verwenden:

- Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux
- Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für macOS
- Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Windows Server

Arbeiten mit SSM Agent auf EC2-Instances für Linux

AWS Systems Manager Agent (SSM Agent) verarbeitet Systems Manager Manager-Anfragen und konfiguriert Ihren Computer wie in der Anfrage angegeben. Verwenden Sie die Verfahren in den folgenden Themen, um SSM Agent auf Linux-Betriebssystemen zu installieren, zu konfigurieren oder zu deinstallieren.

Themen

- Verifizieren der Signatur von SSM Agent
- Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux
- Konfiguration SSM Agent für die Verwendung eines Proxys auf Linux-Knoten

Verifizieren der Signatur von SSM Agent

Die AWS Systems Manager Agent (SSM Agent) deb- und rpm-Installationspakete für Linux-Instances sind kryptografisch signiert. Sie können einen öffentlichen Schlüssel verwenden, um sicherzustellen, dass das Paket des Agenten original und unverändert ist. Wenn die Dateien beschädigt oder verändert werden, schlägt die Verifizierung fehl. Sie können die Signatur des Installer-Pakets entweder mit RPM oder GPG überprüfen. Die folgenden Informationen sind für SSM Agent-Versionen 3.1.1141.0 oder höher.

> **Important**
>
> Der öffentliche Schlüssel, der später in diesem Thema gezeigt wird, läuft am 17.02.2025 (17. Februar 2025) ab. Systems Manager wird in diesem Thema einen neuen öffentlichen Schlüssel veröffentlichen, bevor der alte abläuft. Wir empfehlen Ihnen, den RSS-Feed für dieses Thema zu abonnieren, um eine Benachrichtigung zu erhalten, wenn der neue Schlüssel verfügbar ist.

Die richtige Signaturdatei für die Architektur und das Betriebssystem Ihrer Instance finden Sie in der folgenden Tabelle.

region steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

Architektur	Betriebssystem	URL der Signaturdatei	Agent-Download-Dateiname
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`

Architektur	Betriebssystem	URL der Signaturdatei	Agent-Download-Dateiname
		amazon-ssm-agent.rpm.sig	
x86_64	Debian Server, Ubuntu Server	https://s3. *region*.amazonaws.com/amazon-ssm-*region*/latest/debian_amd64/amazon-ssm-agent.deb.sig https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_amd64/amazon-ssm-agent.deb.sig	amazon-ssm-agent.deb

Architektur	Betriebssystem	URL der Signaturdatei	Agent-Download-Dateiname
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`

Architektur	Betriebssystem	URL der Signaturdatei	Agent-Download-Dateiname
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`

Architektur	Betriebssystem	URL der Signaturdatei	Agent-Download-Dateiname
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`

Bevor Sie beginnen

Bevor Sie die Signatur von überprüfen könnenSSM Agent, müssen Sie das entsprechende Agentenpaket für Ihr Betriebssystem herunterladen. z. B. `https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_arm64/amazon-ssm-agent.rpm`. Weitere Informationen zum Herunterladen von SSM Agent Paketen finden Sie unterManuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.

GPG

So überprüfen Sie das SSM Agent-Paket auf einem Linux-Server

1. Kopieren Sie den folgenden Schlüssel und speichern Sie ihn in einer Datei mit dem Namen `amazon-ssm-agent.gpg`.

```
-----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. Importieren Sie den öffentlichen Schlüssel in Ihren Schlüsselbund und notieren Sie den zurückgegebenen Schlüsselwert.

```
gpg --import amazon-ssm-agent.gpg
```

3. Überprüfen Sie den Fingerprint Ersetzen Sie *key-value* durch den Wert des Schlüssels aus dem vorherigen Schritt. Es wird empfohlen, GPG zu verwenden, um den Fingerprint zu überprüfen, auch wenn Sie RPM verwenden, um das Installer-Paket zu überprüfen.

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

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
pub   2048R/97DD04ED 2023-08-28 [expires: 2025-02-17]
      Key fingerprint = DE92 C7DA 3E56 E923 31D6  2A36 BC1F 495C 97DD 04ED
uid                  SSM Agent <ssm-agent-signer@amazon.com>
```

Der Fingerabdruck sollte wie folgt aussehen.

```
DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
```

Wenn die Fingerabdruck-Zeichenfolge nicht übereinstimmt, installieren Sie den Agent nicht. Kontakt AWS Support.

4. Laden Sie die Signaturdatei entsprechend der Architektur und dem Betriebssystem Ihrer Instance herunter.
5. Überprüfen Sie die Installer-Paketsignatur. Achten Sie darauf, den *Signaturdateinamen* und *agent-download-filename*durch die Werte zu ersetzen, die Sie beim Herunterladen der Signaturdatei und des Agenten angegeben haben, wie in der Tabelle weiter oben in diesem Thema aufgeführt.

```
gpg --verify signature-filename agent-download-filename
```

Zum Beispiel für die x86_64-Architektur auf Amazon Linux 2:

```
gpg --verify amazon-ssm-agent.rpm.sig amazon-ssm-agent.rpm
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
gpg: Signature made Thu 31 Aug 2023 07:46:49 PM UTC using RSA key ID 97DD04ED
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: DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
```

Wenn die Ausgabe die Bezeichnung `BAD signature`enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben. Wenn Sie weiterhin diese Antwort erhalten, wenden Sie sich an den Agenten AWS Support und installieren Sie ihn nicht. Die Vertrauens-Warnmeldung bedeutet nicht, dass die Signatur ungültig ist, sondern nur, dass Sie den öffentlichen Schlüssel nicht überprüft haben. Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Wenn die Ausgabe die Phrase `Can't check signature:`

No `public key` enthält, überprüfen Sie, ob Sie SSM Agent Version 3.1.1141.0 oder höher heruntergeladen haben.

RPM

So überprüfen Sie das SSM Agent-Paket auf einem Linux-Server

1. Kopieren Sie den folgenden Schlüssel und speichern Sie ihn in einer Datei mit dem Namen `amazon-ssm-agent.gpg`.

```
-----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. Importieren Sie den öffentlichen Schlüssel in Ihren Schlüsselbund und notieren Sie den zurückgegebenen Schlüsselwert.

```
rpm --import amazon-ssm-agent.gpg
```

3. Überprüfen Sie den Fingerprint Ersetzen Sie *key-value* durch den Wert des Schlüssels aus dem vorherigen Schritt. Es wird empfohlen, GPG zu verwenden, um den Fingerprint zu überprüfen, auch wenn Sie RPM verwenden, um das Installer-Paket zu überprüfen.

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

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
pub   2048R/97DD04ED 2023-08-28 [expires: 2025-02-17]
      Key fingerprint = DE92 C7DA 3E56 E923 31D6  2A36 BC1F 495C 97DD 04ED
uid                  SSM Agent <ssm-agent-signer@amazon.com>
```

Der Fingerabdruck sollte wie folgt aussehen.

```
DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
```

Wenn die Fingerabdruck-Zeichenfolge nicht übereinstimmt, installieren Sie den Agent nicht. Kontakt AWS Support.

4. Überprüfen Sie die Installer-Paketsignatur. Achten Sie darauf, den *Signaturdateinamen* und *agent-download-filename*durch die Werte zu ersetzen, die Sie beim Herunterladen der Signaturdatei und des Agenten angegeben haben, wie in der Tabelle weiter oben in diesem Thema aufgeführt.

```
rpm --checksig signature-filename agent-download-filename
```

Zum Beispiel für die x86_64-Architektur auf Amazon Linux 2:

```
rpm --checksig amazon-ssm-agent.rpm.sig amazon-ssm-agent.rpm
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
amazon-ssm-agent-3.1.1141.0-1.amzn2.x86_64.rpm: rsa sha1 (md5) pgp md5 OK
```

Wenn `pgp` in der Ausgabe fehlt und Sie den öffentlichen Schlüssel importiert haben, ist der Agent nicht signiert. Wenn die Ausgabe die Phrase `NOT OK (MISSING KEYS: (MD5) key-id)` enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben, und überprüfen Sie, ob Sie SSM Agent Version 3.1.1141.0 oder höher heruntergeladen haben. Wenn Sie weiterhin diese Antwort erhalten, wenden Sie sich an den Agenten AWS Support und installieren Sie ihn nicht.

Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux

Lesen Sie die folgenden Informationen, bevor Sie AWS Systems Manager Agent (SSM Agent) manuell auf einem Amazon Elastic Compute Cloud (Amazon EC2) Linux-Betriebssystem installieren.

SSM Agent-URLs für Installationsdateien

Sie können auf die Installationsdateien zugreifenSSM Agent, die in jeder kommerziellen AWS-Region Version gespeichert sind. Wir stellen auch Installationsdateien in einem global verfügbaren Amazon Simple Storage Service (Amazon S3)-Bucket bereit, den Sie als Alternative oder Sicherungsquelle für Dateien verwenden können.

Wenn Sie den Agenten manuell auf einer oder zwei Instances installieren, können Sie die Befehle in der von uns bereitgestellten Schnellinstallation verwenden, um Zeit zu sparen. Die in diesen Verfahren bereitgestellten Befehle können auch als Skripte über Benutzerdaten an Amazon-EC2-Instances übergeben werden.

Wenn Sie ein Skript oder eine Vorlage für die Installation des Agenten auf mehreren Instances erstellen, empfehlen wir Ihnen, die Installationsdateien in oder in der Nähe einer AWS-Region an Ihrem geografischen Standort zu verwenden. Bei Masseninstallationen kann dies die Geschwindigkeit Ihrer Downloads erhöhen und die Latenz verringern. In diesen Fällen empfehlen wir die Verwendung der Verfahren im Abschnitt Create custom installation commands (Erstellen benutzerdefinierter Installationsbefehle) in den Installationsthemen.

Amazon Machine Images mit dem Agenten vorinstalliert

SSM Agentist auf einigen Amazon Machine Images (AMIs) vorinstalliert, die von AWS bereitgestellt werden. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Installation auf anderen Maschinentypen

Wenn Sie den Agenten auf einem lokalen Server oder einer virtuellen Maschine (VM) installieren müssen, damit er mit Systems Manager verwendet werden kann, finden Sie weitere Informationen unter So installieren Sie den SSM Agent auf Hybrid-Linux-Knoten. Weitere Informationen zum Installieren des Agenten auf Edge-Geräten finden Sie unter Verwaltung von Edge-Geräten mit Systems Manager.

Den Agenten auf dem neuesten Stand halten

Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisen aufGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.

Auswahl Ihres Betriebssystems

Um das Verfahren zur manuellen Installation von SSM Agent unter dem angegebenen Betriebssystem anzuzeigen wählen sie einen Link aus der folgenden Liste aus:

> **Note**
>
> Eine Liste der unterstützten Versionen der folgenden Betriebssysteme finden Sie unter Unterstützte Betriebssysteme für Systems Manager.

- AlmaLinux
- Amazon Linux 2 und Amazon Linux 2023
- Amazon Linux 1 1
- CentOS
- CentOS Stream
- Debian Server
- Oracle Linux
- Red Hat Enterprise Linux
- Rocky Linux

- SUSE Linux Enterprise Server
- Ubuntu Server

Deinstallieren des SSM Agent von Linux-Instances

Verwenden Sie den Paketmanager für Ihr Betriebssystem, um Linux-Instances zu SSM Agent deinstallieren. Je nach Betriebssystem ähnelt der Deinstallationsbefehl dem folgenden Beispielbefehl:

```
sudo dpkg -r amazon-ssm-agent
```

Manuelle Installation von SSM Agent auf AlmaLinux-Instances

Verwenden Sie die Informationen in diesem Abschnitt, um Sie bei der manuellen Installation oder Neuinstallation SSM Agent auf einer AlmaLinux Instanz zu unterstützen.

Bevor Sie beginnen

Beachten Sie vor SSM Agent der Installation auf einer AlmaLinux Instance Folgendes:

- Stellen Sie sicher, dass Python 3 auf Ihrer AlmaLinux Instanz installiert ist. Dies ist erforderlich, damit SSM Agent richtig funktioniert.
- Wichtige Informationen, die für die Installation von SSM Agent auf allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.

Themen

- Schnellinstallationsbefehle für SSM Agent on AlmaLinux
- Erstellen Sie benutzerdefinierte Agenteninstallationsbefehle für AlmaLinux Ihre Region

Schnellinstallationsbefehle für SSM Agent on AlmaLinux

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Bevor Sie beginnen

Beachten Sie vor SSM Agent der Installation auf einer AlmaLinux Instanz Folgendes:

- Stellen Sie sicher, dass Python 3 auf Ihrer AlmaLinux Instanz installiert ist. Dies ist erforderlich, damit SSM Agent richtig funktioniert.

Zur Installation SSM Agent auf AlmaLinux

1. Stellen Sie mit Ihrer bevorzugten Methode, AlmaLinux z. B. SSH, eine Connect zu Ihrer Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Auch wenn die URLs in den folgenden Befehlen ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für AlmaLinux.

   x86_64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_amd64/amazon-ssm-agent.rpm
   ```

   ARM64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_arm64/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

   ```
   # 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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen Sie benutzerdefinierte Agenteninstallationsbefehle für AlmaLinux Ihre Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallationsbefehle für SSM Agent on AlmaLinux weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelles Installieren von SSM Agent auf Instances von Amazon Linux 2 und Amazon Linux 2023

> **Important**
>
> Dieses Thema enthält Befehle für die Arbeit mit SSM Agent Amazon Linux 2- und Amazon Linux 2023-Instances. Einige dieser Befehle werden auf Amazon Linux 1-Instances nicht unterstützt. Bevor Sie fortfahren, überprüfen Sie, ob Sie das richtige Thema für Ihren Instance-Typ sehen. Befehle zur Ausführung auf Amazon Linux 1-Instances finden Sie unterManuelles Installieren SSM Agent auf Amazon Linux 1-Instances.

In den meisten Fällen ist AWS Systems Manager Agent Amazon Machine Images (AMIsSSM Agent) für Amazon Linux 2 und Amazon Linux 2023, die von AWS bereitgestellt werden, standardmäßig

vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Für den Fall, dass SSM Agent auf einer neuen Instance von Amazon Linux 2 oder Amazon Linux 2023 nicht vorinstalliert ist, oder wenn Sie den Agenten manuell neu installieren müssen, verwenden Sie die Informationen auf dieser Seite, um Ihnen zu helfen.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer Instance von Amazon Linux 2 oder Amazon Linux 2023 beachten Sie bitte Folgendes:

- Wichtige Informationen für die Installation von SSM Agent auf allen Linux-Betriebssystemen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.
- Wenn Sie einen `yum`-Befehl zum Aktualisieren des SSM Agent auf einem verwalteten Knoten verwenden, sehen Sie, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, möglicherweise die folgende Meldung: „Warning: RPMDB altered outside of yum.“ (Warnung: RPMDB wurde außerhalb von yum geändert.) Diese Meldung wird erwartet und kann ignoriert werden.

Themen

- Schnellinstallations-Befehle für SSM Agent auf Amazon Linux 2 oder Amazon Linux 2023
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Amazon Linux 2 oder Amazon Linux 2023 in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf Amazon Linux 2 oder Amazon Linux 2023

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent schnell mit Befehlen zum Kopieren und Einfügen auf Amazon Linux 2 oder Amazon Linux 2023

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung zu Ihrer Amazon-Linux-2- oder Amazon-Linux-2023-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für Amazon Linux 2 und 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. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
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--
```

Um den Agenten in diesen Fällen zu aktivieren, führen Sie den folgenden Befehl aus.

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Amazon Linux 2 oder Amazon Linux 2023 in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallationsbefehle für SSM Agent unter Amazon Linux 1 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *`region`* mit Ihren eigenen Informationen. Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelles Installieren SSM Agent auf Amazon Linux 1-Instances

> **Important**
>
> Amazon Linux 1 hat am 31. Dezember 2020 das Ende seines Standardsupports erreicht und am 31. Dezember 2023 das Ende seiner Lebensdauer erreicht, wie im Update zu AMI end-of-life Amazon Linux im AWS News-Blog angekündigt. AWS bietet Amazon Machine Images (AMIs) für dieses Betriebssystem nicht mehr an. AWS Systems Manager bietet jedoch weiterhin Support für bestehende Amazon Linux 1-Instances.
> Dieses Thema enthält Befehle für die Arbeit mit SSM Agent Amazon Linux 1-Instances. Einige dieser Befehle werden auf Amazon-Linux-2 und Amazon-Linux-2023-Instances nicht unterstützt. Bevor Sie fortfahren, überprüfen Sie, ob Sie das richtige Thema für Ihren Instance-Typ sehen. Informationen zu Befehlen für Instances von Amazon Linux 2 oder Amazon Linux 2023 finden Sie unter Manuelles Installieren von SSM Agent auf Instances von Amazon Linux 2 und Amazon Linux 2023.

In den meisten Fällen ist AWS Systems Manager Agent Amazon Machine Images (AMIsSSM Agent) für Amazon Linux 1, die von AWS bereitgestellt werden, standardmäßig vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Falls Sie den Agenten auf Amazon Linux 1 manuell neu installieren müssen, helfen Ihnen die Informationen auf dieser Seite.

Bevor Sie beginnen

Beachten Sie vor SSM Agent der Installation auf einer Amazon Linux 1-Instance Folgendes:

- Wichtige Informationen für die Installation von SSM Agent auf allen Linux-Betriebssystemen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.
- Wenn Sie einen `yum`-Befehl zum Aktualisieren des SSM Agent auf einem verwalteten Knoten verwenden, sehen Sie, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, möglicherweise die folgende Meldung: „Warning: RPMDB altered outside of yum.“ (Warnung: RPMDB wurde außerhalb von yum geändert.) Diese Meldung wird erwartet und kann ignoriert werden.

Themen

- Schnellinstallationsbefehle für SSM Agent unter Amazon Linux 1
- Erstellen Sie benutzerdefinierte Agenteninstallationsbefehle für Amazon Linux 1 in Ihrer Region

Schnellinstallationsbefehle für SSM Agent unter Amazon Linux 1

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Zur Installation SSM Agent auf Amazon Linux 1 mithilfe von Schnellbefehlen zum Kopieren und Einfügen

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Connect zu Ihrer Amazon Linux 1-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Auch wenn die URLs in den folgenden Befehlen ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für 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. (Empfohlen) Führen Sie den Befehl für die Architektur Ihrer Instance aus, um zu überprüfen, ob der Agent ausgeführt wird.

   x86_64 und x86

   ```
   sudo status amazon-ssm-agent
   ```

   ARM64

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie in folgenden Beispielen gezeigt.

   x86_64 und 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--
   ```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie in den folgenden Beispielen gezeigt.

x86_64 und 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--
```

Um den Agenten in diesen Fällen zu aktivieren, führen Sie den Befehl für die Architektur Ihrer Instance aus.

x86_64 und x86

```
sudo start amazon-ssm-agent
```

ARM64

```
sudo systemctl start amazon-ssm-agent
```

Erstellen Sie benutzerdefinierte Agenteninstallationsbefehle für Amazon Linux 1 in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallationsbefehle für SSM Agent unter Amazon Linux 1 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf CentOS-Instances

Die Amazon Machine Images (AMIs) für CentOS, die von bereitgestellt werden, sind standardmäßig AWS nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Eine Liste von AWS -verwalteten AMIs, auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer CentOS-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer CentOS-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen für die Installation von SSM Agent auf allen Linux-Betriebssystemen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.
- Wenn Sie einen `yum`-Befehl zum Aktualisieren des SSM Agent auf einem verwalteten Knoten verwenden, sehen Sie, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, möglicherweise die folgende Meldung: „Warning: RPMDB altered outside of yum.“ (Warnung: RPMDB wurde außerhalb von yum geändert.) Diese Meldung wird erwartet und kann ignoriert werden.

Themen

- So installieren Sie SSM Agent auf CentOS 8.x
- So installieren Sie SSM Agent auf CentOS 7.x
- So installieren Sie SSM Agent auf CentOS 6.x

So installieren Sie SSM Agent auf CentOS 8.x

Die von AWS bereitgestellten Amazon Machine Images (AMIs) für CentOS 8 sind standardmäßig nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Verwenden Sie die Informationen auf dieser Seite, um den Agenten auf CentOS-8-Instances zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer CentOS-8-Instance beachten Sie bitte Folgendes:

- Stellen Sie sicher, dass Python 2 oder Python 3 auf Ihrer CentOS 8-Instance installiert ist. Dies ist erforderlich, damit SSM Agent richtig funktioniert.

Themen

- Schnelle Installationsbefehle für SSM Agent auf CentOS 8
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS 8 in Ihrer Region

Schnelle Installationsbefehle für SSM Agent auf CentOS 8

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent auf CentOS 8.x

1. Stellen Sie mithilfe Ihrer bevorzugten Methode. z. B. SSH, eine Verbindung zu Ihrer CentOS-8-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für CentOS 8.

x86_64-Instances

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

ARM64-Instances

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; disabled; vend>
   Active: inactive (dead)
            --truncated--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS 8 in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnelle Installationsbefehle für SSM Agent auf CentOS 8 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Systems-Manager-Service-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

So installieren Sie SSM Agent auf CentOS 7.x

Die von AWS bereitgestellten Amazon Machine Images (AMIs) für CentOS 7 sind standardmäßig nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Verwenden Sie die Informationen auf dieser Seite, um den Agenten auf CentOS-7-Instances zu installieren oder neu zu installieren.

Themen

- Schnellinstallations-Befehle für SSM Agent auf CentOS 7

- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS 7 in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf CentOS 7

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent auf CentOS 7.x

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung zu Ihrer CentOS-7-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für CentOS 7.

   x86_64-Instances

   ```
   sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_amd64/amazon-ssm-agent.rpm
   ```

   ARM64-Instances

   ```
   sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_arm64/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

   ```
   # 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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS 7 in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf CentOS 7 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Systems-Manager-Service-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

So installieren Sie SSM Agent auf CentOS 6.x

Die von AWS bereitgestellten Amazon Machine Images (AMIs) für CentOS 6 sind standardmäßig nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Verwenden Sie die Informationen auf dieser Seite, um den Agenten auf CentOS-6-Instances zu installieren oder neu zu installieren.

Themen

- Schnelle Installationsbefehle für SSM Agent auf CentOS 6
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS 6 in Ihrer Region

Schnelle Installationsbefehle für SSM Agent auf CentOS 6

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent auf CentOS 6.x

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung zu Ihrer CentOS-6-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für CentOS 6.
   > Die folgenden Befehle geben das Versionsverzeichnis `3.0.1479.0` anstelle eines `latest`-Verzeichnisses an. Das liegt daran, dass SSM Agent Version 3.1 und höher für CentOS 6 nicht unterstützt wird.

   x86_64-Instances

   ```
   sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/
   SSMAgent/3.0.1479.0/linux_amd64/amazon-ssm-agent.rpm
   ```

   x86-Instances

   ```
   sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/
   SSMAgent/3.0.1479.0/linux_386/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

   ```
   sudo status amazon-ssm-agent
   ```

   In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

   ```
   amazon-ssm-agent start/running, process 1744
   ```

   In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
amazon-ssm-agent stop/waiting
```

Um den Agenten in diesen Fällen zu aktivieren, führen Sie den folgenden Befehl aus.

```
sudo start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS 6 in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnelle Installationsbefehle für SSM Agent auf CentOS 6 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *`region`* mit Ihren eigenen Informationen. Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Systems-Manager-Service-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

> **Note**
>
> Die folgenden Befehle geben das Versionsverzeichnis `3.0.1390.0` anstelle eines `latest`-Verzeichnisses an. Das liegt daran, dass SSM Agent Version 3.1 und höher für CentOS 6 nicht unterstützt wird.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/3.0.1479.0/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Manuelle Installation von SSM Agent auf CentOS Stream-Instances

Die Amazon Machine Images (AMIs) dafürCentOS Stream, die von bereitgestellt werden, sind standardmäßig AWS nicht mit vorinstalliertem AWS Systems Manager Agent (SSM Agent) ausgestattet. Eine Liste von AWS -verwalteten AMIs, auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer CentOS Stream-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer CentOS Stream-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent auf allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.

Themen

- Schnellinstallations-Befehle für SSM Agent auf CentOS Stream
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS Stream in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf CentOS Stream

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer CentOS Stream-Instance beachten Sie bitte Folgendes:

- Stellen Sie sicher, dass Python 2 oder Python 3 auf Ihrer CentOS Stream-8-Instance installiert ist. Dies ist erforderlich, damit SSM Agent richtig funktioniert.

Installieren Sie den SSM Agent auf CentOS Stream wie folgt

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer CentOS Stream-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für CentOS Stream.

   x86_64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_amd64/amazon-ssm-agent.rpm
   ```

   ARM64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_arm64/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

   ```
   sudo systemctl status amazon-ssm-agent
   ```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS Stream in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf CentOS Stream weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf Debian Server-Instances

Die Amazon Machine Images (AMIs) dafürDebian Server, die von bereitgestellt werden, sind standardmäßig AWS nicht mit vorinstalliertem AWS Systems Manager Agent (SSM Agent) ausgestattet. Eine Liste von AWS -verwalteten AMIs, auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer Debian Server-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer Debian Server-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent auf allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.

Themen

- Schnellinstallations-Befehle für SSM Agent auf Debian Server
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Debian Server in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf Debian Server

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Installieren Sie den SSM Agent auf Debian Server wie folgt

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Debian Server-Instance her.
2. Führen Sie den folgenden Befehl aus, um ein temporäres Verzeichnis auf der Instance zu erstellen.

   ```
   mkdir /tmp/ssm
   ```

3. Führen Sie den folgenden Befehl aus, um in das temporäre Verzeichnis zu wechseln.

   ```
   cd /tmp/ssm
   ```

4. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für Debian Server. Für Debian Server 8 wird nur die x86_64-Architektur unterstützt.

x86_64-Instances

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
debian_amd64/amazon-ssm-agent.deb
```

ARM64-Instances

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
debian_arm64/amazon-ssm-agent.deb
```

5. Führen Sie den folgenden Befehl aus.

```
sudo dpkg -i amazon-ssm-agent.deb
```

6. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Debian Server in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf Debian Server weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *`region`* mit Ihren eigenen Informationen. Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

> **Note**
>
> Für Debian Server 8 wird nur die x86_64-Architektur unterstützt.

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
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Manuelle Installation von SSM Agent auf Oracle Linux-Instances

Die Amazon Machine Images (AMIs) dafürOracle Linux, die von bereitgestellt werden, sind standardmäßig AWS nicht mit vorinstalliertem AWS Systems Manager Agent (SSM Agent)

ausgestattet. Eine Liste von AWS -verwalteten AMIs, auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer Oracle Linux-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer Oracle Linux-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen für die Installation von SSM Agent auf allen Linux-Betriebssystemen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.
- Wenn Sie einen `yum`-Befehl zum Aktualisieren des SSM Agent auf einem verwalteten Knoten verwenden, sehen Sie, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, möglicherweise die folgende Meldung: „Warning: RPMDB altered outside of yum.“ (Warnung: RPMDB wurde außerhalb von yum geändert.) Diese Meldung wird erwartet und kann ignoriert werden.

Themen

- Schnellinstallations-Befehle für SSM Agent auf Oracle Linux
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Oracle Linux in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf Oracle Linux

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent schnell mit Befehlen zum Kopieren und Einfügen auf Oracle Linux

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Oracle Linux-Instance her.
2. Kopieren Sie den folgenden Befehl und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl die URL im folgenden Befehl ein `ec2-downloads-windows`-Verzeichnis enthält, sind dies die richtigen globalen Installationsdateien für Oracle Linux.

x86_64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
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--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Oracle Linux in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf Oracle Linux weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *`region`* mit Ihren eigenen Informationen. Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf Red Hat Enterprise Linux-Instances

Die von bereitgestellten Amazon Machine Images Red Hat Enterprise Linux (AMIsRHEL) for () sind standardmäßig AWS nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Eine Liste der AWS verwalteten Programme, AMIs auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter. Finden Sie AMIs mit dem SSM Agent vorinstallierten

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer RHEL-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer RHEL-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen für die Installation von SSM Agent auf allen Linux-Betriebssystemen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.
- Wenn Sie einen `yum`-Befehl zum Aktualisieren des SSM Agent auf einem verwalteten Knoten verwenden, sehen Sie, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, möglicherweise die folgende Meldung: „Warning: RPMDB altered outside of yum.“ (Warnung: RPMDB wurde außerhalb von yum geändert.) Diese Meldung wird erwartet und kann ignoriert werden.

Themen

- Installieren Sie SSM Agent auf RHEL 8.x und 9.x
- So installieren Sie SSM Agent auf RHEL 7.x
- So installieren Sie SSM Agent auf RHEL 6.x

Installieren Sie SSM Agent auf RHEL 8.x und 9.x

Die von bereitgestellten Amazon Machine Images AWS (AMIs) für RHEL 8 und 9 sind standardmäßig nicht mit dem - AWS Systems Manager Agent (SSM Agent) vorinstalliert. Verwenden Sie die Informationen auf dieser Seite, um den Agenten auf RHEL-8- und 9-Instances zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer RHEL-8- oder 9-Instance beachten Sie bitte Folgendes:

- Stellen Sie sicher, dass Python 2 oder Python 3 auf Ihrer RHEL-8- oder 9-Instance installiert ist. Dies ist erforderlich, damit SSM Agent richtig funktioniert.

Themen

- Schnellinstallations-Befehle für SSM Agent auf RHEL 8 oder -9
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 8 und 9 in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf RHEL 8 oder -9

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Zur Installation von SSM Agent auf RHEL 8.x oder 9.x

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer RHEL-8- oder 9-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für RHEL 8 und 9.

   x86_64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_amd64/amazon-ssm-agent.rpm
   ```

   ARM64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_arm64/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

   ```
   # 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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 8 und 9 in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf RHEL 8 oder -9 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

So installieren Sie SSM Agent auf RHEL 7.x

Die von AWS bereitgestellten Amazon Machine Images (AMIs) für RHEL 7 sind standardmäßig nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Verwenden Sie die Informationen auf dieser Seite, um den Agenten auf RHEL-7-Instances zu installieren oder neu zu installieren.

Themen

- Schnellinstallations-Befehle für SSM Agent auf RHEL 7
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 7 in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf RHEL 7

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent auf RHEL 7.x

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer RHEL-7-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten sind, dies die richtigen globalen Installationsdateien für RHEL 7.

   x86_64-Instances

   ```
   sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_amd64/amazon-ssm-agent.rpm
   ```

   ARM64-Instances

   ```
   sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_arm64/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

   ```
   # 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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 7 in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf RHEL 7 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *`region`* mit Ihren eigenen Informationen. Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Systems-Manager-Service-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

So installieren Sie SSM Agent auf RHEL 6.x

Die von AWS bereitgestellten Amazon Machine Images (AMIs) für RHEL 6 sind standardmäßig nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Verwenden Sie die Informationen auf dieser Seite, um den Agenten auf RHEL-6-Instances zu installieren oder neu zu installieren.

Themen

- Schnellinstallations-Befehle für SSM Agent auf RHEL 6
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 6 in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf RHEL 6

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent auf RHEL 6.x

1. Stellen Sie mithilfe Ihrer bevorzugten MEthode, z. B. SSH, eine Verbindung mit Ihrer RHEL-6-Instance her.

2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für RHEL 6.
> Die folgenden Befehle geben das Versionsverzeichnis `3.0.1479.0` anstelle eines `latest`-Verzeichnisses an. Das liegt daran, dass SSM Agent Version 3.1 und höher für RHEL 6 nicht unterstützt wird.

x86_64-Instances

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/
SSMAgent/3.0.1479.0/linux_amd64/amazon-ssm-agent.rpm
```

x86-Instances

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/
SSMAgent/3.0.1479.0/linux_386/amazon-ssm-agent.rpm
```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
amazon-ssm-agent start/running, process 1788
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
amazon-ssm-agent stop/waiting
```

Um den Agenten in diesen Fällen zu aktivieren, führen Sie den folgenden Befehl aus.

```
sudo start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 6 in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf RHEL 6 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *`region`* mit Ihren eigenen Informationen. Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Systems-Manager-Service-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

> **Note**
>
> Die folgenden Befehle geben das Versionsverzeichnis `3.0.1390.0` anstelle eines `latest`-Verzeichnisses an. Das liegt daran, dass SSM Agent Version 3.1 und höher für RHEL 6 nicht unterstützt wird.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/3.0.1479.0/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Manuelle Installation von SSM Agent auf Rocky Linux-Instances

Die Amazon Machine Images (AMIs) dafürRocky Linux, die von bereitgestellt werden, sind standardmäßig AWS nicht mit vorinstalliertem AWS Systems Manager Agent (SSM Agent) ausgestattet. Eine Liste von AWS -verwalteten AMIs, auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer Rocky Linux-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer Rocky Linux-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent auf allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.

Themen

- Schnellinstallations-Befehle für SSM Agent auf Rocky Linux
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Rocky Linux in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf Rocky Linux

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer Rocky Linux-Instance beachten Sie bitte Folgendes:

- Stellen Sie sicher, dass Python 2 oder Python 3 auf Ihrer Rocky Linux-Instance installiert ist. Dies ist erforderlich, damit SSM Agent richtig funktioniert.

Installieren Sie den SSM Agent auf Rocky Linux wie folgt

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Rocky Linux-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für Rocky Linux.

   x86_64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_amd64/amazon-ssm-agent.rpm
   ```

   ARM64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_arm64/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

   ```
   sudo systemctl status amazon-ssm-agent
   ```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Rocky Linux in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf Rocky Linux weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf SUSE Linux Enterprise Server-Instances

In den meisten Fällen sind die von bereitgestellten Amazon Machine Images SUSE Linux Enterprise Server (AMIsSLES) für () AWS standardmäßig mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Für den Fall, dass SSM Agent auf einer neuen SLES-Instance nicht vorinstalliert ist, oder wenn Sie den Agenten manuell neu installieren müssen, verwenden Sie die Informationen auf dieser Seite, um Ihnen zu helfen.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer SLES-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent auf allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.

Themen

- Schnellinstallations-Befehle für SSM Agent auf SLES
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für SLES in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf SLES

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent schnell mit Befehlen zum Kopieren und Einfügen auf SLES

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer SLES-Instance her.
2. Option 1: Benutzen Sie einen `zypper`-Befehl:

   - Führen Sie den folgenden Befehl aus:

     ```
     sudo zypper install amazon-ssm-agent
     ```

   - Geben Sie `y` als Reaktion auf alle Eingabeaufforderungen ein.

   Option 2: Benutzen Sie einen `rpm`-Befehl.

   - Erstellen Sie ein temporäres Verzeichnis auf der Instance.

     ```
     mkdir /tmp/ssm
     ```

- Ändern Sie das temporäre Verzeichnis.

```
cd /tmp/ssm
```

- Führen Sie nacheinander die folgenden Befehle aus, um das SSM Agent-Installationsprogramm herunterzuladen und es auszuführen.

> **Note**
>
> Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für SLES.

x86_64-Instances:

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/
amazon-ssm-agent.rpm
```

ARM64-Instances:

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_arm64/
amazon-ssm-agent.rpm
```

- Führen Sie den folgenden Befehl aus.

```
sudo rpm --install amazon-ssm-agent.rpm
```

- (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# 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--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für SLES in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallationsbefehle für SSM Agent unter Amazon Linux 1 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

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
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Manuelle Installation von SSM Agent auf Ubuntu Server-Instances

> **Important**
>
> Stellen Sie vor der Installation von SSM Agent auf einer 64-Bit-Version von Ubuntu Server sicher, dass Sie die richtigen Installationstools verwenden. Beginnend mit Amazon Machine Images (AMIs), die mit 20180627 identifiziert werden, ist SSM Agent unter Verwendung von Snap-Paketen auf Version 16.04 vorinstalliert. Auf Instances, die aus früheren AMIs erstellt wurden, muss SSM Agent mit Deb-Installationsprogramm-Paketen installiert werden. Weitere Informationen finden Sie unter Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04.

In den meisten Fällen werden die Amazon Machine Images (AMIs) dafür von AWS bereitgestelltUbuntu Server, dass AWS Systems Manager Agent (SSM Agent) standardmäßig vorinstalliert ist. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Für den Fall, dass SSM Agent auf einer neuen Ubuntu Server-Instance nicht vorinstalliert ist, oder wenn Sie den Agenten manuell neu installieren müssen, verwenden Sie die Informationen in diesem Abschnitt, um Ihnen zu helfen.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer Ubuntu Server-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent auf allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux.

Themen

- Installieren Sie SSM Agent auf Ubuntu Server 22.04 LTS, 20.10 STR & 20.04, 18.04 und 16.04 LTS 64-bit (Snap)
- Installieren Sie SSM Agent auf Ubuntu Server 16.04 und 14.04 64-Bit (deb)
- Installieren Sie SSM Agent auf Ubuntu Server 16.04 und 14.04 32-Bit (deb)
- Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04

Installieren Sie SSM Agent auf Ubuntu Server 22.04 LTS, 20.10 STR & 20.04, 18.04 und 16.04 LTS 64-bit (Snap)

Bevor Sie beginnen

Beachten Sie vor der Installation von SSM Agent auf einem Ubuntu Server 22.04 LTS, 20.10 STR und 20.04, 18.04 und 16.04 LTS 64-Bit (Snap) Folgendes:

Installation von Version 16.04 durch Snaps oder Deb-Installationsprogramme

Auf Ubuntu Server 16.04, wird der SSM Agent, abhängig von der Version des 16.04-AMI, entweder mit Snaps oder mit deb-Installationspaketen installiert.

Speicherorte von SSM Agent-Installationsprogramm-Dateien

Auf Ubuntu Server 22.04 LTS, 20.10 STR und 20.04, 18.04 und 16.04 LTS (mit Snap) werden SSM Agent-Installationsprogramm-Dateien, einschließlich Agentenbinär- und Konfigurationsdateien, im folgenden Verzeichnis gespeichert: `/snap/amazon-ssm-agent/current/`. Wenn Sie Änderungen an einer Konfigurationsdatei in diesem Verzeichnis vornehmen, müssen Sie dies Datei aus dem Verzeichnis `/snap` in das Verzeichnis `/etc/amazon/ssm/` kopieren. Protokoll- und Bibliotheksdateien wurden nicht geändert (`/var/lib/amazon/ssm`, `/var/log/amazon/ssm`).

Verwenden des Snap-`candidate`-Kanals

Der Kandidat-Kanal im Snap Store enthält die neueste Version von SSM Agent (einschließlich aller neuesten Fehlerbehebungen); nicht den stabilen Kanal. Weitere Informationen zu den Unterschieden zwischen den Kandidaten und stabilen Kanälen finden Sie unter Risikostufen auf https://snapcraft.io/docs/channels.

Wenn Sie SSM Agent-Versionsinformationen auf dem Kandidatenkanal verfolgen, führen Sie den folgenden Befehl auf Ihren Ubuntu Server 20.10 STR und 20.04, 18.04 und 16.04 LTS 64-Bit-Instances aus.

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

Für Version 18.04 und höher empfohlene Snaps

Auf Ubuntu Server 22.04 LTS, 20.10 STR und 20.04 und 18.04 LTS empfehlen wir, nur Snaps zu verwenden. Stellen Sie außerdem sicher, dass nur eine Instance des Agenten auf Ihren Instances installiert ist und ausgeführt wird. Wenn Sie SSM Agent ohne Snaps verwenden möchten, deinstallieren Sie den SSM Agent. Installieren Sie dann SSM Agent als Debian-Paket gemäß den

Anweisungen zum Installieren von SSM Agent auf Ubuntu Server 16.04 und 14.04 64-Bit (deb). Stellen Sie vor der Installation sicher, dass Sie keine Snaps installiert haben, die sich mit der Liste der Pakete überschneiden, die Sie als Debian-Pakete verwalten möchten.

`Maximum timeout exceeded`-Fehlermeldung

Aufgrund eines bekannten Problems mit Snap sehen Sie bei `snap`-Befehlen möglicherweise einen `Maximum timeout exceeded`-Fehler. Wenn Sie diese Fehlermeldung erhalten, führen Sie die folgenden Befehle nacheinander aus, um den Agenten zu starten, zu stoppen und seinen Status zu überprüfen:

```
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
```

So installieren Sie SSM Agent auf Ubuntu Server 22.04 LTS, 20.10 STR und 20.04, 18.04 und 16.04 LTS-64-Bit-Instances (mit dem Snap-Paket)

1. SSM Agent wird standardmäßig auf Ubuntu Server 22.04 LTS-, 20.04-, 18.04- und 16.04-LTS-64-Bit-AMIs mit der Kennung `20180627` oder höher erstellt.

   Sie können das folgende Skript verwenden, wenn Sie SSM Agent auf einem On-Premises-Server installieren oder wenn Sie den Agenten neu installieren möchten. Sie müssen keine URL für den Download angeben, da der `snap`-Befehl den Agenten automatisch aus dem Snap App Store unter https://snapcraft.io herunterlädt.

   ```
   sudo snap install amazon-ssm-agent --classic
   ```

2. Führen Sie den folgenden Befehl aus, um sicherzustellen, dass der SSM Agent ausgeführt wird.

   ```
   sudo snap list amazon-ssm-agent
   ```

3. Führen Sie den folgenden Befehl aus, um den Service zu starten, wenn der vorherige Befehl die Meldung `amazon-ssm-agent is stopped`, `inactive` oder `disabled` zurückgibt.

   ```
   sudo snap start amazon-ssm-agent
   ```

4. Prüfen Sie den Status des Agents.

```
sudo snap services amazon-ssm-agent
```

Installieren Sie SSM Agent auf Ubuntu Server 16.04 und 14.04 64-Bit (deb)

> **Important**
>
> Stellen Sie vor der Installation von SSM Agent auf einer 64-Bit-Version von Ubuntu Server sicher, dass Sie die richtigen Installationstools verwenden. Beginnend mit Amazon Machine Images (AMIs), die mit 20180627 identifiziert werden, ist SSM Agent unter Verwendung von Snap-Paketen auf Version 16.04 vorinstalliert. Auf Instances, die aus früheren AMIs erstellt wurden, muss SSM Agent mit Deb-Installationsprogramm-Paketen installiert werden. Weitere Informationen finden Sie unter Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04. Wenn SSM Agent in Verbindung mit einem Snap auf Ihrer Instance installiert ist und Sie SSM Agent mithilfe eines deb-Installationsprogramm-Pakets installieren oder aktualisieren, schlagen die Installation oder SSM Agent-Vorgänge möglicherweise fehl.

In den meisten Fällen ist auf den Amazon Machine Images (AMIs) Ubuntu Server 16.04, die von AWS bereitgestellt werden, AWS Systems Manager Agent (SSM Agent) standardmäßig vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Falls SSM Agent auf einer neuen Ubuntu Server-16.04-Instance vor Version 20180627 nicht vorinstalliert ist, Sie auf Ubuntu Server 14.04 installieren oder den Agenten manuell neu installieren müssen, verwenden Sie die Informationen auf dieser Seite, um Ihnen zu helfen.

Schnellinstallations-Befehle für SSM Agent auf Ubuntu Server 16.04 und 14.04 64-Bit (deb)

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent schnell mit Befehlen zum Kopieren und Einfügen auf Ubuntu Server 16.04 und 14.04 64-Bit (deb)

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Ubuntu Server-Instance her.

2. Führen Sie den folgenden Befehl aus, um ein temporäres Verzeichnis auf der Instance zu erstellen.

```
mkdir /tmp/ssm
```

3. Ändern Sie das temporäre Verzeichnis.

```
cd /tmp/ssm
```

4. Führen Sie die folgenden Befehle aus.

> **Note**
>
> Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für Ubuntu Server 16.04 und 14,04 64-Bit.

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_amd64/
amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

5. (Empfohlen) Führen Sie einen der folgenden Befehle aus, um festzustellen, ob der SSM Agent ausgeführt wird.

   Ubuntu Server 16.04

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   Ubuntu Server14.04

   ```
   sudo status amazon-ssm-agent
   ```

   In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird.

   In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

6. Führen Sie einen der folgenden Befehle aus, um den Service zu starten, wenn der vorherige Befehl die Meldung `amazon-ssm-agent is stopped`, `inactive` oder `disabled` zurückgibt.

   Ubuntu Server 16.04:

   ```
   sudo systemctl enable amazon-ssm-agent
   ```

   Ubuntu Server 14.04:

   ```
   sudo start amazon-ssm-agent
   ```

Erstellen Sie benutzerdefinierte Installationsbefehle für SSM Agent auf Ubuntu Server 16.04 und 14.04 64-Bit (deb) in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf Ubuntu Server 16.04 und 14.04 64-Bit (deb) weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *`region`* mit Ihren eigenen Informationen. Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_amd64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Installieren Sie SSM Agent auf Ubuntu Server 16.04 und 14.04 32-Bit (deb)

In den meisten Fällen ist auf den Amazon Machine Images (AMIs) Ubuntu Server 16.04, die von AWS bereitgestellt werden, AWS Systems Manager Agent (SSM Agent) standardmäßig vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Für den Fall, dass SSM Agent auf einer neuen Ubuntu Server 16.04-Instance nicht vorinstalliert ist, Sie auf Ubuntu Server 14.04 installieren oder den Agenten manuell neu installieren müssen, verwenden Sie die Informationen auf dieser Seite, um Ihnen zu helfen.

Schnellinstallations-Befehle für SSM Agent auf Ubuntu Server 16.04 und 14.04 32-Bit (deb)

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent schnell mit Befehlen zum Kopieren und Einfügen auf Ubuntu Server 16.04 und 14.04 32-Bit (deb)

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Ubuntu Server-Instance her.
2. Führen Sie den folgenden Befehl aus, um ein temporäres Verzeichnis auf der Instance zu erstellen.

   ```
   mkdir /tmp/ssm
   ```

3. Ändern Sie das temporäre Verzeichnis.

   ```
   cd /tmp/ssm
   ```

4. Führen Sie die folgenden Befehle aus.

> **Note**
>
> Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für Ubuntu Server 16.04 und 14.04 32-Bit.

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_386/
amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

5. (Empfohlen) Führen Sie einen der folgenden Befehle aus, um festzustellen, ob der SSM Agent ausgeführt wird.

   Ubuntu Server 16.04

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   Ubuntu Server14.04

   ```
   sudo status amazon-ssm-agent
   ```

   In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird.

   In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

6. Führen Sie einen der folgenden Befehle aus, um den Service zu starten, wenn der vorherige Befehl die Meldung `amazon-ssm-agent is stopped`, `inactive` oder `disabled` zurückgibt.

   Ubuntu Server 16.04:

   ```
   sudo systemctl enable amazon-ssm-agent
   ```

   Ubuntu Server 14.04:

```
sudo start amazon-ssm-agent
```

Erstellen Sie benutzerdefinierte Installationsbefehle für SSM Agent auf Ubuntu Server 16.04 und 14.04 32-Bit (deb) in Ihrer Region

Wenn Sie SSM Agent bei mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf Ubuntu Server 16.04 und 14.04 32-Bit (deb) weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie im folgenden Befehl *`region`* mit Ihren eigenen Informationen. Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_386/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Sehen Sie sich das folgende -Beispiel an.

```
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
```

Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04

> **Important**
>
> Stellen Sie vor der Installation von SSM Agent auf einer 64-Bit-Version von Ubuntu Server sicher, dass Sie die richtigen Installationstools verwenden. Beginnend mit Amazon Machine Images (AMIs), die mit 20180627 identifiziert werden, ist SSM Agent unter Verwendung von Snap-Paketen auf Version 16.04 vorinstalliert. Auf Instances, die aus früheren AMIs erstellt wurden, muss SSM Agent mit Deb-Installationsprogramm-Paketen installiert werden. Weitere Informationen finden Sie unter Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04
> Beachten Sie, dass bei Instances mit mehr als einer Installation des SSM Agent (z. B. eine mit einem Snap und eine mit einem deb-Installationsprogramm installierte Instance) Ihre Agenten-Operationen nicht ordnungsgemäß ausgeführt werden.

Sie können das AMI-ID-Quell-Erstellungsdatum für eine Instance mit einem der folgenden Verfahren verifizieren. Diese Verfahren gelten nur für AWS-verwaltete AMIs.

Überprüfen eines AMI-ID-Quell-Erstellungsdatums (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im linken Navigationsbereich die Option Instances aus.
3. Wählen Sie eine Instance aus.
4. Suchen Sie auf der Registerkarte Details nach einer YYYYMMDD-Kennung in dem Wert im Feld AMI-Name. Beispiel: `ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20180627`.

Überprüfen eines AMI-ID-Quell-Erstellungsdatums (AWS CLI)

- Führen Sie den folgenden Befehl aus.

```
aws ec2 describe-images --image-ids ami-id
```

*ami-id*repräsentiert die ID eines von AWS bereitgestellten AMI, wie beispielsweise `ami-07c8bc5c1ce9598c3`.

Bei Erfolg gibt der Befehl Informationen wie die folgenden zurück, in denen Sie die `CreationDate`- und `Name`-Felder auf Informationen prüfen können.

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

Konfiguration SSM Agent für die Verwendung eines Proxys auf Linux-Knoten

Sie können AWS Systems Manager Agent (SSM Agent) für die Kommunikation über einen HTTP-Proxy konfigurieren, indem Sie eine Override-Konfigurationsdatei erstellen und der Datei `http_proxyhttps_proxy`,, und `no_proxy` Einstellungen hinzufügen. Eine Override-Datei behält auch die Proxy-Einstellungen bei, wenn Sie neuere oder ältere Versionen von SSM Agent installieren. Dieser Abschnitt enthält Vorgänge zum Erstellen einer Override-Datei für die Umgebungen upstart und systemd. Wenn Sie dies verwenden möchtenSession Manager, beachten Sie, dass HTTPS-Proxyserver nicht unterstützt werden.

Themen

- Konfigurieren des SSM Agent zur Nutzung eines Proxys (Upstart)
- Konfigurieren des SSM Agent zur Nutzung eines Proxys (systemd)

Konfigurieren des SSM Agent zur Nutzung eines Proxys (Upstart)

Gehen Sie folgendermaßen vor, um eine Override-Konfigurationsdatei für eine `upstart`-Umgebung zu erstellen.

Konfigurieren des SSM Agent zur Nutzung eines Proxys (Upstart)

1. Stellen Sie eine Verbindung mit der verwalteten Instance her, auf der Sie den SSM Agent installiert haben.
2. Öffnen Sie einen einfachen Editor wie VIM und geben Sie je nachdem, ob Sie einen HTTP-Proxy-Server oder HTTPS-Proxy-Server verwenden, eine der folgenden Konfigurationen an:

   Für einen HTTP-Proxy-Server:

   ```
   env http_proxy=http://hostname:port
   env https_proxy=http://hostname:port
   env no_proxy=IP address for instance metadata services (IMDS)
   ```

   Bei einem HTTPS-Proxy-Server:

   ```
   env http_proxy=http://hostname:port
   env https_proxy=https://hostname:port
   env no_proxy=IP address for instance metadata services (IMDS)
   ```

   > **Important**
   >
   > Fügen Sie die `no_proxy` Einstellung zur Datei hinzu und geben Sie die IP-Adresse an. Die IP-Adresse für `no_proxy` ist der IMDS-Endpunkt (Instance Metadata Services) für Systems Manager. Wenn Sie dies nicht angeben`no_proxy`, übernehmen Aufrufe von Systems Manager die Identität des Proxydienstes (wenn IMDSv1-Fallback aktiviert ist) oder Aufrufe von Systems Manager schlagen fehl (wenn IMDSv2 erzwungen wird).
   >
   > - `no_proxy=169.254.169.254`Geben Sie für IPv4 an.
   > - Geben Sie für IPv6 an. `no_proxy=[fd00:ec2::254]` Die IPv6-Adresse des Instance-Metadatendienstes ist mit IMDSv2-Befehlen kompatibel. Auf die IPv6-Adresse kann nur auf Instances zugegriffen werden, die auf dem AWS Nitro-System basieren. Weitere Informationen finden Sie unter So funktioniert Instance Metadata Service Version 2 im Amazon EC2 EC2-Benutzerhandbuch.
3. Speichern Sie die Datei unter dem Namen `amazon-ssm-agent.override` am folgenden Speicherort: `/etc/init/`
4. Mit den folgenden Befehlen wird der SSM Agent beendet und neu gestartet.

```
sudo service stop amazon-ssm-agent
sudo service start amazon-ssm-agent
```

> **Note**
>
> Weitere Informationen zum Arbeiten mit `.override`-Dateien in Upstart-Umgebungen finden Sie unter init: Upstart-init-Daemon-Auftragskonfiguration.

Konfigurieren des SSM Agent zur Nutzung eines Proxys (systemd)

Gehen Sie wie folgt vor, um SSM Agent für die Verwendung eines Proxy in einer `systemd`-Umgebung zu konfigurieren.

> **Note**
>
> Einige der Schritte in diesem Verfahren enthalten explizite Anweisungen für Ubuntu Server-Instances, wenn SSM Agent unter Verwendung von Snap installiert wurden.

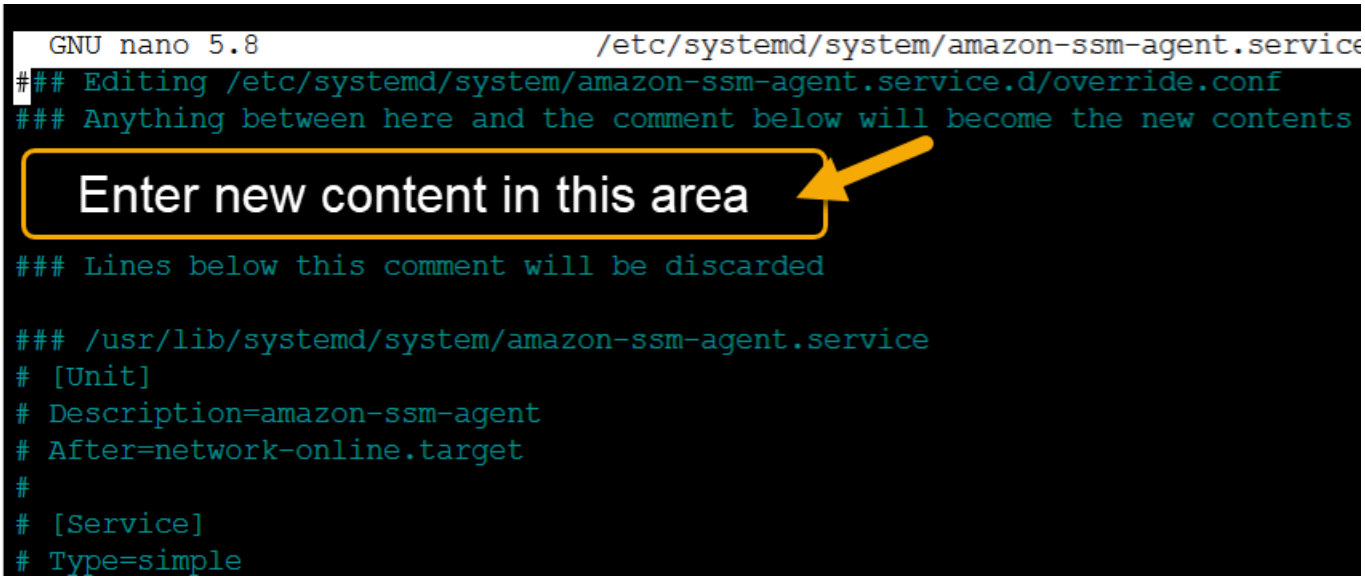
1. Stellen Sie eine Verbindung mit der Instance her, auf der Sie den SSM Agent installiert haben.
2. Führen Sie abhängig von der Art des Betriebssystems einen der folgenden Befehle aus.
   - Auf Ubuntu Server-Instances, wenn SSM Agent unter Verwendung von Snap installiert wurde:

     ```
     sudo systemctl edit snap.amazon-ssm-agent.amazon-ssm-agent
     ```

     Auf anderen Betriebssystemen:

     ```
     sudo systemctl edit amazon-ssm-agent
     ```
3. Öffnen Sie einen einfachen Editor wie VIM und geben Sie je nachdem, ob Sie einen HTTP-Proxy-Server oder HTTPS-Proxy-Server verwenden, eine der folgenden Konfigurationen an:

   Stellen Sie sicher, dass Sie die Informationen über dem Kommentar mit der Aufschrift "`### Lines below this comment will be discarded`„eingeben, wie in der folgenden Abbildung dargestellt.



Für einen HTTP-Proxy-Server:

```
[Service]
Environment="http_proxy=http://hostname:port"
Environment="https_proxy=http://hostname:port"
Environment="no_proxy=IP address for instance metadata services (IMDS)"
```

Bei einem HTTPS-Proxy-Server:

```
[Service]
Environment="http_proxy=http://hostname:port"
Environment="https_proxy=https://hostname:port"
Environment="no_proxy=IP address for instance metadata services (IMDS)"
```

Important

Fügen Sie die `no_proxy` Einstellung zur Datei hinzu und geben Sie die IP-Adresse an. Die IP-Adresse für `no_proxy` ist der IMDS-Endpunkt (Instance Metadata Services) für Systems Manager. Wenn Sie dies nicht angeben`no_proxy`, übernehmen Aufrufe von Systems Manager die Identität des Proxydienstes (wenn IMDSv1-Fallback aktiviert ist) oder Aufrufe von Systems Manager schlagen fehl (wenn IMDSv2 erzwungen wird).

- `no_proxy=169.254.169.254`Geben Sie für IPv4 an.
- Geben Sie für IPv6 an. `no_proxy=[fd00:ec2::254]` Die IPv6-Adresse des Instance-Metadatendienstes ist mit IMDSv2-Befehlen kompatibel. Auf die IPv6-

Adresse kann nur auf Instances zugegriffen werden, die auf dem AWS Nitro-System basieren. Weitere Informationen finden Sie unter So funktioniert Instance Metadata Service Version 2 im Amazon EC2 EC2-Benutzerhandbuch.

4. Speichern Sie Ihre Änderungen. Abhängig vom Typ des Betriebssystems erstellt das System automatisch eine der folgenden Dateien.

   - Auf Ubuntu Server-Instances, wenn SSM Agent unter Verwendung von Snap installiert wurde:

     `/etc/systemd/system/snap.amazon-ssm-agent.amazon-ssm-agent.service.d/override.conf`

   - Auf Instances von Amazon Linux 2 und Amazon Linux 2023:

     `/etc/systemd/system/amazon-ssm-agent.service.d/override.conf`

   - Auf anderen Betriebssystemen

     `/etc/systemd/system/amazon-ssm-agent.service.d/amazon-ssm-agent.override`

5. Starten Sie SSM Agent abhängig von der Art des Betriebssystems über einen der folgenden Befehle neu.

   - Auf Ubuntu Server-Instances, die unter Verwendung von Snap installiert wurden:

     ```
     sudo systemctl daemon-reload && sudo systemctl restart snap.amazon-ssm-agent.amazon-ssm-agent
     ```

   - Auf anderen Betriebssystemen:

     ```
     sudo systemctl daemon-reload && sudo systemctl restart amazon-ssm-agent
     ```

> **Note**
>
> Weitere Informationen zum Arbeiten mit `.override`-Dateien in systemd-Umgebungen finden Sie unter Modifying Existing Unit Files im Systemadministrator-Handbuch für Red Hat Enterprise Linux 7.

Arbeiten mit SSM Agent auf EC2-Instances für macOS

AWS Systems Manager (SSM Agent) verarbeitet Systems Manager Manager-Anfragen und konfiguriert Ihren Computer wie in der Anfrage angegeben. Sie können den SSM Agent für macOS auf folgende Weise installieren, konfigurieren oder deinstallieren.

> **Note**
>
> SSM Agent ist standardmäßig auf Amazon Machine Images (AMIs) für macOS vorinstalliert. Sie müssen SSM Agent nicht auf einer Amazon Elastic Compute Cloud (Amazon EC2)-Instance für macOS installieren, es sei denn, Sie haben es deinstalliert.

Der Quellcode für SSM Agent ist auf verfügbar, GitHubsodass Sie den Agenten an Ihre Bedürfnisse anpassen können. Wir möchten Sie bitten, uns eventuelle Änderungswünsche mitzuteilen. AWS bietet jedoch keine Unterstützung für die Ausführung modifizierter Kopien dieser Software.

> **Note**
>
> Details zu den unterschiedlichen SSM Agent-Versionen finden Sie in den Versionshinweisen.

Bevor Sie SSM Agent manuell auf einem macOS-Betriebssystem installieren, lesen Sie die folgenden Informationen.

- SSM Agent ist standardmäßig auf den folgenden EC2-Instances und Amazon Machine Images installiert:
  - macOS 10.14.x (Mojave)
  - macOS 10.15.x (Catalina)
  - macOS11.x () BigSur
  - macOS 12.x (Monterey)
  - macOS13,x (Ventura)
  - macOS14,x (Sonoma)

  SSM Agent muss nicht manuell auf macOSEC2-Instances installiert werden, es sei denn, es wurde deinstalliert.

- EC2-Instances für macOS werden nicht in allen unterstützt. AWS-Regionen Eine Liste der Regionen, in denen x86-basierte und M1 EC2-Instances für macOS unterstützt werden, finden Sie unter macOS-Workloads in den Häufig gestellten Fragen zu Amazon EC2.
- Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisen aufGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.

Themen

- Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für macOS

Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für macOS

Stellen Sie eine Verbindung mit Ihrer macOS-Instance her und folgen Sie den Schritten, um den AWS Systems Manager -Agent (SSM Agent) zu installieren. Führen Sie diese Schritte mit Systems Manager auf jeder Instance aus, auf der Befehle ausgeführt werden. Die in diesem Verfahren bereitgestellten Befehle können auch als Skripts über Benutzerdaten an Amazon-EC2-Instances übergeben werden.

Installieren Sie den SSM Agent auf macOS wie folgt

1. Laden Sie die Agent-Installationsdatei für x86_64-Instances mit dem folgenden Befehl herunter.

   Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

   ```
   sudo wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/darwin_amd64/
   amazon-ssm-agent.pkg
   ```

   Verwenden Sie für Apple silicon Instanzen den folgenden Befehl.

```
sudo wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/darwin_arm64/
amazon-ssm-agent.pkg
```

Ein Beispiel.

```
sudo wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
darwin_amd64/amazon-ssm-agent.pkg
```

2. Verwenden Sie den folgenden Befehl, um das SSM Agent-Installationsprogramm auszuführen.

   x86_64:

   ```
   sudo installer -pkg amazon-ssm-agent.pkg -target /
   ```

3. Prüfen Sie den Status des Agents.

   Um zu überprüfen, ob der SSM Agent ausgeführt wird, prüfen Sie das Agent-Protokoll unter `/var/log/amazon/ssm/amazon-ssm-agent.log`.

4. Führen Sie den folgenden Befehl aus, um den Dienst zu starten, falls das Agentenprotokoll anzeigt, dass "gestoppt amazon-ssm-agent ist“.

   ```
   sudo launchctl load -w /Library/LaunchDaemons/com.amazon.aws.ssm.plist && sudo
    launchctl start com.amazon.aws.ssm
   ```

> **Important**
>
> Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisenGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.

Deinstallieren des SSM Agent von macOS-Instances

macOS unterstützt die Deinstallation von PKG-Dateien nicht nativ. Um AWS Systems Manager Agent (SSM Agent) von einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance für zu deinstallierenmacOS, können Sie das AWS verwaltete Skript vom folgenden Speicherort aus verwenden.

https://github.com/aws/amazon-ssm-agent/blob/mainline/Tools/src/update/darwin/uninstall.sh

Arbeiten mit SSM Agent auf EC2-Instances für Windows Server

AWS Systems Manager Agent (SSM Agent) ist standardmäßig auf den Amazon Machine Images (AMIs) vorinstalliert, die von AWS bereitgestellt werden. Windows Server Support wird für die folgenden Betriebssystemversionen bereitgestellt.

- Windows Server 2008- bis 2012 R2-AMIs, die im November 2016 oder später veröffentlicht wurden
- Windows Server 2016, 2019 und 2022

Support-Hinweise für frühere Versionen

Windows Server-AMIs, die vor November 2016 veröffentlicht wurden, nutzen den EC2Config-Service, um Anforderungen zu verarbeiten und Instances zu konfigurieren.

Wenn Sie keinen besonderen Grund für die Verwendung des EC2Config-Dienstes oder einer früheren Version von SSM Agent für die Verarbeitung von Systems Manager-Anfragen haben, empfehlen wir Ihnen, die neueste Version von SSM Agent herunterzuladen und auf jeder Ihrer Instances der Amazon Elastic Compute Cloud (Amazon EC2) oder Nicht-EC2-Maschinen zu installieren, die für Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung konfiguriert sind.

Ab 14. Januar 2020 wird Windows Server 2008 für Feature- oder Sicherheitsupdates von Microsoft nicht mehr unterstützt. Legacy Amazon Machine Images (AMIs) für Windows Server 2008 und 2008 R2 enthalten immer noch die Version 2 vom vorinstallierten SSM Agent, aber Systems Manager unterstützt offiziell nicht mehr die 2008-Versionen und aktualisiert den Agenten für diese Versionen von Windows Server. Darüber hinaus ist SSM Agent Version 3 möglicherweise nicht mit allen Operationen auf Windows Server 2008 und 2008 R2 kompatibel. Die endgültige offiziell unterstützte Version von SSM Agent für Windows Server 2008 Versionen ist 2.3.1644.0.

Aktualisieren von SSM Agent

Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisenGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.

Details zu den unterschiedlichen SSM Agent-Versionen finden Sie in den Versionshinweisen.

Themen

- Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Windows Server
- Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances

Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Windows Server

AWS Systems Manager Agent (SSM Agent) ist standardmäßig auf dem folgenden Amazon Machine Images (AMIs) von Amazon Windows Server bereitgestellten Gerät vorinstalliert:

- Windows Server 2008- bis 2012 R2-AMIs, die im November 2016 oder später veröffentlicht wurden
- Windows Server 2016, 2019 und 2022

Installieren Sie SSM Agent auf EC2-Instances für Windows Server

Bei Bedarf können Sie die aktuelle Version des SSM Agent manuell auf Ihre Amazon Elastic Compute Cloud (Amazon EC2)-Instance für Windows Server herunterladen und installieren, indem Sie die folgenden Schritte ausführen. Die in diesem Verfahren bereitgestellten Befehle können auch als Skripts über Benutzerdaten an Amazon-EC2-Instances übergeben werden.

SSM Agenterfordert Windows PowerShell 3.0 oder höher, um bestimmte AWS Systems Manager Dokumente (SSM-Dokumente) auf Windows Server Instanzen auszuführen (z. B. das ältere `AWS-ApplyPatchBaseline` Dokument). Vergewissern Sie sich, dass Ihre Windows Server-Instances auf Windows Management Framework 3.0 oder höher ausgeführt werden. Dieses Framework umfasst Windows PowerShell. Weitere Informationen finden Sie unter Windows Management Framework 3.0

> **Note**
>
> Dieses Verfahren gilt für die Installation oder Neuinstallation von SSM Agent auf einer EC2-Instance für Windows Server. Wenn Sie den Agenten auf einem lokalen Server oder einer virtuellen Maschine (VM) installieren müssen, damit er mit Systems Manager verwendet werden kann, finden Sie unter So installieren Sie den SSM Agent auf hybriden Windows-Knoten.

So installieren Sie die neueste Version von SSM Agent auf EC2-Instances für Windows Server

1. Stellen Sie mithilfe von Remote Desktop oder Windows eine Connect zu Ihrer Instance her PowerShell. Weitere Informationen finden Sie unter Connect to your Instance im Amazon EC2 EC2-Benutzerhandbuch.
2. Laden Sie die neueste Version des SSM Agent auf Ihre Instance herunter. Sie können entweder PowerShell Befehle oder einen direkten Download-Link verwenden.

   > **Note**
   >
   > Mit den URLs in diesem Schritt können Sie SSM Agent von beliebigen URLs herunterladen AWS-Region. Wenn Sie den Agenten aus einer bestimmten Region herunterladen möchten, verwenden Sie stattdessen eine regionsspezifische URL: `https://amazon-ssm-region.s3.region.amazonaws.com/latest/windows_amd64/AmazonSSMAgentSetup.exe`
   > *region* steht für den Bezeichner für eine Region AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

   PowerShell

   Führen Sie die folgenden drei PowerShell Befehle der Reihe nach aus. Mit diesen Befehlen können Sie SSM Agent herunterladen, ohne die erweiterten Sicherheitseinstellungen von Internet Explorer (IE) anzupassen, und dann den Agent installieren und die Installationsdatei entfernen.

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"
```

```
rm -Force $env:USERPROFILE\Desktop\SSMAgent_latest.exe
```

Direkter Download

Laden Sie über den folgenden Link die neueste Version von SSM Agent auf Ihre Instance herunter. Wenn Sie möchten, aktualisieren Sie diese URL mit einer AWS-Region-spezifischen URL.

https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/AmazonSSM .exe AgentSetup

Führen Sie die heruntergeladene `AmazonSSMAgentSetup.exe`-Datei aus, um SSM Agent zu installieren.

3. Starten oder starten Sie neu, SSM Agent indem Sie den folgenden Befehl einsenden PowerShell:

```
Restart-Service AmazonSSMAgent
```

Deinstallation SSM Agent von EC2-Instances für Windows Server

Um das SSM Agent von einer Windows Server Instance zu deinstallieren, öffnen Sie die Systemsteuerung unter Programme. Wählen Sie die Option Uninstall a program (Programm deinstallieren). Öffnen Sie das Kontextmenü (Rechtsklick) für Amazon SSM Agent und wählen Sie Uninstall (Deinstallieren) aus.

Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances

Die Informationen in diesem Thema gelten für Windows Server-Instances, die im oder nach dem November 2016 erstellt wurden, die nicht die Nano-Installationsoption verwenden. Wenn Sie die Verwendung beabsichtigenSession Manager, beachten Sie, dass HTTPS-Proxyserver nicht unterstützt werden.

> **Note**
>
> Ab 14. Januar 2020 wird Windows Server 2008 für Feature- oder Sicherheitsupdates von Microsoft nicht mehr unterstützt. Legacy Amazon Machine Images (AMIs) für Windows Server 2008 und 2008 R2 enthalten immer noch die Version 2 vom vorinstallierten SSM Agent, aber Systems Manager unterstützt offiziell nicht mehr die 2008-Versionen und aktualisiert den Agenten für diese Versionen von Windows Server. Darüber hinaus ist SSM Agent Version 3 möglicherweise nicht mit allen Operationen auf Windows Server 2008 und 2008 R2 kompatibel. Die endgültige offiziell unterstützte Version von SSM Agent für Windows Server 2008 Versionen ist 2.3.1644.0.

Bevor Sie beginnen

Beachten Sie SSM Agent die folgenden wichtigen Informationen, bevor Sie die Konfiguration für die Verwendung eines Proxys vornehmen.

Im folgenden Verfahren führen Sie einen Befehl aus, um die Verwendung eines Proxys SSM Agent zu konfigurieren. Der Befehl enthält eine `no_proxy` Einstellung mit einer IP-Adresse. Die IP-Adresse ist der IMDS-Endpunkt (Instance Metadata Services) für Systems Manager. Wenn Sie dies nicht angeben`no_proxy`, übernehmen Aufrufe von Systems Manager die Identität des Proxydienstes

(wenn IMDSv1-Fallback aktiviert ist) oder Aufrufe von Systems Manager schlagen fehl (wenn IMDSv2 erzwungen wird).

- `no_proxy=169.254.169.254`Geben Sie für IPv4 an.
- Geben Sie für IPv6 an. `no_proxy=[fd00:ec2::254]` Die IPv6-Adresse des Instance-Metadatendienstes ist mit IMDSv2-Befehlen kompatibel. Auf die IPv6-Adresse kann nur auf Instances zugegriffen werden, die auf dem AWS Nitro-System basieren. Weitere Informationen finden Sie unter So funktioniert Instance Metadata Service Version 2 im Amazon EC2 EC2-Benutzerhandbuch.

So konfigurieren Sie den SSM Agent zur Nutzung eines Proxys

1. Stellen Sie über Remote Desktop oder Windows PowerShell eine Verbindung zu der Instance her, die Sie für die Verwendung eines Proxys konfigurieren möchten.
2. Führen Sie den folgenden Befehlsblock in aus PowerShell. Ersetzen Sie *`Hostname`* und *`Port`* durch die Informationen zu Ihrem 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
```

Nachdem Sie den vorherigen Befehl ausgeführt haben, können Sie die SSM Agent-Protokolle überprüfen, um zu bestätigen, dass die Proxy-Einstellungen angewendet wurden. Einträge in den Protokollen ähneln den folgenden. Weitere Informationen über SSM Agent-Protokolle finden Sie unter Anzeigen von SSM Agent-Protokollen.

```
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
```

So setzen die Proxy-Konfiguration des SSM Agent zurück

1. Stellen Sie mithilfe von Remote Desktop oder Windows PowerShell eine Verbindung zu der zu konfigurierenden Instanz her.
2. Wenn Sie über Remote Desktop eine Verbindung hergestellt haben, starten Sie PowerShell als Administrator.
3. Führen Sie den folgenden Befehlsblock in aus PowerShell.

   ```
   Remove-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\AmazonSSMAgent -
   Name Environment
   Restart-Service AmazonSSMAgent
   ```

Priorität der SSM Agent-Proxy-Einstellung

Beim Konfigurieren von Proxyeinstellungen für die SSM Agent unter Windows Server-Instances muss beachtet werden, dass diese Einstellungen ausgewertet und auf die Agentenkonfiguration angewendet werden, wenn der SSM Agent gestartet wird. Wie Sie Ihre Proxy-Einstellungen für eine Windows Server-Instance konfigurieren, kann bestimmen, ob andere Einstellungen Ihre beabsichtigten Einstellungen möglicherweise ersetzen.

> **Important**
>
> SSM Agent kommuniziert mit dem HTTPS-Protokoll. Aus diesem Grund müssen Sie die `HTTPS proxy`-Parameter mithilfe einer der folgenden Einstellungsoptionen konfigurieren.

SSM Agent-Proxy-Einstellungen werden in der folgenden Reihenfolge ausgewertet.

1. AmazonSSMAgent-Registrierungseinstellungen (`HKLM:\SYSTEM\CurrentControlSet\Services\AmazonSSMAgent`)
2. Systemumgebungsvariablen (`http_proxy`, `https_proxy`, `no_proxy`)
3. LocalSystem Umgebungsvariablen für Benutzerkonten`http_proxy`,`https_proxy`,`no_proxy`)
4. Internet-Explorer-Einstellungen (`HTTP`, `secure`, `exceptions`)
5. WinHTTP-Proxy-Einstellungen (`http=`, `https=`, `bypass-list=`)

SSM Agent-Proxy-Einstellungen und Systems Manager-Services

Wenn Sie den für SSM Agent die Verwendung eines Proxys konfiguriert haben und AWS Systems Manager Funktionen wie Run Command und verwenden Patch Manager PowerShell , die den Windows Update-Client während der Ausführung auf Windows Server Instanzen verwenden, konfigurieren Sie zusätzliche Proxyeinstellungen. Andernfalls schlägt der Vorgang möglicherweise fehl, da die vom PowerShell und vom Windows Update-Client verwendeten Proxyeinstellungen nicht von der SSM Agent Proxykonfiguration übernommen werden.

Konfigurieren Sie für Run Command `WinINet`-Proxy-Einstellungen auf Ihren Windows Server-Instances. Die `[System.Net.WebRequest]`-Befehle werden pro Sitzung bereitgestellt.
Um diese Konfigurationen auf nachfolgende Netzwerkbefehle anzuwenden, die in ausgeführt werdenRun Command, müssen diese Befehle vor anderen PowerShell Befehlen in derselben `aws:runPowershellScript` Plugin-Eingabe stehen.

Die folgenden PowerShell Befehle geben die aktuellen `WinINet` Proxyeinstellungen zurück und wenden Ihre Proxyeinstellungen auf an`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
```

Sie müssen für Patch Manager systemweite Proxy-Einstellungen konfigurieren, damit der Windows Update-Client nach Updates suchen und herunterladen kann. Es wird empfohlen, dass Sie Run Command verwenden, um die folgenden Befehle auszuführen, da sie auf dem SYSTEM-Konto ausgeführt werden und die Einstellungen systemweit gelten. Mit den folgenden `netsh`-Befehlen

werden die aktuellen Proxy-Einstellungen zurückgegeben und die Proxy-Einstellungen werden auf das lokale System angewendet.

```
netsh winhttp show proxy

netsh winhttp set proxy proxy-server="hostname:port" bypass-list="169.254.169.254"
```

Weitere Informationen zur Verwendung von Run Command finden Sie unter AWS Systems Manager Run Command.

Prüfen des SSM Agent-Status und Starten des Agenten

In diesem Thema werden die Befehle aufgeführt, mit denen überprüft werden kann, ob AWS Systems Manager Agent (SSM Agent) auf jedem unterstützten Betriebssystem ausgeführt wird. Es enthält auch die Befehle, mit denen der Agent gestartet wird, wenn er nicht ausgeführt wird.

Betriebssystem	Befehl zum Überprüfen des SSM Agent-Status	Befehl zum Starten von SSM Agent
Amazon Linux 1	`sudo status amazon-ssm-agent`	`sudo start amazon-ssm-agent`
Amazon Linux 2 und 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 und 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 und 10	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent`

Betriebssystem	Befehl zum Überprüfen des SSM Agent-Status	Befehl zum Starten von SSM Agent
		`sudo systemctl start amazon-ssm-agent`
macOS	Überprüfen Sie die Agent-Protokolldatei unter `/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`
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 und 9.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 (alle) und 16.04 (32-Bit)	`sudo status amazon-ssm-agent`	`sudo start amazon-ssm-agent`

Betriebssystem	Befehl zum Überprüfen des SSM Agent-Status	Befehl zum Starten von SSM Agent
Ubuntu Server 16.04-64-Bit-Instances (deb-Installationsprogrammpaket)	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
Ubuntu Server 16.04, 18.04 und 20.04 LTS, 20.10 STR 64-bit und 22.04 LTS (Snap-Paketinstallation)	`sudo systemctl status snap.amazon-ssm-agent.amazon-ssm-agent.service`	`sudo snap start amazon-ssm-agent`
Windows Server	Führen Sie ein: PowerShell `Get-Service AmazonSSMAgent`	Im PowerShell Administratormodus ausführen: `Start-Service AmazonSSMAgent`

Weitere Informationen

- Arbeiten mit SSM Agent auf EC2-Instances für Linux
- Arbeiten mit SSM Agent auf EC2-Instances für Windows Server
- Überprüfen der SSM Agent-Versionsnummer

Überprüfen der SSM Agent-Versionsnummer

Bestimmte AWS Systems Manager Funktionen erfordern die Installation einer Mindestversion von Systems Manager Agent (SSM Agent) auf Ihren verwalteten Knoten. Sie können die aktuell installierte SSM Agent-Version auf Ihren verwalteten Knoten über die Systems-Manager-Konsole abrufen oder sich bei Ihren verwalteten Knoten anmelden.

In den folgenden Verfahren wird beschrieben, wie Sie die aktuell installierte SSM Agent-Version auf Ihren verwalteten Knoten abrufen.

So ermitteln Sie die Versionsnummer von SSM Agent auf einem verwalteten Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Notieren Sie sich in SSM Agent-Version die Nummer der Agent-Version.

So rufen Sie die aktuell installierte SSM Agent-Version innerhalb des Betriebssystems ab

Wählen Sie aus den folgenden Registerkarten aus, um die aktuell installierte SSM Agent-Version innerhalb des Betriebssystems abzurufen.

Amazon Linux 1, Amazon Linux 2, and Amazon Linux 2023

> **Note**
>
> Dieser Befehl variiert je nach Paketmanager für Ihr Betriebssystem.

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl aus.

```
yum info amazon-ssm-agent
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
Name        : amazon-ssm-agent
Arch        : x86_64
Version     : 3.0.655.0
```

CentOS

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl für CentOS 6 und 7 aus.

```
yum info amazon-ssm-agent
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
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. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl aus.

```
apt list amazon-ssm-agent
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
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. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl aus.

```
pkgutil --pkg-info com.amazon.aws.ssm
```

RHEL

1. Melden Sie sich bei Ihrem verwalteten Knoten an.

2. Führen Sie den folgenden Befehl für RHEL 6, 7, 8 und 9 aus.

```
yum info amazon-ssm-agent
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
Name        : amazon-ssm-agent
Arch        : x86_64
Version     : 3.0.655.0
```

Führen Sie den folgenden Befehl für das DNF-Paketdienstprogramm aus.

```
dnf info amazon-ssm-agent
```

SLES

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl für SLES 12 und 15 aus.

```
zypper info amazon-ssm-agent
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
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**
>
> Ob Ihre Ubuntu Server 16.04-Instance Deb- oder Snap-Pakete verwendet, finden Sie unter Manuelle Installation von SSM Agent auf Ubuntu Server-Instances heraus.

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl auf Ubuntu Server 16.04 und 14.04 64-Bit (mit dem deb-Installationsprogrammpaket) aus.

   ```
   apt list amazon-ssm-agent
   ```

   Dieser Befehl gibt etwa die folgende Ausgabe zurück.

   ```
   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
   ```

   Führen Sie den folgenden Befehl für Ubuntu Server 22.04 LTS, 20.10 STR und 20.04, 18.04 und 16.04 LTS-64-Bit-Instances (mit Snap-Paket) aus.

   ```
   sudo snap list amazon-ssm-agent
   ```

   Dieser Befehl gibt etwa die folgende Ausgabe zurück.

   ```
   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. Melden Sie sich bei Ihrem verwalteten Knoten an.

2. Führen Sie den folgenden PowerShell Befehl aus.

```
& "C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe" -version
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
SSM Agent version: 3.1.804.0
```

Wir empfehlen, die neueste SSM Agent-Version zu verwenden, damit Sie von neuen oder aktualisierten Funktionen profitieren können. Um sicherzustellen, dass auf Ihren verwalteten Instanzen immer die neueste up-to-date Version von ausgeführt wirdSSM Agent, können Sie den Aktualisierungsprozess von automatisierenSSM Agent. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent.

Anzeigen von SSM Agent-Protokollen

AWS Systems Manager Agent (SSM Agent) schreibt Informationen über Ausführungen, Befehle, geplante Aktionen, Fehler und den Integritätsstatus in Protokolldateien auf jedem verwalteten Knoten. Sie können Protokolldateien anzeigen, indem Sie manuell eine Verbindung zu einem verwalteten Knoten herstellen, oder Sie können Protokolle automatisch an Amazon CloudWatch Logs senden. Weitere Informationen zum Senden von Protokollen an CloudWatch Logs finden Sie unterÜberwachung AWS Systems Manager.

Sie können SSM Agent-Protokolle auf verwalteten Knoten in den folgenden Speicherorten anzeigen.

Linux and macOS

```
/var/log/amazon/ssm/
```

Windows

```
%PROGRAMDATA%\Amazon\SSM\Logs\
```

Für Linux-verwaltete Knoten werden die Dateien SSM Agent, `stderr` und `stdout` in das folgende Verzeichnis geschrieben: `/var/lib/amazon/ssm/`.

Für Windows-verwaltete Knoten werden die Dateien SSM Agent, `stderr` und `stdout` in das folgende Verzeichnis geschrieben: `%PROGRAMDATA%\Amazon\SSM\InstanceData\`.

Informationen zur Aktivierung der SSM Agent-Debug-Protokollierung finden Sie unter Zulassen von SSM Agent-Debug-Protokollierung.

Weitere Informationen zur `cihub/seelog` Konfiguration finden Sie im Seelog-Wiki unterGitHub. Beispiele für `cihub/seelog` Konfigurationen finden Sie im cihub/seelog-Beispiel-Repository unter. GitHub

Zulassen von SSM Agent-Debug-Protokollierung

Mit den folgenden Verfahren können Sie die SSM Agent-Debug-Protokollierung auf Ihren verwalteten Knoten erlauben.

Linux and macOS

So erlauben Sie die SSM Agent-Debug-Protokollierung auf Linux- und macOS-verwalteten Knoten

1. Verwenden Sie entweder die Fähigkeit von Session Manager AWS Systems Manager, um eine Verbindung zu dem verwalteten Knoten herzustellen, für den Sie die Debug-Protokollierung zulassen möchten, oder melden Sie sich am verwalteten Knoten an. Weitere Informationen finden Sie unter Arbeiten mit Session Manager.
2. Suchen Sie die Datei seelog.xml.template.

   Linux:

   Bei den meisten von Linux verwalteten Knotentypen befindet sich die Datei im Verzeichnis `/etc/amazon/ssm/seelog.xml.template`.

   Auf Ubuntu Server 20.10 STR und 20.04, 18.04 und 16.04 LTS befindet sich die Datei im Verzeichnis `/snap/amazon-ssm-agent/current/seelog.xml.template`. Kopieren Sie diese Datei aus dem `/snap/amazon-ssm-agent/current/`-Verzeichnis in das `/etc/amazon/ssm/`-Verzeichnis, bevor Sie Änderungen vornehmen.

   macOS:

   Bei macOS-Instance-Typen befindet sich die Datei im Verzeichnis `/opt/aws/ssm/seelog.xml.template`.
3. Ändern Sie den Dateinamen von `seelog.xml.template` in `seelog.xml`.

> **Note**
>
> Auf Ubuntu Server 20.10 STR und 20.04, 18.04 und 16.04 LTS muss die Datei `seelog.xml` im Verzeichnis `/etc/amazon/ssm/` erstellt werden. Sie können dieses Verzeichnis und die Datei mit den folgenden Befehlen erstellen.
>
> ```
> sudo mkdir -p /etc/amazon/ssm
> ```
>
> ```
> sudo cp -p /snap/amazon-ssm-agent/current/seelog.xml.template /etc/
> amazon/ssm/seelog.xml
> ```

4. Bearbeiten Sie die Datei `seelog.xml`, um das Standardverhalten für die Protokollierung zu ändern. Ändern Sie den Wert für minlevel (Mindeststufe) von info (Info) in debug (Debuggen) wie im folgenden Beispiel gezeigt.

   ```
   <seelog type="adaptive" mininterval="2000000" maxinterval="100000000"
   critmsgcount="500" minlevel="debug">
   ```

5. (Optional) Starten Sie SSM Agent mit dem folgenden Befehl neu.

   Linux:

   ```
   sudo service amazon-ssm-agent restart
   ```

   macOS:

   ```
   sudo /opt/aws/ssm/bin/amazon-ssm-agent restart
   ```

Windows

So erlauben Sie die SSM Agent-Debug-Protokollierung auf Windows Server-verwalteten Knoten

1. Verwenden Sie entweder Session Manager, um eine Verbindung mit dem verwalteten Knoten herzustellen, für den Sie die Debug-Protokollierung aktivieren möchten, oder melden Sie sich bei dem verwalteten Knoten an. Weitere Informationen finden Sie unter Arbeiten mit Session Manager.

2. Erstellen Sie eine Kopie der Datei seelog.xml.template. Ändern Sie den Namen der Kopie auf seelog.xml. Die Datei befindet sich im folgenden Verzeichnis.

   ```
   %PROGRAMFILES%\Amazon\SSM\seelog.xml.template
   ```

3. Bearbeiten Sie die Datei `seelog.xml`, um das Standardverhalten für die Protokollierung zu ändern. Ändern Sie den Wert für minlevel (Mindeststufe) von info (Info) in debug (Debuggen) wie im folgenden Beispiel gezeigt.

   ```
   <seelog type="adaptive" mininterval="2000000" maxinterval="100000000"
   critmsgcount="500" minlevel="debug">
   ```

4. Suchen Sie den folgenden Eintrag.

   ```
   filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\{{EXECUTABLENAME}}.log"
   ```

   Ändern Sie diesen Eintrag, sodass der folgende Pfad verwendet wird.

   ```
   filename="C:\ProgramData\Amazon\SSM\Logs\amazon-ssm-agent.log"
   ```

5. Suchen Sie den folgenden Eintrag.

   ```
   filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\errors.log"
   ```

   Ändern Sie diesen Eintrag, sodass der folgende Pfad verwendet wird.

   ```
   filename="C:\ProgramData\Amazon\SSM\Logs\errors.log"
   ```

6. Starten Sie SSM Agent mit dem folgenden PowerShell Befehl im Administratormodus neu.

   ```
   Restart-Service AmazonSSMAgent
   ```

Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent

AWS Systems Manager Agent (SSM Agent) wird auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances und anderen Maschinentypen in Hybrid- und Multi-Cloud-Umgebungen mit Root-Rechten (Linux) oder SYSTEM-Berechtigungen (Windows Server) ausgeführt. Da dies die höchste Stufe der Systemzugriffsberechtigungen ist, verfügt jede vertrauenswürdige Entität, der die Berechtigung zum Senden von Befehlen an den SSM Agent erteilt wurde, über Root- oder SYSTEM-Berechtigungen. (In AWS wird eine vertrauenswürdige Entität, die Aktionen ausführen und auf Ressourcen zugreifen kann, als AWS Principal bezeichnet. Ein Principal kann ein Root-Benutzer des AWS-Kontos Benutzer oder eine Rolle sein.)

Dieses Zugriffslevel ist erforderlich, damit ein Prinzipal autorisierte Systems Manager-Befehle an den SSM Agent senden kann, es ermöglicht einem Prinzipal aber auch, bösartigen Code auszuführen, indem potenzielle Schwachstellen in SSM Agent ausgenutzt werden.

Insbesondere sollten die Berechtigungen zum Ausführen der Befehle SendCommand und StartSession sorgfältig beschränkt werden. Eine guter erster Schritt ist es, Berechtigungen für jeden Befehl nur für bestimmte Prinzipale in Ihrer Organisation zu gewähren. Allerdings empfehlen wir, Ihren Sicherheitsstatus weiter zu verbessern, indem Sie einschränken, auf welchen verwalteten Knoten ein Prinzipal diese Befehle ausführen kann. Dies kann in der IAM-Richtlinie erfolgen, die dem Prinzipal zugeordnet ist. In die IAM-Richtlinie können Sie eine Bedingung einfügen, mit der der Benutzer nur Befehle auf verwalteten Knoten ausführen kann, die mit spezifischen Tags oder Kombinationen von Tags markiert sind.

Nehmen wir an, Sie haben zwei Serverflotten, eine für Tests und eine für die Produktion. In der IAM-Richtlinie, die für nachrangige Ingenieure gilt, geben Sie an, dass sie Befehle nur auf Instances ausführen können, die mit `ssm:resourceTag/testServer` gekennzeichnet sind. Aber für eine kleinere Gruppe von leitenden Ingenieuren, die alle Instances zugreifen können sollten, gewähren Sie Zugriff auf Instances, die mit `ssm:resourceTag/testServer` und `ssm:resourceTag/productionServer` gekennzeichnet sind.

Mit diesem Ansatz wird nachrangigen Ingenieuren, die versuchen, einen Befehl auf einer Produktions-Instance auszuführen, der Zugriff verweigert, da ihre zugewiesenen IAM-Richtlinie keinen expliziten Zugriff auf Instances zulässt, die mit `ssm:resourceTag/productionServer` gekennzeichnet sind.

Weitere Informationen und Beispiele finden Sie in den folgenden Themen:

- Den Zugriff von Run Command anhand von Tags beschränken
- Beschränkung des Sitzungszugriffs auf Instance-Tags

Automatisieren von Updates für SSM Agent

AWS veröffentlicht eine neue Version von AWS Systems Manager Agent (SSM Agent), wenn wir Systems Manager Manager-Funktionen hinzufügen oder aktualisieren. Wenn Ihre verwalteten Knoten eine ältere Version des Agents verwenden, können Sie weder die neuen Funktionen nutzen noch von den aktualisierten Funktionen profitieren. Aus diesen Gründen empfehlen wir, die Aktualisierung von SSM Agent auf Ihren verwalteten Knoten mit einer der folgenden Methoden zu automatisieren.

Agent-Updates auf dem Bottlerocket-Betriebssystem

SSM Agent auf dem Bottlerocket-Betriebssystem kann nicht mit dem Befehlsdokument `AWS-UpdateSSMAgent` von Systems Manager aktualisiert werden. Updates werden im Bottlerocket-Control-Container verwaltet. Weitere Informationen finden Sie unter Bottlerocket Control Container und Bottlerocket Update Infrastructure on. GitHub

macOS-Versionsanforderung

Wenn auf einer Instance macOS-Version 11.0 (Big Sur) oder höher ausgeführt wird, muss die Instance über die SSM Agent-Version 3.1.941.0 oder höher verfügen, um das AWS-UpdateSSMAgent-Dokument auszuführen. Wenn auf der Instance eine Version von SSM Agent ausgeführt wird, die vor 3.1.941.0 veröffentlicht wurde, aktualisieren Sie Ihr SSM Agent, um die AWS-UpdateSSMAgent auszuführen, indem Sie die `brew update`- und `brew upgrade amazon-ssm-agent`-Befehle ausführen.

Methode	Details
Automatisierte Aktualisierung auf allen verwalteten Knoten mit einem Klick (Empfohlen)	Sie können alle verwalteten Knoten in Ihrem so konfigurieren, AWS-Konto dass automatisch nach neuen Versionen von gesucht und diese heruntergeladen werden. SSM Agent Wählen Sie dazu auf der Registerkarte Einstellungen in Fleet Manager die Option Automatische Aktualisierung von SSM Agent, wie nachfolgend in diesem Thema beschrieben.
Globale oder selektive Aktualisierung	Sie können eine Funktion von verwendenState Manager, um eine Zuordnung zu erstellen AWS Systems Manager, die automatisch SSM Agent auf Ihre verwalteten Knoten heruntergeladen und dort installiert wird. Wenn Sie die Unterbrechung Ihrer Workloads begrenzen möchten, können Sie ein Systems Manager-Wartungsfenster erstellen, um die Installation in festgelegten Zeiträumen durchzuführen. Bei beiden Methoden können Sie entweder eine globale Aktualisierungs-Konfiguration für alle Ihre verwalteten Knoten erstellen oder auswählen, welche Instances aktualisiert werden. Informati

Methode	Details
	onen zum Erstellen einer State Manager-Z uordnung finden Sie unter Anleitung: Automatische Aktualisierung von SSM Agent (CLI). Weitere Informationen zur Verwendung eines Wartungsfensters finden Sie unter Anleitung: Erstellen eines Wartungsfensters zum Aktualisieren von SSM Agent (AWS CLI) und Walkthrough: Erstellen eines Wartungsfensters zum automatischen Aktualisieren von SSM Agent (Konsole).
Globale oder selektive Aktualisierung für neue Umgebungen	Wenn Sie mit Systems Manager beginnen, empfehlen wir Ihnen, den Update Systems Manager (SSM) -Agenten alle zwei Wochen zu verwendenQuick Setup, eine Fähigkeit von AWS Systems Manager. Quick Setupermöglicht es Ihnen, entweder eine globale Update-Konfiguration für alle Ihre verwalteten Knoten zu erstellen oder selektiv auszuwählen, welche verwalteten Knoten aktualisiert werden. Weitere Informationen finden Sie unter Amazon-EC2-Host-Verwaltung.

Wenn Sie es vorziehen, Ihre verwalteten Knoten manuell zu aktualisierenSSM Agent, können Sie Benachrichtigungen abonnieren, die AWS veröffentlicht werden, wenn eine neue Version des Agenten veröffentlicht wird. Weitere Informationen finden Sie unter Abonnieren von SSM Agent-Benachrichtigungen. Nachdem Sie Benachrichtigungen abonniert haben, können Sie Run Command verwenden, um einen oder mehrere verwalteten Knoten manuell mit der neuesten Version zu aktualisieren. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

Automatische Aktualisierung von SSM Agent

Sie können Systems Manager so konfigurieren, dass er SSM Agent automatisch auf allen Linux-basierten und Windows-basierten verwalteten Knoten in Ihrem AWS-Konto aktualisiert. Wenn

Sie diese Option aktivieren, sucht Systems Manager automatisch alle zwei Wochen nach einer neuen Version des Agenten. Wenn es eine neue Version gibt, aktualisiert Systems Manager den Agent mit Hilfe des SSM-Dokuments `AWS-UpdateSSMAgent` automatisch auf die neueste freigegebene Version. Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass auf Ihren verwalteten Knoten immer die neueste up-to-date Version von ausgeführt wirdSSM Agent.

> **Note**
>
> Wenn Sie einen `yum`-Befehl zum Aktualisieren des SSM Agent auf einem verwalteten Knoten verwenden, sehen Sie, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, möglicherweise die folgende Meldung: „Warning: RPMDB altered outside of yum.“ (Warnung: RPMDB wurde außerhalb von yum geändert.) Diese Meldung wird erwartet und kann ignoriert werden.

So aktualisieren Sie SSM Agent automatisch

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Registerkarte Settings.
4. Wählen Sie im Bereich Automatische Aktualisierung des Agenten die Option Automatische Aktualisierung von SSM Agent.

Um zu ändern, auf welche Version von SSM Agent Ihre Flotte aktualisiert wird, wählen Sie unter Agent auto update (Automatische Agent-Aktualisierung) auf der Registerkarte Settings (Einstellungen) Edit (Bearbeiten). Geben Sie dann bei Version unter Parameter die Versionsnummer von SSM Agent ein, auf die Sie aktualisieren möchten. Ist hierfür nichts angegeben, wird der Agent auf die neueste Version aktualisiert.

Um die automatische Bereitstellung aktualisierter Versionen von SSM Agent auf alle verwalteten Knoten in Ihrem Konto zu beenden, wählen Sie unter Agent auto update (Automatische Agent-Aktualisierung) auf der Registerkarte Settings (Einstellungen) Delete (Löschen). Diese Aktion löscht die State Manager-Zuordnung gelöscht, durch die SSM Agent auf Ihren verwalteten Knoten automatisch aktualisiert wird.

Abonnieren von SSM Agent-Benachrichtigungen

Amazon Simple Notification Service (Amazon SNS) kann Sie benachrichtigen, wenn neue Versionen von AWS Systems Manager Agent (SSM Agent) veröffentlicht werden. Führen Sie die folgenden Schritte durch, um diese Benachrichtigungen zu abonnieren.

> **Tip**
>
> Sie können Benachrichtigungen auch abonnieren, indem Sie sich die Seite mit den SSM AgentVersionshinweisen unter GitHub ansehen.

So abonnieren Sie SSM Agent-Benachrichtigungen

1. Öffnen Sie die Amazon SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home.
2. Wählen Sie über die Regionsauswahl in der Navigationsleiste USA Ost (Nord-Virginia) aus, falls diese Option nicht bereits ausgewählt ist. Sie müssen diese Option auswählen AWS-Region , da die Amazon SNS SNS-BenachrichtigungenSSM Agent, für die Sie sich anmelden, nur aus dieser Region generiert werden.
3. Wählen Sie im Navigationsbereich Subscriptions aus.
4. Wählen Sie Create subscription.
5. Führen Sie unter Create subscription (Abonnement erstellen) die folgenden Schritte aus:
   a. Verwenden Sie für Topic ARN den folgenden Amazon-Ressourcennamen (ARN):

      ```
      arn:aws:sns:us-east-1:720620558202:SSM-Agent-Update
      ```

   b. Wählen Sie für Protokoll `Email` oder `SMS` aus.
   c. Geben Sie für Endpoint (Endpunkt) je nachdem, ob Sie im vorherigen Schritt `Email` oder `SMS` gewählt haben, eine E-Mail-Adresse oder eine Vorwahl und Nummer ein, um Benachrichtigungen zu erhalten.
   d. Wählen Sie Create subscription (Abonnement erstellen) aus.
6. Wenn Sie `Email` auswählen, erhalten Sie eine Nachricht, in der Sie aufgefordert werden, Ihr Abonnement zu bestätigen. Öffnen Sie die Nachricht und befolgen Sie die Anweisungen, um Ihr Abonnement abzuschließen.

Sobald eine neue Version von SSM Agent veröffentlicht wird, senden wir den Abonnenten Benachrichtigungen. Wenn Sie diese Benachrichtigungen nicht mehr erhalten möchten, führen Sie die folgenden Schritte aus, um sich abzumelden.

So melden Sie sich von den SSM Agent-Benachrichtigungen ab

1. Öffnen Sie die Amazon SNS-Konsole.
2. Wählen Sie im Navigationsbereich Subscriptions aus.
3. Wählen Sie das Abonnement und dann Delete (Löschen) aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen) aus.

Fehlerbehebung für SSM Agent

Wenn Sie Probleme bei der Ausführung von Vorgängen auf Ihren verwalteten Knoten haben, liegt möglicherweise ein Problem mit AWS Systems Manager Agent (SSM Agent) vor. Verwenden Sie die folgenden Informationen, um die SSM Agent-Protokolldateien anzuzeigen und Fehler für den Agent zu beheben.

Themen

- SSM Agent ist veraltet
- Probleme mithilfe von SSM Agent-Protokolldateien beheben
- Agent-Protokolldateien werden nicht gedreht (Windows)
- Keine Verbindung mit SSM-Endpunkten möglich
- Verwenden Sie ssm-cli, um die Verfügbarkeit von verwalteten Knoten zu überprüfen

SSM Agent ist veraltet

Eine aktualisierte Version von SSM Agent wird veröffentlicht, wenn neue Funktionen zu Systems Manager hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisen aufGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.

Probleme mithilfe von SSM Agent-Protokolldateien beheben

SSM Agent protokolliert Informationen in den folgenden Dateien. Die Informationen in diesen Dateien können Ihnen auch bei der Problembehandlung behilflich sein. Weitere Informationen zu SSM Agent-Protokolldateien, einschließlich der Aktivierung der Debug-Protokollierung, finden Sie unter Anzeigen von SSM Agent-Protokollen.

> **Note**
>
> Wenn Sie diese Protokolle mithilfe von Windows File Explorer anzeigen möchten, überprüfen Sie, dass Sie die Anzeige von ausgeblendeten Dateien und Systemdateien unter „Folder Options“ (Ordneroptionen) aktiviert ist.

Unter Windows

- `%PROGRAMDATA%\Amazon\SSM\Logs\amazon-ssm-agent.log`
- `%PROGRAMDATA%\Amazon\SSM\Logs\errors.log`

Unter Linux und macOS

- `/var/log/amazon/ssm/amazon-ssm-agent.log`
- `/var/log/amazon/ssm/errors.log`

Für Linux-verwaltete Knoten finden Sie möglicherweise weitere Informationen in der `messages`-Datei, die in das folgende Verzeichnis geschrieben ist: `/var/log`.

Weitere Informationen zur Fehlerbehebung mithilfe von Agentenprotokollen finden Sie unter Wie verwende ich SSM Agent-Protokolle zur Behebung von Problemen mit SSM Agent in meiner verwalteten Instance? im AWS re:POST Knowledge Center.

Agent-Protokolldateien werden nicht gedreht (Windows)

Wenn Sie die datumsbasierte Drehung der Protokolldatei in der Datei seelog.xml (auf von Windows Server verwalteten Knoten) angeben und die Protokolle nicht gedreht werden, geben Sie den `fullname=true`-Parameter an. Hier finden Sie ein Beispiel für eine seelog.xml -Konfigurationsdatei mit angegebenem `fullname=true`-Parameter.

```
<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>
```

Keine Verbindung mit SSM-Endpunkten möglich

SSM Agent muss ausgehenden HTTPS-Verkehr (Port 443) zu den folgenden Endpunkten zulassen:

- `ssm.region.amazonaws.com`
- `ssmmessages.region.amazonaws.com`

Region steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

> **Note**
>
> Vor 2024 `ec2messages.region.amazonaws.com` war dies ebenfalls erforderlich. Bei der AWS-Regionen Veröffentlichung vor 2024 ist es weiterhin erforderlich,

Datenverkehr zuzulassen, dies `ssmmessages.region.amazonaws.com` ist jedoch optionalec2messages.*region*.amazonaws.com.
Für Regionen, die 2024 und später eingeführt wurden, `ssmmessages.region.amazonaws.com` ist die Zulassung von Datenverkehr erforderlich, `ec2messages.region.amazonaws.com` Endpunkte werden für diese Regionen jedoch nicht unterstützt.

SSM Agentfunktioniert nicht, wenn es nicht wie beschrieben mit den vorherigen Endpunkten kommunizieren kann, selbst wenn Sie AWS provided Amazon Machine Images (AMIs) wie Amazon Linux 2 oder Amazon Linux 2023 verwenden. Ihre Netzwerkkonfiguration muss über einen offenen Internetzugang verfügen, oder Sie müssen benutzerdefinierte Virtual Private Cloud (VPC)-Endpunkte konfiguriert haben. Wenn Sie keinen benutzerdefinierten VPC-Endpunkt erstellen möchten, überprüfen Sie Ihre Internet-Gateways oder NAT-Gateways. Weitere Informationen dazu, wie Sie VPC-Endpunkte verwalten, finden Sie unter Verbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

Verwenden Sie **`ssm-cli`**, um die Verfügbarkeit von verwalteten Knoten zu überprüfen

Ab SSM Agent-Version 3.1.501.0 können Sie mit `ssm-cli` feststellen, ob ein verwalteter Knoten die primären Voraussetzungen erfüllt, um von Systems Manager verwaltet zu werden und in den Listen der verwalteten Knoten in Fleet Manager zu erscheinen. Die `ssm-cli` ist ein eigenständiges Befehlszeilentool, das in der SSM Agent-Installation enthalten ist. Es sind vorkonfigurierte Befehle enthalten, die die erforderlichen Informationen sammeln, um Ihnen bei der Diagnose zu helfen, warum eine Amazon-EC2-Instance oder ein Nicht-EC2-Gerät, dessen Betrieb Sie bestätigt haben, nicht in Ihren Listen der verwalteten Knoten im Systems Manager enthalten ist. Diese Befehle werden ausgeführt, wenn Sie die `get-diagnostics`-Option angeben.

Weitere Informationen finden Sie unter Problembehandlung bei der Verfügbarkeit von verwalteten Knoten mit `ssm-cli`.

AWS Systems Manager Quick Setup

Verwenden SieQuick Setup, eine Fähigkeit von AWS Systems Manager, um häufig verwendete Amazon Web Services und -Funktionen schnell mit empfohlenen Best Practices zu konfigurieren. Quick Setupvereinfacht die Einrichtung von Diensten, einschließlich Systems Manager, durch die Automatisierung häufiger oder empfohlener Aufgaben. Zu diesen Aufgaben gehören beispielsweise die Erstellung der erforderlichen Instanzprofilrollen AWS Identity and Access Management (IAM) und die Einrichtung betrieblicher Best Practices wie regelmäßige Patchscans und Inventarerfassung. Bei der Nutzung des Quick Setup-Service fallen keine Kosten an. Abhängig von der Art der von Ihnen eingerichteten Services und den Nutzungsbeschränkungen können jedoch Kosten anfallen, ohne dass Gebühren für die Services anfallen, die für die Einrichtung Ihres Services verwendet werden. Um mit Quick Setup zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Quick Setup aus.

> **Note**
>
> Wenn Sie zu Quick Setup weitergeleitet wurden, um Ihre Instances für die Verwaltung durch Systems Manager zu konfigurieren, führen Sie das Verfahren in Amazon-EC2-Host-Verwaltung aus.

Was sind die Vorteile von Quick Setup?

Quick Setup bietet folgende Vorteile:

- Vereinfachte Service- und Featurekonfiguration

  Quick Setup führt Sie durch die Konfiguration betrieblicher bewährter Methoden und stellt diese Konfigurationen automatisch bereit. Das Quick Setup-Dashboard zeigt eine Echtzeitansicht Ihres Konfigurationsbereitstellungsstatus an.
- Automatisches Bereitstellen von Konfigurationen über mehrere Konten hinweg

  Sie können es einzeln AWS-Konto oder Quick Setup in mehreren verwenden AWS-Konten und AWS-Regionen durch Integration mit AWS Organizations. Durch Verwenden von Quick Setup über mehrere Konten hinweg können Sie sicherstellen, dass Ihre Organisation konsistente Konfigurationen aufrechterhält.
- Beseitigen von Konfigurationsabweichungen

Die Konfigurationsabweichung tritt auf, wenn ein Benutzer Änderungen an einem Service oder Feature vornimmt, die mit der Auswahl über Quick Setup im Konflikt stehen. Quick Setup überprüft regelmäßig auf Konfigurationsabweichung und versucht, diese zu beheben.

An wen richtet sich Quick Setup?

Quick Setup ist besonders vorteilhaft für Kunden, die bereits Erfahrung mit den von ihnen eingerichteten Services und Features haben und ihren Einrichtungsprozess vereinfachen möchten. Wenn Sie mit dem, mit dem AWS-Service Sie die Konfiguration durchführen, nicht vertraut sindQuick Setup, empfehlen wir Ihnen, mehr über den Service zu erfahren. Überprüfen Sie den Inhalt im entsprechenden Benutzerhandbuch, bevor Sie eine Konfiguration mit Quick Setuperstellen.

Verfügbarkeit von Quick Setup in AWS-Regionen

Im Folgenden AWS-Regionen können Sie alle Quick Setup Konfigurationstypen für eine gesamte Organisation, wie unter konfiguriert AWS Organizations, oder nur für die von Ihnen ausgewählten Organisationskonten und Regionen verwenden. Sie können Quick Setup in diesen Regionen auch mit nur einem einzigen Konto verwenden.

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Europa (Frankfurt)
- Europa (Stockholm)
- Europa (Irland)
- Europe (London)

- Europe (Paris)
- Südamerika (São Paulo)

In den folgenden Regionen ist nur der Konfigurationstyp Host-Verwaltung für einzelne Konten verfügbar:

- Europa (Milan)
- Asien-Pazifik (Hongkong)
- Naher Osten (Bahrain)
- China (Peking)
- China (Ningxia)
- AWS GovCloud (USA-Ost)
- AWS GovCloud (US-West)

Eine Liste aller von Systems Manager unterstützten Regionen finden Sie im Allgemeine Amazon Web Services-Referenz unter Service-Endpunkte von Systems Manager in der Spalte Region.

Erste Schritte mit Quick Setup

Verwenden Sie die Informationen in diesem Thema, um sich auf die Verwendung von Quick Setup vorzubereiten.

Themen

- Konfigurieren der Heimat- AWS-Region
- IAM-Rollen und -Berechtigungen für das Quick Setup-Onboarding

Konfigurieren der Heimat- AWS-Region

Um mit Quick Setup einer Fähigkeit von zu beginnen AWS Systems Manager, müssen Sie ein Zuhause auswählen AWS-Region und dann mit einsteigenQuick Setup. In der Heimatregion werden die AWS Ressourcen Quick Setup erstellt, die für die Bereitstellung Ihrer Konfigurationen verwendet werden. Nachdem Sie die Heimatregion ausgewählt haben, kann sie nicht mehr geändert werden.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie unter Wählen Sie eine Heimatregion aus, AWS-Region in der Sie die AWS Ressourcen erstellen Quick Setupmöchten, die für die Bereitstellung Ihrer Konfigurationen verwendet werden.
4. Wählen Sie Erste Schritte.

Um mit der Nutzung von Quick Setup zu beginnen, wählen Sie einen Service oder ein Feature aus der Liste der verfügbaren Konfigurationstypen. Ein Konfigurationstyp in Quick Setup ist spezifisch für eine AWS-Service Oder-Funktion. Wenn Sie einen Konfigurationstyp auswählen, wählen Sie die Optionen aus, die Sie für diesen Service oder dieses Feature konfigurieren möchten. Standardmäßig helfen Ihnen Konfigurationstypen, den Service oder das Feature so einzurichten, dass empfohlene bewährte Methoden verwendet werden.

Nach dem Einrichten einer Konfiguration können Sie Details zu dieser Konfiguration und ihrem Bereitstellungsstatus über Organisationseinheiten (OUs) und Regionen hinweg anzeigen. Sie können auch den State Manager Zuordnungsstatus für die Konfiguration anzeigen. State Managerist eine Fähigkeit von AWS Systems Manager. Im Bereich Configuration details (Konfigurationsdetails) können Sie eine Zusammenfassung der Quick Setup-Konfiguration anzeigen. Diese Zusammenfassung enthält Services aus allen Konten und allen erkannten Konfigurationsabweichungen.

IAM-Rollen und -Berechtigungen für das Quick Setup-Onboarding

Quick SetupErstellt während des Onboardings die folgenden AWS Identity and Access Management (IAM-) Rollen in Ihrem Namen:

- `AWS-QuickSetup-StackSet-Local-ExecutionRole`— Gewährt AWS CloudFormation -Berechtigungen, um eine beliebige Vorlage zu verwenden.
- `AWS-QuickSetup-StackSet-Local-AdministrationRole`— Erteilt Berechtigungen AWS CloudFormation zur Übernahme. `AWS-QuickSetup-StackSet-Local-ExecutionRole`

Wenn Sie ein Verwaltungskonto einrichten — das Konto, mit dem Sie eine Organisation erstellen AWS Organizations— werden in Ihrem Quick Setup Namen auch die folgenden Rollen erstellt:

- `AWS-QuickSetup-SSM-RoleForEnablingExplorer`— Erteilt Berechtigungen dem `AWS-EnableExplorer`-Automation-Runbook. Das `AWS-EnableExplorer` Runbook

konfiguriertExplorer, eine Funktion von Systems Manager, so, dass Informationen für mehrere AWS-Konten und angezeigt werden. AWS-Regionen

- `AWSServiceRoleForAmazonSSM`— Eine dienstbezogene Rolle, die Zugriff auf AWS Ressourcen gewährt, die von Systems Manager verwaltet und verwendet werden.
- `AWSServiceRoleForAmazonSSM_AccountDiscovery`— Eine dienstbezogene Rolle, die Systems Manager die Berechtigung erteilt, beim Synchronisieren von Daten anzurufen AWS-Services , um AWS-Konto Informationen zu ermitteln. Weitere Informationen finden Sie unter Informationen über die `AWSServiceRoleForAmazonSSM_AccountDiscovery`-Rolle.

Quick SetupErmöglicht beim Onboarding eines Verwaltungskontos den vertrauenswürdigen Zugriff zwischen AWS Organizations und CloudFormation die Bereitstellung von Quick Setup Konfigurationen im gesamten Unternehmen. Um vertrauenswürdigen Zugriff zu aktivieren, muss Ihr Verwaltungskonto über Administratorberechtigungen verfügen. Nach dem Onboarding benötigen Sie keine Administratorberechtigungen mehr. Weitere Informationen finden Sie unter Enable trusted access with Organizations (Aktivieren des vertrauenswürdigen Zugriffs mit Organizations).

Informationen zu AWS Organizations Kontotypen finden Sie unter AWS Organizations Terminologie und Konzepten im AWS Organizations Benutzerhandbuch.

> **Note**
>
> Quick Setupverwendet AWS CloudFormation StackSets , um Ihre Konfigurationen AWS-Konten regionsübergreifend bereitzustellen. Wenn die Anzahl der Zielkonten multipliziert mit der Anzahl der Regionen 10 000 übersteigt, kann die Konfiguration nicht bereitgestellt werden. Wir empfehlen Ihnen, Ihren Anwendungsfall zu überprüfen und Konfigurationen zu erstellen, die weniger Ziele verwenden, um dem Wachstum Ihres Unternehmens Rechnung zu tragen. Stack-Instances werden nicht für das Verwaltungskonto Ihrer Organisation bereitgestellt. Weitere Informationen finden Sie unter Considerations when creating a stack set with service-managed permissions (Überlegungen beim Erstellen eines Stack-Sets mit service-verwalteten Berechtigungen).

Sie können alle Features von Quick Setup nutzen, wenn Ihr Benutzer, Ihre Gruppe oder Rolle Zugriff auf die in der folgenden Tabelle aufgeführten API-Operationen hat. Es gibt zwei Registerkarten für API-Operationen, eine für alle Konten und eine für die zusätzlichen Berechtigungen, die Sie für das Verwaltungskonto Ihrer Organisation benötigen.

Non-management account

```
"iam:CreateRole",
"iam:AttachRolePolicy",
"iam:PutRolePolicy",
"iam:GetRole",
"iam:ListRoles",
"iam:PassRole"
"ssm:ListAssociations",
"ssm:ListDocuments",
"ssm:GetDocument",
"ssm:DescribeAssociation",
"ssm:DescribeAutomationExecutions",
"cloudformation:DescribeStackSet",
"cloudformation:DescribeStackInstance",
"cloudformation:DescribeStacks",
"cloudformation:DescribeStackResources",
"cloudformation:ListStackSetOperations",
"cloudformation:ListStackSets",
"cloudformation:ListStacks",
"cloudformation:ListStackInstances",
"cloudformation:ListStackSetOperationResults",
"cloudformation:TagResource",
"cloudformation:CreateStack",
"cloudformation:DeleteStackSet",
"cloudformation:UpdateStackSet",
"cloudformation:CreateStackSet",
"cloudformation:DeleteStackInstances",
"cloudformation:CreateStackInstances"
```

Management account

```
"ssm:createResourceDataSync",
"ssm:listResourceDataSync",
"ssm:getOpsSummary",
"ssm:createAssociation",
"ssm:createDocument",
"ssm:startAssociationsOnce",
"ssm:startAutomationExecution",
"ssm:updateAssociation",
"ssm:listAssociations",
```

```
"ssm:listDocuments",
"ssm:getDocument",
"ssm:describeAssociation",
"ssm:describeAutomationExecutions",
"organizations:ListRoots",
"organizations:DescribeOrganization",
"organizations:ListOrganizationalUnitsForParent"
"organizations:EnableAWSServiceAccess",
"cloudformation:describe*"
```

Verwenden von Quick Setup

Quick Setup, eine Funktion von AWS Systems Manager, zeigt die Ergebnisse der einzelnen Konfigurationen in der Tabelle Configurations (Konfigurationen) auf der Homepage von Quick Setup. Auf dieser Seite können Sie über View details (Details anzeigen) die Details jeder Konfiguration anzeigen, Konfigurationen über das Dropdown-Menü Actions (Aktionen) löschen oder über Create (Erstellen) Konfigurationen erstellen. Die Tabelle Configurations (Konfigurationen) enthält die folgenden Informationen:

- Configuration type (Konfigurationstyp) – Der Konfigurationstyp, der beim Erstellen der Konfiguration ausgewählt wurde.
- Bereitstellungstyp – Gibt an, ob die Bereitstellung für die gesamte Organisation (`Organizational`) oder nur für Ihr Konto (`Local`) gilt.
- Organizational units (Organisationseinheiten) – Zeigt die Organisationseinheiten, für die die Konfiguration bei Auswahl von Custom (benutzerdefinierten) Zielen bereitgestellt wird. Organisationseinheiten und benutzerdefinierte Ziele stehen nur im Verwaltungskonto Ihrer Organisation zur Verfügung. Das Verwaltungskonto ist das Konto, das Sie zum Erstellen einer Organisation in AWS Organizations verwenden.
- Regions (Regionen) – Die Regionen, in denen die Konfiguration bereitgestellt wird, wenn Sie Custom (benutzerdefinierte) Ziele oder Ziele in Ihrem Current account (aktuellem Konto) auswählen.
- Deployment status (Bereitstellungsstatus) – Der Bereitstellungsstatus zeigt an, ob AWS CloudFormation die Ziel- oder Stack-Instance erfolgreich bereitgestellt hat. Die Ziel- und Stack-Instances enthalten die Konfigurationsoptionen, die Sie bei der Erstellung der Konfiguration ausgewählt haben.

- Association status (Zuordnungsstatus) – Der Zuordnungsstatus ist der Status aller Zuordnungen, die durch die von Ihnen erstellte Konfiguration generiert wurden. Die Zuordnungen für alle Ziele müssen erfolgreich ausgeführt werden, andernfalls lautet der Status Failed (Fehlgeschlagen).

  Quick Setup erstellt und führt eine State Manager-Zuordnung für jedes Konfigurationsziel aus. State Manager ist eine Funktion von AWS Systems Manager.

Konfigurationsdetails

Die Seite Configuration details (Konfigurationsdetails) zeigt Informationen über die Bereitstellung der Konfiguration und die entsprechenden Zuordnungen. Auf dieser Seite können Sie Konfigurationsoptionen bearbeiten, Ziele aktualisieren oder die Konfiguration löschen. Außerdem können Sie die Details der einzelnen Konfigurationsbereitstellungen anzeigen, um weitere Informationen über die Zuordnungen zu erhalten.

Je nach Art der Konfiguration wird eines oder mehrere der folgenden Statusdiagramme angezeigt:

Configuration deployment status (Status der Konfigurationsbereitstellungen)

Zeigt die Anzahl der Bereitstellungen, die erfolgreich waren, fehlgeschlagen sind, ausgeführt werden oder noch ausstehen. Die Bereitstellungen erfolgen in den angegebenen Zielkonten und Regionen, die von der Konfiguration betroffene Knoten enthalten.

Configuration association status (Status der Konfigurationszuordnung)

Zeigt die Anzahl der State Manager-Zuordnungen, die erfolgreich waren, fehlgeschlagen sind oder noch ausstehen. Quick Setup erstellt für die ausgewählten Konfigurationsoptionen eine Zuordnung in jeder Bereitstellung.

Einrichtung des Status

Zeigt die Anzahl der vom Konfigurationstyp durchgeführten Aktionen und ihren aktuellen Status an.

Ressourcen-Compliance

Zeigt die Anzahl der Ressourcen an, die mit der angegebenen Konfigurationsrichtlinie konform sind.

Die Seite Configuration details (Konfigurationsdetails) zeigt Informationen über die Bereitstellung Ihrer Konfiguration. Weitere Details zu den einzelnen Bereitstellungen können Sie anzeigen, indem

Sie eine Bereitstellung auswählen und dann auf View details (Details anzeigen) klicken. Auf der Detailseite werden die Zuordnungen angezeigt, die für die Knoten in der jeweiligen Bereitstellung bereitgestellt wurden.

Bearbeiten und Löschen Ihrer Konfiguration

Die Konfigurationsoptionen einer Konfiguration können Sie auf der Seite Configuration details (Konfigurationsdetails) bearbeiten. Dazu wählen Sie Actions (Aktionen) und dann Edit configuration options (Konfigurationsoptionen bearbeiten) aus. Wenn Sie der Konfiguration neue Optionen hinzufügen, führt Quick Setup Ihre Bereitstellungen aus und erstellt neue Zuordnungen. Wenn Sie Optionen aus einer Konfiguration entfernen, führt Quick Setup die Bereitstellungen aus und entfernt alle entsprechenden Zuordnungen.

> **Note**
>
> Die Quick Setup-Konfigurationen für Ihr Konto können Sie jederzeit ändern. Um die Konfiguration einer Organisation zu bearbeiten, muss Configuration status (Konfigurationsstatus) entweder Success (Erfolg) oder Failed (Fehlgeschlagen) lauten.

Außerdem können Sie die Ziele in Ihren Konfigurationen aktualisieren. Dazu wählen Sie Actions (Aktionen) und Add OUs (Organisationseinheiten hinzufügen), Add Regions (Regionen hinzufügen), Remove OUs (Organisationseinheiten entfernen) oder Remove Regions (Regionen entfernen) aus. Wenn Ihr Konto nicht als Verwaltungskonto konfiguriert ist oder Sie die Konfiguration nur für das aktuelle Konto erstellt haben, können Sie die Ziel-Organisationseinheiten nicht aktualisieren. Beim Entfernen einer Region oder Organisationseinheit werden deren Zuordnungen ebenfalls entfernt.

Sie können eine Konfiguration aus Quick Setup löschen, indem Sie die Konfiguration, danach Actions (Aktionen) und dann Delete configuration (Konfiguration löschen) auswählen. Alternativ können Sie die Konfiguration auf der Seite Configuration details (Konfigurationsdetails) unter dem Dropdown-Menü Actions (Aktionen) über die Option Delete configuration (Konfiguration löschen) löschen. Quick Setup fordert Sie dann auf, Remove all OUs and Regions (Alle Organisationseinheiten und Regionen zu entfernen). Dies kann einige Zeit in Anspruch nehmen. Beim Löschen einer Konfiguration werden alle entsprechenden Zuordnungen ebenfalls gelöscht. Bei diesem zweistufigen Löschvorgang werden alle bereitgestellten Ressourcen aus allen Konten und Regionen entfernt. Anschließend wird die Konfiguration gelöscht.

Compliance von Konfigurationen

Sie können anzeigen, ob Ihre Instances den von Ihren Konfigurationen erstellten Zuordnungen entsprechen, und zwar in Explorer oder Compliance, die beide Funktionen von AWS Systems Manager sind. Weitere Informationen zur Compliance finden Sie unter Arbeiten mit Compliance. Weitere Informationen zum Anzeigen der Compliance in Explorer finden Sie unter AWS Systems Manager Explorer.

Unterstützte Quick Setup-Konfigurationstypen

Unterstützte Konfigurationstypen

Quick Setup bietet Unterstützung für die folgenden Konfigurationstypen.

- Amazon-EC2-Host-Verwaltung
- Standard-Host-Verwaltung für eine Organisation
- AWS Config Configuration Recorder
- AWS Config Bereitstellung von Konformitätspaketen
- Patch Manager Patching-Konfiguration der Organisation
- Einrichtung der Organisation durch Change Manager
- DevOpsGuru-Konfiguration
- Distributor-Paket-Bereitstellung
- Amazon-EC2-Instance-Resource-Scheduler
- Einrichtung der Organisation durch OpsCenter
- AWS Ressourcen Explorer Konfiguration

Amazon-EC2-Host-Verwaltung

Verwenden SieQuick Setup, eine Funktion von AWS Systems Manager, um schnell erforderliche Sicherheitsrollen und häufig verwendete Systems Manager Manager-Funktionen auf Ihren Amazon Elastic Compute Cloud (Amazon EC2) -Instances zu konfigurieren. Sie können es Quick Setup in einem einzelnen Konto oder über mehrere Konten hinweg und AWS-Regionen durch Integration mit AWS Organizations verwenden. Diese Funktionen helfen Ihnen, den Zustand Ihrer Instances zu verwalten und zu überwachen und gleichzeitig das Minimum an erforderlichen Berechtigungen für den Einstieg bereitzustellen.

Wenn Sie mit den Services und Features von Systems Manager nicht vertraut sind, empfehlen wir, das AWS Systems Manager -Benutzerhandbuch zu lesen, bevor Sie eine Konfiguration mit Quick Setup erstellen. Weitere Informationen zur Systems Manager finden Sie unter Was ist AWS Systems Manager?.

Important

Quick Setup ist möglicherweise nicht das richtige Tool für das EC2-Management, wenn einer der folgenden Punkte auf Sie zutrifft:

- Sie versuchen zum ersten Mal, eine EC2-Instance zu erstellen, um AWS Funktionen auszuprobieren.
- Sie haben noch nicht viel Erfahrung mit EC2-Instance-Management.

Stattdessen empfehlen wir Ihnen, die folgenden Inhalte zu untersuchen:

- Erste Schritte mit Amazon EC2
- Starten Sie eine Instance mithilfe des Assistenten zum Starten neuer Instances im Amazon EC2-Benutzerhandbuch
- Starten Sie eine Instance mithilfe des Assistenten zum Starten neuer Instances im Amazon EC2-Benutzerhandbuch
- Tutorial: Erste Schritte mit Amazon EC2 EC2-Linux-Instances im Amazon EC2 EC2-Benutzerhandbuch

Wenn Sie bereits mit dem EC2-Instance-Management vertraut sind und die Konfiguration und Verwaltung für mehrere EC2-Instances optimieren möchten, verwenden Sie Quick Setup. Unabhängig davon, ob Ihr Unternehmen über Dutzende, Tausende oder Millionen von EC2-Instances verfügt, verwenden Sie das folgende Quick Setup-Verfahren, um mehrere Optionen gleichzeitig für sie zu konfigurieren.

Voraussetzungen

Die Heimatregion für Quick Setup muss angegeben werden, bevor Sie die folgenden Aufgaben ausführen. Weitere Informationen finden Sie unter Konfigurieren der Heimat- AWS-Region.

> **Note**
>
> Mit diesem Konfigurationstyp können Sie mehrere Optionen für eine gesamte Organisation festlegen AWS Organizations, die in nur einigen Organisationskonten und Regionen oder für ein einzelnes Konto definiert ist. Eine dieser Optionen besteht darin, alle zwei Wochen nach Updates für SSM Agent zu suchen und diese anzuwenden. Organisationsadministratoren können auch alle EC2-Instances in ihrer Organisation alle zwei Wochen mit Agenten-Updates aktualisieren, indem sie die Standardkonfiguration für die Host-Verwaltung verwenden. Weitere Informationen finden Sie unter Standard-Host-Verwaltung für eine Organisation.

Konfiguration von Host-Management-Optionen für EC2-Instances

Um die Hostverwaltung einzurichten, führen Sie die folgenden Aufgaben in der AWS Systems Manager Quick Setup Konsole aus.

Um die Host-Management-Konfigurationsseite zu öffnen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Host-Verwaltung die Option Erstellen aus.

> **Tip**
>
> Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

So konfigurieren Sie die Hostverwaltungsoptionen von Systems Manager

- Um die Systems Manager Manager-Funktionalität zu konfigurieren, wählen Sie im Abschnitt Konfigurationsoptionen die Optionen in der Systems Manager Manager-Gruppe aus, die Sie für Ihre Konfiguration aktivieren möchten:

Aktualisieren Sie den Systems Manager (SSM) -Agenten alle zwei Wochen

Ermöglicht Systems Manager, alle zwei Wochen nach einer neuen Version des Agenten zu suchen. Wenn es eine neue Version gibt, aktualisiert Systems Manager den Agenten auf Ihrem verwalteten Knoten automatisch auf die neueste freigegebene Version. Quick Setup installiert den Agenten nicht auf Instances, auf denen er nicht bereits vorhanden ist. Informationen darüber, welche AMIs SSM Agent vorinstalliert haben, finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass auf Ihren Knoten immer die neueste up-to-date Version von ausgeführt wirdSSM Agent. Weitere Informationen zu SSM Agent, einschließlich Informationen zur manuellen Installation des Agenten, finden Sie unter Arbeiten mit SSM Agent.

Sammeln Sie alle 30 Minuten Inventar von Ihren Instances

Ermöglicht Quick Setup die Konfiguration der Erfassung der folgenden Arten von Metadaten:

- AWS Komponenten — EC2-Treiber, Agenten, Versionen und mehr.
- Anwendungen – Anwendungsnamen, Publisher, Versionen und mehr.
- Knoten-Details – Systemname, Name des Betriebssystems (OS), OS-Version, letzter Boot-Vorgang, DNS, Domain, Arbeitsgruppe, OS-Architektur und mehr.
- Netzwerkkonfiguration – IP-Adresse, MAC-Adresse, DNS, Gateway, Subnetzmaske und mehr.
- Services – Name, Anzeigename, Status, abhängige Services, Servicetyp, Starttyp und mehr (nur Windows Server-Knoten).
- Windows-Rollen – Name, Anzeigename, Pfad, Featuretyp, Installationsstatus und mehr (nur Windows Server-Knoten).
- Windows-Updates – Hotfix-ID, installiert von, Installationsdatum und mehr (nur Windows Server-Knoten).

Für weitere Informationen über Inventory, eine Funktion von AWS Systems Manager, finden Sie unter AWS Systems Manager-Bestand.

> **Note**
>
> Die Bestandserfassungs-Option kann bis zu 10 Minuten dauern, auch wenn Sie nur wenige Knoten ausgewählt haben.

Tägliches Scannen von Instances nach fehlenden Patches

ErmöglichtPatch Manager, eine Funktion von Systems Manager, Ihre Knoten täglich zu scannen und einen Bericht auf der Compliance-Seite zu erstellen. Der Bericht zeigt, wie viele Knoten entsprechend der Standard-Patch-Baseline patchkompatibel sind. Der Bericht enthält eine Liste der einzelnen Knoten und deren Compliance-Status.

Informationen zu Patching-Vorgängen und Patch-Baselines finden Sie unter AWS Systems Manager Patch Manager.

Informationen zur Patch-Compliance finden Sie auf der Seite Systems-Manager-Compliance.

Informationen zum Patchen verwalteter Knoten in mehreren Konten und Regionen in einer Konfiguration finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien und Patch Manager Patching-Konfiguration der Organisation.

> **Important**
>
> Systems Manager unterstützt mehrere Methoden zum Scannen verwalteter Knoten auf Patch-Compliance. Wenn Sie mehr als eine dieser Methoden gleichzeitig implementieren, sind die angezeigten Patch-Compliance-Informationen immer das Ergebnis des letzten Scans. Ergebnisse früherer Scans werden überschrieben. Wenn die Scan-Methoden unterschiedliche Patch-Baselines mit unterschiedlichen Genehmigungsregeln verwenden, können sich die Informationen zur Patch-Compliance unerwartet ändern. Weitere Informationen finden Sie unter Vermeiden von unbeabsichtigtem Überschreiben von Patch-Compliance-Daten.

So konfigurieren Sie CloudWatch Amazon-Host-Management-Optionen

- Um die CloudWatch Funktionalität zu konfigurieren, wählen Sie im Abschnitt Konfigurationsoptionen die Optionen in der CloudWatchAmazon-Gruppe aus, die Sie für Ihre Konfiguration aktivieren möchten:

  Installieren und konfigurieren Sie den CloudWatch Agenten

  Installiert die Basiskonfiguration des Unified CloudWatch Agents auf Ihren Amazon EC2 EC2-Instances. Der Agent sammelt Metriken und Protokolldateien von Ihren Instances für Amazon CloudWatch. Diese Informationen werden zusammengefasst, damit Sie den Zustand Ihrer Instances schnell bestimmen können. Weitere Informationen zur Basiskonfiguration des CloudWatch Agenten finden Sie unter Vordefinierte Metriksätze für CloudWatch Agenten. Es können zusätzliche Kosten anfallen. Weitere Informationen finden Sie unter CloudWatchAmazon-Preise.

  Aktualisieren Sie den CloudWatch Agenten einmal alle 30 Tage

  Ermöglicht Systems Manager, alle 30 Tage nach einer neuen Version des CloudWatch Agenten zu suchen. Wenn es eine neue Version gibt, aktualisiert Systems Manager den Agent automatisch auf Ihrer Instance. Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass auf Ihren Instances immer die neueste up-to-date Version des CloudWatch Agenten ausgeführt wird.

So konfigurieren Sie die Hostverwaltungsoptionen für Amazon EC2 Launch Agent

- Um die Amazon EC2 Launch Agent-Funktionalität zu konfigurieren, wählen Sie im Abschnitt Konfigurationsoptionen die Optionen in der Amazon EC2 Launch Agent-Gruppe aus, die Sie für Ihre Konfiguration aktivieren möchten:

  Aktualisieren Sie den EC2 Launch Agent einmal alle 30 Tage

  Ermöglicht Systems Manager, alle 30 Tage nach einer neuen Version des Launch-Agenten zu suchen, der auf Ihrer Instance installiert ist. Wenn eine neue Version verfügbar ist, aktualisiert Systems Manager den Agenten auf Ihrer Instance. Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass auf Ihren Instances immer die neueste up-to-

date Version des jeweiligen Launch-Agents ausgeführt wird. Für Amazon-EC2-Windows-Instances unterstützt diese Option EC2Launch, EC2Launch v2 und EC2Config. Für Amazon-EC2-Linux-Instances unterstützt diese Option `cloud-init`. Für Amazon-EC2-Mac-Instances unterstützt diese Option `ec2-macos-init`. Quick Setup unterstützt nicht die Aktualisierung von Startagenten, die auf Betriebssystemen, die nicht vom Startagenten unterstützt werden, oder auf AL2023 installiert sind.

Weitere Informationen zu diesen Initialisierungsagenten finden Sie in den folgenden Themen:

- Konfigurieren einer Windows-Instance mithilfe von EC2Launch v2
- Konfigurieren einer Windows-Instance mithilfe von EC2Launch
- Konfigurieren einer Windows-Instance mithilfe des EC2Config-Service
- Cloud-Init-Dokumentation
- ec2-macos-init

Um die EC2-Instances auszuwählen, die durch die Host-Management-Konfiguration aktualisiert werden sollen

- Wählen Sie im Abschnitt Ziele die Methode aus, um die Konten und Regionen zu bestimmen, in denen die Konfiguration bereitgestellt werden soll:

  > **Note**
  >
  > Sie können nicht mehrere Quick Setup Host Management-Konfigurationen erstellen, die auf dieselbe AWS-Region abzielen.

  Entire organization

  Ihre Konfiguration wird in allen Organisationseinheiten (OUs) und AWS-Regionen in Ihrer Organisation bereitgestellt.

  > **Note**
  >
  > Die Option Entire organization (Gesamte Organisation) ist nur verfügbar, wenn Sie die Hostverwaltung über das Verwaltungskonto Ihrer Organisation konfigurieren.

Custom

1. Wählen Sie im Abschnitt Ziel-Organisationseinheiten die Organisationseinheiten aus, in denen Sie diese Hostverwaltungskonfiguration bereitstellen möchten.
2. Wählen Sie im Abschnitt Zielregionen die Regionen aus, in denen Sie diese Hostverwaltungskonfiguration bereitstellen möchten.

Current account

Wählen Sie eine der Regionsoptionen und folgen Sie den Schritten für diese Option.

Aktuelle Region

Wählen Sie aus, wie Sie nur auf Instances in der aktuellen Region abzielen möchten:

- Alle Instances — Die Host-Management-Konfiguration zielt automatisch auf alle EC2 in der aktuellen Region ab.
- Tag — Wählen Sie Hinzufügen und geben Sie den Schlüssel und den optionalen Wert ein, der den Instances hinzugefügt wird, auf die zugegriffen werden soll.
- Ressourcengruppe — Wählen Sie unter Ressourcengruppe eine bestehende Ressourcengruppe aus, die die EC2-Instances enthält, auf die zugegriffen werden soll.
- Manuell — Aktivieren Sie im Abschnitt Instances das Kontrollkästchen jeder EC2-Instance, auf die Sie abzielen möchten.

Wählen Sie Regionen

Wählen Sie aus einer der folgenden Optionen aus, wie Instances in der von Ihnen angegebenen Region als Ziel ausgewählt werden sollen:

- Alle Instances — Alle Instances in den von Ihnen angegebenen Regionen sind als Ziel vorgesehen.
- Tag — Wählen Sie Hinzufügen und geben Sie den Schlüssel und den optionalen Wert ein, der den Instances hinzugefügt wurde, für die das Targeting vorgesehen ist.

Wählen Sie im Abschnitt Zielregionen die Regionen aus, in denen Sie diese Host-Management-Konfiguration bereitstellen möchten.

Um eine Instanzprofiloption anzugeben

- Nur die gesamte Organisation und benutzerdefinierte Ziele.

  Wählen Sie im Abschnitt Instanzprofiloptionen aus, ob Sie die erforderlichen IAM-Richtlinien zu den vorhandenen Instanzprofilen hinzufügen möchten, die Ihren Instances zugeordnet sind, oder ob Sie die Erstellung der IAM-Richtlinien und Instanzprofile mit den für die von Ihnen ausgewählten Konfiguration erforderlichen Berechtigungen zulassen Quick Setup möchten.

Nachdem Sie alle Ihre Konfigurationsoptionen angegeben haben, wählen Sie Create.

Standard-Host-Verwaltung für eine Organisation

Mit Quick Setup einer Funktion von AWS Systems Manager können Sie die Standard-Hostverwaltungskonfiguration für alle Konten und Regionen aktivieren, die Ihrer Organisation in hinzugefügt wurden AWS Organizations. Dadurch wird sichergestellt, dass SSM Agent über alle EC2-Instances (Amazon Elastic Compute Cloud) in der Organisation auf dem Laufenden gehalten wird und dass diese sich mit dem Systems Manager verbinden können.

Bevor Sie beginnen

Stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt sind, bevor Sie diese Einstellung aktivieren.

- Die Heimatregion für Quick Setup muss angegeben werden, bevor Sie die folgenden Aufgaben ausführen. Weitere Informationen finden Sie unter Konfigurieren der Heimat- AWS-Region.
- Die neueste Version von SSM Agent ist bereits auf allen EC2-Instances installiert, die in Ihrer Organisation verwaltet werden sollen.
- Ihre zu verwaltenden Instances verwenden Instance Metadata Service Version 2 (IMDSv2).
- Sie sind mit einer AWS Identity and Access Management (IAM-) Identität (Benutzer, Rolle oder Gruppe) mit Administratorrechten beim Verwaltungskonto für Ihre Organisation angemeldet, wie unter angegeben. AWS Organizations

Verwenden der standardmäßigen EC2-Instance-Verwaltungsrolle

Die Standardkonfiguration für die Host-Verwaltung verwendet die `default-ec2-instance-management-role`-Diensteinstellung für Systems Manager. Dies ist eine Rolle mit Berechtigungen, die Sie allen Konten in Ihrer Organisation zur Verfügung stellen möchten, um die Kommunikation

zwischen SSM Agent auf der Instance und dem Systems-Manager-Dienst in der Cloud zu ermöglichen.

Wenn Sie diese Rolle bereits mit dem CLI-Befehl update-service-setting festgelegt haben, verwendet die Standardkonfiguration für die Host-Verwaltung diese Rolle. Wenn Sie diese Rolle noch nicht festgelegt haben, erstellt Quick Setup sie und wendet sie für Sie an.

Überprüfen Sie mit dem get-service-setting-Befehl, ob diese Rolle bereits für Ihre Organisation festgelegt wurde.

Aktivieren Sie automatische 14-tägige Updates von SSM Agent.

Gehen Sie wie folgt vor, um die Option „Standard-Host-Management-Konfiguration“ für Ihre gesamte AWS Organizations Organisation zu aktivieren.

So aktivieren Sie automatische 14-tägige Updates von SSM Agent

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Standardkonfiguration für die Host-Verwaltung die Option Erstellen aus.

   > **Tip**
   >
   > Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Wählen Sie im Abschnitt Konfigurationsoptionen die Option Automatische Updates von SSM Agent alle zwei Wochen aktivieren aus.
5. Wählen Sie Create (Erstellen) aus.

AWS Config Configuration Recorder

Mit Quick Setup der Funktion von können Sie schnell einen Konfigurationsrekorder erstellen AWS Systems Manager, der von unterstützt wird AWS Config. Verwenden Sie den Konfigurations-Recorder zur Erkennung von Änderungen an Ihren Ressourcenkonfigurationen und zur Erfassung der Änderungen als Konfigurationselemente. Wenn Sie mit dem Service nicht vertraut sind AWS

Config, empfehlen wir Ihnen, mehr über den Service zu erfahren, indem Sie den Inhalt des AWS Config Entwicklerhandbuchs lesen, bevor Sie eine Konfiguration mit Quick Setup erstellen. Weitere Informationen zu finden Sie AWS Config unter Was ist AWS Config? im AWS Config Entwicklerhandbuch.

Standardmäßig zeichnet der Konfigurationsrekorder alle unterstützten Ressourcen in dem Bereich auf, in AWS-Region dem er ausgeführt AWS Config wird. Sie können die Konfiguration so anpassen, dass nur die von Ihnen angegebenen Ressourcentypen aufgezeichnet werden. Weitere Informationen finden Sie im AWS Config Entwicklerhandbuch unter Auswahl der AWS Config Ressourceneinträge.

Wenn Sie mit der Aufzeichnung von Konfigurationen AWS Config beginnen, werden Ihnen Gebühren für die Nutzung des Dienstes berechnet. Preisinformationen finden Sie unter AWS Config Preise.

> **Note**
>
> Wenn Sie bereits einen Konfigurationsrekorder erstellt haben, stoppt Quick Setup die Aufzeichnung nicht und nimmt auch keine Änderungen an den Ressourcentypen vor, die Sie bereits aufzeichnen. Wenn Sie zusätzliche Ressourcentypen mit Quick Setup aufzeichnen möchten, hängt der Service sie an Ihre vorhandenen Rekordergruppen an. Durch das Löschen des Konfigurationstyps Quick Setup Config recording (Config-Aufzeichnung) wird der Konfigurations-Recorder nicht gestoppt. Änderungen werden weiterhin aufgezeichnet, und die Servicenutzungsgebühren fallen an, bis Sie den Konfigurations-Recorder beenden. Weitere Informationen zur Verwaltung des Konfigurations-Recorders finden Sie unter Managing the Configuration Recorder (Verwalten des Konfigurations-Recorders) im AWS Config Developer Guide.

Voraussetzungen

Die Heimatregion für Quick Setup muss angegeben werden, bevor Sie die folgenden Aufgaben ausführen. Weitere Informationen finden Sie unter Konfigurieren der Heimat- AWS-Region.

Führen Sie die folgenden Aufgaben in der AWS Systems Manager Konsole aus, um die AWS Config Aufzeichnung einzurichten.

Um die AWS Config Aufnahme einzurichten mit Quick Setup

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Konfigurationsaufnahme die Option Erstellen aus.

> **Tip**
>
> Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Gehen Sie im Abschnitt Konfigurationsoptionen wie folgt vor:

   a. Geben Sie unter Wählen Sie die aufzuzeichnenden AWS Ressourcentypen an, ob alle unterstützten Ressourcen oder nur die von Ihnen ausgewählten Ressourcentypen aufgezeichnet werden sollen.

   b. Geben Sie unter Versandeinstellungen an, ob Sie einen neuen Amazon Simple Storage Service (Amazon S3) -Bucket erstellen oder einen vorhandenen Bucket auswählen möchten, an den Konfigurations-Snapshots gesendet werden sollen.

   c. Wählen Sie unter Benachrichtigungsoptionen die von Ihnen bevorzugte Benachrichtigungsoption aus. AWS Config verwendet Amazon Simple Notification Service (Amazon SNS), um Sie über wichtige AWS Config Ereignisse im Zusammenhang mit Ihren Ressourcen zu informieren. Wenn Sie die Option Bestehende SNS-Themen verwenden wählen, müssen Sie die AWS-Konto ID und den Namen des vorhandenen Amazon SNS SNS-Themas in dem Konto angeben, das Sie verwenden möchten. Wenn Sie mehrere AWS-Regionen anvisieren, müssen die Themennamen in jeder Region identisch sein.

5. Wählen Sie im Abschnitt Schedule aus, wie häufig Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, beheben soll. Die Standard-Option wird einmal ausgeführt. Wenn Sie nicht möchten, dass Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, wiederherstellt, wählen Sie unter Custom (Benutzerdefiniert) die Option Disable remediation (Wiederherstellung deaktivieren).
6. Wählen Sie im Abschnitt Ziele eine der folgenden Optionen aus, um die Konten und Regionen für die Aufzeichnung zu identifizieren.

> **Note**
>
> Wenn Sie in einem einzelnen Konto arbeiten, sind Optionen für die Arbeit mit Organizations und Organisationseinheiten (OUs) nicht verfügbar. Sie können wählen,

ob Sie diese Konfiguration auf alle AWS-Regionen in Ihrem Konto oder nur auf die von Ihnen ausgewählten Regionen anwenden möchten.

- Entire organization (Gesamte Organisation) – Alle Konten und Regionen in Ihrer Organisation.
- Custom (Benutzerdefiniert) – Nur die von Ihnen angegebenen Organisationseinheiten und Regionen.
  - Wählen Sie im Abschnitt Ziel-Organisationseinheiten die Organisationseinheiten aus, für die Sie die Aufzeichnung zulassen möchten.
  - Wählen Sie im Abschnitt Zielregionen die Regionen aus, in denen Sie die Aufzeichnung zulassen möchten.
- Current account (Aktuelles Konto) – Nur die Regionen, die Sie in dem Konto angeben, bei dem Sie derzeit angemeldet sind, werden als Ziel ausgewählt. Wählen Sie eine der folgenden Optionen aus:
  - Current Region (Aktuelle Region) – Nur verwaltete Knoten in der Region, die in der Konsole ausgewählt wurde, werden als Ziel ausgewählt.
  - Regionen wählen — Wählen Sie die einzelnen Regionen aus, auf die die Aufnahmekonfiguration angewendet werden soll.

7. Wählen Sie Create (Erstellen) aus.

AWS Config Bereitstellung von Konformitätspaketen

Ein Konformitätspaket ist eine Sammlung von AWS Config Regeln und Abhilfemaßnahmen. Mit Quick Setup können Sie ein Konformitätspaket als einzelne Entität in einem Konto und einer AWS-Region oder organisationsweit in einem AWS Organizations bereitstellen. Auf diese Weise können Sie mithilfe eines gemeinsamen Framework- und Paketmodells die Einhaltung der Konfiguration Ihrer AWS Ressourcen in großem Umfang verwalten, von der Richtliniendefinition über die Prüfung bis hin zur aggregierten Berichterstattung.

Voraussetzungen

Die Heimatregion für Quick Setup muss angegeben werden, bevor Sie die folgenden Aufgaben ausführen. Weitere Informationen finden Sie unter Konfigurieren der Heimat- AWS-Region.

Um Conformance Packs bereitzustellen, führen Sie die folgenden Aufgaben in der AWS Systems Manager Quick Setup Konsole aus.

> **Note**
>
> Sie müssen die AWS Config Aufzeichnung aktivieren, bevor Sie diese Konfiguration bereitstellen können. Weitere Informationen finden Sie unter Konformitätspakete im AWS Config Developer Guide.

Bereitstellen von Konformitätspaketen mit Quick Setup

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Konformitätspakete die Option Erstellen.

   > **Tip**
   >
   > Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Wählen Sie im Abschnitt Konformitätspakete auswählen die Konformitätspakete aus, die Sie einsetzen möchten.

   > **Note**
   >
   > Zusätzlich zu den AWS verwalteten Conformance Packs können Sie aus benutzerdefinierten Conformance Packs wählen, die Sie erstellt haben. Weitere Informationen finden Sie in den folgenden Themen im AWS Config Entwicklerhandbuch:
   >
   > - Benutzerdefinierte Conformance Packs
   > - Bereitstellen eines Konformitätspakets mithilfe der AWS Config -Konsole
   > - Bereitstellen eines Conformance Packs mit dem AWS Command Line Interface

5. Wählen Sie im Abschnitt Schedule aus, wie häufig Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, beheben soll. Die Standard-Option wird einmal ausgeführt. Wenn Sie nicht möchten, dass Quick Setup Änderungen an Ressourcen, die von Ihrer

Konfiguration abweichen, wiederherstellt, wählen Sie unter Custom (Benutzerdefiniert) die Option Disabled (deaktiviert).

6. Wählen Sie im Abschnitt Ziele aus, ob Sie Conformance Packs für Ihre gesamte Organisation, einige oder für das Konto bereitstellen möchten AWS-Regionen, mit dem Sie derzeit angemeldet sind.

   Fahren Sie mit Schritt 8 fort, wenn Sie Entire organization (Ganze Organisation) wählen.

   Fahren Sie mit Schritt 7 fort, wenn Sie Custom (Benutzerdefiniert) wählen.

7. Aktivieren Sie im Abschnitt Target Regions (Zielregionen) die Kontrollkästchen der Regionen, für die Sie Konformitätspakete bereitstellen möchten.

8. Wählen Sie Create (Erstellen) aus.

Patch Manager Patching-Konfiguration der Organisation

Mit Quick Setup einer Funktion von können Sie Patch-Richtlinien erstellen AWS Systems Manager, die von. Patch Manager Eine Patch-Richtlinie definiert den Zeitplan und die Baseline, die beim automatischen Patchen Ihrer Amazon Elastic Compute Cloud (Amazon EC2)-Instances und anderer verwalteter Knoten verwendet werden sollen. Mit einer einzelnen Patch-Richtlinienkonfiguration können Sie Patches für alle Konten in mehreren AWS-Regionen in Ihrer Organisation, nur für die von Ihnen ausgewählten Konten und Regionen oder für ein einzelnes Konto-Region-Paar definieren. Weitere Informationen zu Patch-Richtlinien finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien.

Voraussetzung

Um eine Patch-Richtlinie für einen Knoten mit Quick Setup zu definieren, muss es sich bei dem Knoten um einen verwalteten Knoten handeln. Weitere Informationen zum Verwalten Ihrer Knoten finden Sie unter Einrichten AWS Systems Manager.

> **Important**
>
> Scanmethoden für Patch-Konformität — Systems Manager unterstützt mehrere Methoden zum Scannen verwalteter Knoten auf Patch-Konformität. Wenn Sie mehr als eine dieser Methoden gleichzeitig implementieren, sind die angezeigten Patch-Compliance-Informationen immer das Ergebnis des letzten Scans. Ergebnisse früherer Scans werden überschrieben. Wenn die Scan-Methoden unterschiedliche Patch-Baselines mit unterschiedlichen

Genehmigungsregeln verwenden, können sich die Informationen zur Patch-Compliance unerwartet ändern. Weitere Informationen finden Sie unter Vermeiden von unbeabsichtigtem Überschreiben von Patch-Compliance-Daten.

Zuordnungs-Compliance-Status und Patch-Richtlinien — Der Patching-Status für einen verwalteten Knoten, für den eine Quick Setup Patch-Richtlinie gilt, entspricht dem Status der State Manager Zuordnungsausführung für diesen Knoten. Wenn der Status der Zuordnungsausführung lautet`Compliant`, wird der Patching-Status für den verwalteten Knoten ebenfalls markiert. `Compliant` Wenn der Ausführungsstatus der Assoziation lautet`Non-Compliant`, wird der Patching-Status für den verwalteten Knoten ebenfalls markiert. `Non-Compliant`

Unterstützte Regionen für Patch-Richtlinienkonfigurationen

Patch-Richtlinien-Konfigurationen in Quick Setup werden derzeit in den folgenden Regionen unterstützt:

- USA Ost (Ohio): (us-east-2)
- USA Ost (Nord-Virginia): (us-east-1)
- USA West (Nordkalifornien) (us-west-1)
- USA West (Oregon): (us-west-2)
- Asien-Pazifik (Mumbai): (ap-south-1)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Kanada (Zentral): (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Europa (Stockholm) (eu-north-1)
- Südamerika (São Paulo) (sa-east-1)

Berechtigungen für den S3-Bucket mit der Patch-Richtlinie

Wenn Sie eine Patch-Richtlinie erstellen, erstellt Quick Setup einen Amazon-S3 Bucket, der eine Datei mit dem Namen `baseline_overrides.json` enthält. In dieser Datei werden Informationen zu den Patch-Baselines gespeichert, die Sie für Ihre Patch-Richtlinie angegeben haben.

Der S3-Bucket-Name hat das folgende Format `aws-quicksetup-patchpolicy-`*`account-id`*`-`*`quick-setup-configuration-id`*.

Beispiel: `aws-quicksetup-patchpolicy-123456789012-abcde`

Wenn Sie eine Patch-Richtlinie für eine Organisation erstellen, wird der Bucket im Verwaltungskonto Ihrer Organisation erstellt.

Es gibt zwei Anwendungsfälle, in denen Sie anderen AWS Ressourcen die Erlaubnis erteilen müssen, mithilfe von AWS Identity and Access Management (IAM-) Richtlinien auf diesen S3-Bucket zuzugreifen:

- Fall 1: Verwenden Sie Ihr eigenes Instance-Profil oder Ihre eigene Servicerolle mit Ihren verwalteten Knoten, anstatt eines von Quick Setup bereitgestellten
- Fall 2: Verwenden Sie VPC-Endpunkte, um eine Verbindung zu Systems Manager herzustellen

Die Richtlinien für die Berechtigungen, die Sie in beiden Fällen benötigen, finden Sie im folgenden Abschnitt, Richtlinienberechtigungen für Quick Setup-S3-Buckets.

Fall 1: Verwenden Sie Ihr eigenes Instance-Profil oder Ihre eigene Servicerolle mit Ihren verwalteten Knoten, anstatt eines von Quick Setup bereitgestellten

Patch-Richtlinienkonfigurationen enthalten eine Option zum Hinzufügen erforderlicher IAM-Richtlinien zu bestehenden Instance-Profilen, die mit Ihren Instances verbunden sind.

Wenn Sie diese Option nicht wählen, aber möchten, dass Quick Setup Ihre verwalteten Knoten mit dieser Richtlinie patcht, müssen Sie sicherstellen, dass Folgendes implementiert ist:

- Die von IAM verwaltete Richtlinie `AmazonSSMManagedInstanceCore` muss an das IAM-Instance-Profil oder die IAM-Servicerolle angehängt werden, die verwendet wird, um Systems-Manager-Berechtigungen für Ihre verwalteten Knoten bereitzustellen.
- Sie müssen dem IAM-Instance-Profil oder der IAM-Servicerolle Berechtigungen für den Zugriff auf Ihren Patch-Richtlinien-Bucket als Inline-Richtlinie hinzufügen. Sie können Wildcard-Zugriff auf alle

`aws-quicksetup-patchpolicy`-Buckets oder nur auf den spezifischen Bucket gewähren, der für Ihre Organisation oder Ihr Konto erstellt wurde, wie in den früheren Codebeispielen gezeigt.

- Sie müssen Ihr IAM-Instance-Profil oder Ihre IAM-Servicerolle mit dem folgenden Schlüssel-Wert-Paar taggen.

  `Key: QSConfigId-`*`quick-setup-configuration-id`*`, Value: `*`quick-setup-configuration-id`*

  `quick-setup-configuration-id` steht für den Wert des Parameters, der auf den AWS CloudFormation Stack angewendet wird und der bei der Erstellung Ihrer Patch-Policy-Konfiguration verwendet wird. Gehen Sie wie nachfolgend beschrieben vor, um diese ID abzurufen:

  1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/cloudformation. AWS CloudFormation
  2. Wählen Sie den Namen des Stacks aus, der zur Erstellung Ihrer Patch-Richtlinie verwendet wird. Der Name hat ein Format wie `StackSet-AWS-QuickSetup-PatchPolicy-LA-q4bkg-52cd2f06-d0f9-499e-9818-d887cEXAMPLE`.
  3. Wählen Sie die Registerkarte Parameters aus.
  4. Suchen Sie in der Parameterliste in der Spalte Schlüssel nach dem Schlüssel QS ConfigurationId. Suchen Sie in der Spalte Wert für die entsprechende Zeile nach der Konfigurations-ID, z. B. `abcde`

     In diesem Beispiel lautet der Schlüssel für das Tag, das auf Ihr Instance-Profil oder Ihre Servicerolle angewendet werden soll `QSConfigId-abcde`, und der Wert lautet `abcde`.

Informationen zum Hinzufügen von Tags zu einer IAM-Rolle finden Sie unter Taggen von IAM-Rollen und Verwalten von Tags in Instanzprofilen (AWS CLI oder AWS APIs) im IAM-Benutzerhandbuch.

Fall 2: Verwenden Sie VPC-Endpunkte, um eine Verbindung zu Systems Manager herzustellen

Wenn Sie VPC-Endpunkte für die Verbindung zu Systems Manager verwenden, muss Ihre VPC-Endpunktrichtlinie für S3 den Zugriff auf Ihren S3-Bucket für Quick Setup-Patch-Richtlinien erlauben.

Informationen zum Hinzufügen von Berechtigungen zu einer VPC-Endpunkt-Richtlinie für S3 finden Sie unter Steuerung des Zugriffs von VPC-Endpunkten mit Bucket-Richtlinien im Amazon S3-Benutzerhandbuch.

Richtlinienberechtigungen für Quick Setup-S3-Buckets

Sie können Wildcard-Zugriff auf alle `aws-quicksetup-patchpolicy`-Buckets oder nur auf den speziellen Bucket gewähren, der für Ihre Organisation oder Ihr Konto erstellt wurde. Verwenden Sie eines der beiden Formate, um die erforderlichen Berechtigungen für die beiden unten beschriebenen Fälle bereitzustellen.

All patch policy buckets

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessToAllPatchPolicyRelatedBuckets",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::aws-quicksetup-patchpolicy-*"
    }
  ]
}
```

Specific patch policy bucket

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessToMyPatchPolicyRelatedBucket",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::aws-quicksetup-patchpolicy-account-id-quick-setup-
configuration-
id"[1]
    }
  ]
}
```

[1] Nachdem die Konfiguration der Patch-Richtlinie erstellt wurde, können Sie den vollständigen Namen Ihres Buckets in der S3-Konsole finden. Beispiel: `aws-quicksetup-patchpolicy-123456789012-abcde`

Zufällige Patch-Baseline-IDs bei Patch-Richtlinien-Operationen

Patching-Operationen für Patch-Richtlinien verwenden den `BaselineOverride`-Parameter im `AWS-RunPatchBaseline`-SSM-Befehlsdokument.

Wenn Sie `AWS-RunPatchBaseline` zum Patchen außerhalb einer Patch-Richtlinie verwenden, können Sie mit `BaselineOverride` eine Liste von Patch-Baselines angeben, die während des Vorgangs verwendet werden sollen und sich von den angegebenen Standardwerten unterscheiden. Sie erstellen diese Liste in einer Datei mit dem Namen `baseline_overrides.json` und fügen sie manuell zu einem Amazon-S3-Bucket hinzu, den Sie besitzen, wie in Verwenden des BaselineOverride Parameters erklärt.

Für Patching-Operationen, die auf Patch-Richtlinien basieren, erstellt Systems Manager jedoch automatisch ein S3 Bucket und fügt diesem eine `baseline_overrides.json`-Datei hinzu. Jedes Mal, wenn Quick Setup einen Patching-Vorgang durchführt (unter Verwendung der Run Command-Fähigkeit), erzeugt das System eine zufällige ID für jede Patch-Baseline. Diese ID ist für jeden Patch-Vorgang der Richtlinie unterschiedlich, und die Patch-Baseline, die sie repräsentiert, ist in Ihrem Konto weder gespeichert noch für Sie zugänglich.

Daher wird die ID der in Ihrer Konfiguration ausgewählten Patch-Baseline in den Patching-Protokollen nicht angezeigt. Dies gilt sowohl für AWS verwaltete Patch-Baselines als auch für benutzerdefinierte Patch-Baselines, die Sie möglicherweise ausgewählt haben. Die im Protokoll angegebene Baseline-ID ist stattdessen diejenige, die für diesen speziellen Patching-Vorgang erzeugt wurde.

Wenn Sie außerdem versuchen, in Patch Manager Details zu einer Patch-Baseline anzuzeigen, die mit einer zufälligen ID erstellt wurde, meldet das System, dass die Patch-Baseline nicht existiert. Dieses Verhalten ist zu erwarten und kann ignoriert werden.

Erstellen einer Patch-Richtlinie

Voraussetzungen

Die Heimatregion für Quick Setup muss angegeben werden, bevor Sie die folgenden Aufgaben ausführen. Weitere Informationen finden Sie unter Konfigurieren der Heimat- AWS-Region.

Führen Sie zum Erstellen einer Patch-Richtlinie die folgenden Aufgaben in der Systems-Manager-Konsole aus.

So erstellen Sie eine Patch-Richtlinie mit Quick Setup

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/systems-manager/ AWS Systems Manager .

   Wenn Sie das Patchen für eine Organisation einrichten, stellen Sie sicher, dass Sie beim Verwaltungskonto der Organisation angemeldet sind. Sie können die Richtlinie nicht mit dem delegierten Administratorkonto oder einem Mitgliedskonto einrichten.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Patch Manager (Patch-Manager) die Option Create (Erstellen) aus.

   > **Tip**
   >
   > Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.
4. Geben Sie für Configuration name (Konfigurationsname) einen Namen ein, um die Patch-Richtlinie zu identifizieren.
5. Wählen Sie im Abschnitt Scanning and installation (Scannen und Installation) unter Patch operation (Patching-Vorgang) aus, ob die Patch-Richtlinie die angegebenen Ziele Scan (Scannen) oder Patches auf den angegebenen Zielen Scan and install (Scannen und installieren) soll.
6. Wählen Sie unter Scanning schedule (Scan-Zeitplan) die Option Use recommended defaults (Empfohlene Standardwerte verwenden) oder Custom scan schedule (Benutzerdefinierter Scan-Zeitplan) aus. Der standardmäßige Scan-Zeitplan scannt Ihre Ziele täglich um 01:00 Uhr UTC.

   - Wenn Sie Custom scan schedule (Benutzerdefinierten Scan-Zeitplan) auswählen, wählen Sie die Scanning frequency (Scan-Frequenz) aus.
   - Wenn Sie Daily (Täglich) auswählen, geben Sie die Zeit in UTC ein, zu der Sie Ihre Ziele scannen möchten.
   - Wenn Sie Custom CRON Expression (Benutzerdefinierter CRON-Ausdruck) wählen, geben Sie den Zeitplan als CRON expression (CRON-Ausdruck) ein. Weitere Informationen zum Formatieren von CRON-Ausdrücken für Systems Manager finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Wählen Sie außerdem Wait to scan targets until first CRON interval (Mit dem Scannen von Zielen bis zum ersten CRON-Intervall warten). Standardmäßig scannt Patch Manager Knoten sofort, sobald diese zu Zielen werden.

7. Wenn Sie Scan and install (Scannen und installieren) gewählt haben, wählen Sie den Installation schedule (Installationszeitplan) aus, der beim Installieren von Patches auf den angegebenen Zielen verwendet werden soll. Wenn Sie die Option Use recommended defaults (Empfohlene Standardeinstellungen verwenden) auswählen, installiert Patch Manager die wöchentlichen Patches am Sonntag um 02:00 Uhr UTC.

   - Wenn Sie Custom install schedule (Benutzerdefinierter Installationszeitplan) auswählen, wählen Sie die Installation Frequency (Installationsfrequenz).
   - Wenn Sie Daily (Täglich) auswählen, geben Sie die Zeit in UTC ein, zu der Sie Updates auf Ihren Zielen installieren möchten.
   - Wenn Sie Custom CRON expression (Benutzerdefinierter CRON-Ausdruck) auswählen, geben Sie den Zeitplan als CRON expression (CRON-Ausdruck) ein. Weitere Informationen zum Formatieren von CRON-Ausdrücken für Systems Manager finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

     Deaktivieren Sie außerdem Wait to install updates until first CRON interval (Mit der Installation von Updates bis zum ersten CRON-Intervall warten), um Updates sofort auf Knoten zu installieren, sobald diese zu Zielen werden. Patch Manager wartet standardmäßig mit der Installation von Updates bis zum ersten CRON-Intervall.
   - Wählen Sie Reboot if needed (Bei Bedarf neu starten), um die Knoten nach der Patch-Installation neu zu starten. Ein Neustart nach der Installation wird empfohlen, kann jedoch zu Verfügbarkeitsproblemen führen.

8. Wählen Sie im Abschnitt Patch baseline (Patch-Baseline) die Patch-Baselines aus, die beim Scannen und Aktualisieren Ihrer Ziele verwendet werden sollen.

   Patch Manager verwendet standardmäßig die vordefinierten Patch-Baselines. Weitere Informationen finden Sie unter Info zu vordefinierten Baselines.

   Wenn Sie Benutzerdefinierte Patch-Baseline wählen, ändern Sie die ausgewählte Patch-Baseline für Betriebssysteme, für die Sie keine vordefinierte AWS Patch-Baseline verwenden möchten.

Die in Quick Setup verfügbaren Patch-Baselines, unabhängig davon, ob Sie AWS -vordefinierte Patch-Baselines oder benutzerdefinierte Patch-Baselines verwenden, sind die der Heimatregion, die Sie ausgewählt haben.

> **Note**
>
> Wenn Sie VPC-Endpunkte für die Verbindung zu Systems Manager verwenden, stellen Sie sicher, dass Ihre VPC-Endpunktrichtlinie für S3 den Zugriff auf diesen S3-Bucket zulässt. Weitere Informationen finden Sie unter Berechtigungen für den S3-Bucket mit der Patch-Richtlinie.

> **Important**
>
> Wenn Sie eine Patch-Richtlinienkonfiguration in Quick Setup verwenden, werden Aktualisierungen, die Sie an benutzerdefinierten Patch-Baselines vornehmen, einmal pro Stunde mit Quick Setup synchronisiert.
> Wenn eine benutzerdefinierte Patch-Baseline gelöscht wird, auf die in einer Patch-Richtlinie verwiesen wurde, wird auf der Seite mit den Quick Setup-Configuration details (Konfigurationsdetails) ein Banner für Ihre Patch-Richtlinie angezeigt. Das Banner informiert Sie darüber, dass die Patch-Richtlinie auf eine nicht mehr vorhandene Patch-Baseline verweist und nachfolgende Patching-Vorgänge fehlschlagen werden. Kehren Sie in diesem Fall zur Seite Quick Setup-Configurations (Konfigurationen) zurück, wählen Sie die Patch Manager-Konfiguration aus und wählen Sie Actions (Aktionen), Edit configuration (Konfiguration bearbeiten). Der Name der gelöschten Patch-Baseline wird hervorgehoben, und Sie müssen eine neue Patch-Baseline für das betroffene Betriebssystem auswählen.

9. (Optional) Wählen Sie im Abschnitt Patching log storage (Patching-Protokollspeicherung) die Option Write output to S3 bucket (Ausgabe in S3-Bucket schreiben) aus, um Patch-Vorgangsprotokolle in einem Amazon-S3-Bucket zu speichern.

> **Note**
>
> Wenn Sie eine Patch-Richtlinie für eine Organisation einrichten, muss das Verwaltungskonto Ihrer Organisation mindestens über schreibgeschützte Berechtigungen für diesen Bucket verfügen. Alle in der Richtlinie enthaltenen

Organisationseinheiten müssen über Schreibzugriff auf den Bucket verfügen. Informationen zum Gewähren von Bucket-Zugriff auf verschiedene Konten finden Sie unter Beispiel 2: Bucket-Besitzer, der kontoübergreifende Bucket-Berechtigungen gewährt im Benutzerhandbuch zu Amazon Simple Storage Service.

10. Wählen Sie S3 durchsuchen, um den Bucket auszuwählen, in dem Sie die Patch-Protokoll-Ausgabe speichern möchten. Das Verwaltungskonto muss über Lesezugriff auf diesen Bucket verfügen. Alle Nicht-Verwaltungskonten und Ziele, die im Abschnitt Targets (Ziele) konfiguriert sind, müssen für die Protokollierung über Schreibzugriff auf den bereitgestellten S3-Bucket verfügen.
11. Wählen Sie im Abschnitt Targets (Ziele) eine der folgenden Optionen aus, um die Konten und Regionen für diesen Patch-Richtlinienvorgang zu identifizieren.

> **Note**
>
> Wenn Sie in einem einzelnen Konto arbeiten, sind Optionen für die Arbeit mit Organizations und Organisationseinheiten (OUs) nicht verfügbar. Sie können wählen, ob Sie diese Konfiguration auf alle AWS-Regionen in Ihrem Konto oder nur auf die ausgewählten Regionen anwenden möchten.

- Entire organization (Gesamte Organisation) – Alle Konten und Regionen in Ihrer Organisation.
- Custom (Benutzerdefiniert) – Nur die von Ihnen angegebenen Organisationseinheiten und Regionen.
  - Wählen Sie im Abschnitt Target OUs (Ziel-Organisationseinheiten) die Organisationseinheiten aus, in denen Sie die Patch-Richtlinie einrichten möchten.
  - Wählen Sie im Abschnitt Target Regions (Zielregionen) die Regionen aus, in denen Sie die Patch-Richtlinie anwenden möchten.
- Current account (Aktuelles Konto) – Nur die Regionen, die Sie in dem Konto angeben, bei dem Sie derzeit angemeldet sind, werden als Ziel ausgewählt. Wählen Sie eine der folgenden Optionen aus:
  - Current Region (Aktuelle Region) – Nur verwaltete Knoten in der Region, die in der Konsole ausgewählt wurde, werden als Ziel ausgewählt.
  - Choose Regions (Regionen auswählen) – Wählen Sie die einzelnen Regionen aus, auf die die Patch-Richtlinie angewendet werden soll.

12. Wählen Sie unter Choose how you want to target instances (Wählen Sie, wie Sie Instances anvisieren möchten) eine der folgenden Möglichkeiten, um die Knoten zu identifizieren, die gepatcht werden sollen:

    - All managed nodes (Alle verwalteten Knoten) – Alle verwalteten Knoten in den ausgewählten Organisationseinheiten und Regionen.
    - Specify the resource group (Angabe der Ressourcengruppe) – Wählen Sie den Namen einer Ressourcengruppe aus der Liste, um die ihr zugeordneten Ressourcen anzuvisieren.

      > **Note**
      >
      > Derzeit wird die Auswahl von Ressourcengruppen nur für Einzelkontokonfigurationen unterstützt. Um Ressourcen in mehreren Konten zu patchen, wählen Sie eine andere Zieloption.

    - Specify a node tag (Angabe eines Knoten-Tags) – Nur Knoten, die mit dem von Ihnen angegebenen Schlüssel-Wert-Paar gekennzeichnet sind, werden in allen von Ihnen ausgewählten Konten und Regionen gepatcht.
    - Manual (Manuell) – Wählen Sie verwaltete Knoten aus allen angegebenen Konten und Regionen manuell aus einer Liste aus.

      > **Note**
      >
      > Diese Option unterstützt derzeit nur Amazon-EC2-Instances.

13. Gehen Sie im Abschnitt Rate control (Ratensteuerung) wie folgt vor:

    - Geben Sie für Concurrency (Gleichzeitigkeit) eine Anzahl oder einen Prozentsatz von Knoten ein, auf denen die Patch-Richtlinie gleichzeitig ausgeführt werden soll.
    - Geben Sie für Error threshold (Fehlerschwellenwert) die Anzahl oder den Prozentsatz der Knoten ein, bei denen ein Fehler auftreten kann, bevor die Patch-Richtlinie fehlschlägt.

14. (Optional) Aktivieren Sie das Kontrollkästchen Erforderliche IAM-Richtlinien zu bestehenden Instance-Profilen hinzufügen, die mit Ihren Instances verbunden sind.

    Diese Auswahl wendet die durch diese Quick Setup-Konfiguration erstellten IAM-Richtlinien auf Knoten an, denen bereits ein Instance-Profil (EC2 Instances) oder eine Servicerolle (hybrid-aktivierte Knoten) zugeordnet ist. Wir empfehlen diese Auswahl, wenn Ihre verwalteten

Knoten bereits über ein Instance-Profil oder eine Servicerolle verfügen, die jedoch nicht alle Berechtigungen enthalten, die für die Arbeit mit Systems Manager erforderlich sind.

Ihre Auswahl hier wird auf verwaltete Knoten angewendet, die später in den Konten und Regionen erstellt werden, für die diese Patch-Richtlinienkonfiguration gilt.

> **Important**
>
> Wenn Sie dieses Kontrollkästchen nicht aktivieren, aber möchten, dass Quick Setup Ihre verwalteten Knoten mit dieser Richtlinie patcht, müssen Sie Folgendes tun:
> Fügen Sie Ihrem IAM-Instance-Profil oder Ihrer IAM-Servicerolle Berechtigungen für den Zugriff auf den S3-Bucket hinzu, der für Ihre Patch-Richtlinie erstellt wurde
> Taggen Sie Ihr IAM-Instance-Profil oder Ihre IAM-Servicerolle mit einem bestimmten Schlüssel-Wert-Paar.
> Weitere Informationen finden Sie unter Fall 1: Verwenden Sie Ihr eigenes Instance-Profil oder Ihre eigene Servicerolle mit Ihren verwalteten Knoten, anstatt eines von Quick Setup bereitgestellten.

15. Wählen Sie Erstellen.

    Um den Patch-Status zu überprüfen, nachdem die Patch-Richtlinie erstellt wurde, können Sie über die Quick Setup-Seite auf die Konfiguration zugreifen.

DevOpsGuru-Konfiguration

Sie können DevOps Guru-Optionen schnell konfigurieren, indem Sie Quick Setup Amazon DevOps Guru ist ein auf maschinellem Lernen (ML) basierender Service, der es einfach macht, die Betriebsleistung und Verfügbarkeit einer Anwendung zu verbessern. DevOpsGuru erkennt Verhaltensweisen, die sich von normalen Betriebsmustern unterscheiden, sodass Sie Betriebsprobleme erkennen können, lange bevor sie sich auf Ihre Kunden auswirken. DevOpsGuru nimmt automatisch Betriebsdaten aus Ihren AWS Anwendungen auf und bietet ein einziges Dashboard, um Probleme in Ihren Betriebsdaten zu visualisieren. Sie können mit DevOps Guru loslegen, um die Verfügbarkeit und Zuverlässigkeit von Anwendungen zu verbessern, ohne dass Sie sich mit manueller Einrichtung oder maschinellem Lernen auskennen müssen.

Die Konfiguration von DevOps Guru mit Quick Setup ist in den folgenden Bereichen verfügbar AWS-Regionen:

- USA Ost (Nord-Virginia)
- USA Ost (Ohio)
- USA West (Oregon)
- Europe (Frankfurt)
- Europa (Irland)
- Europa (Stockholm)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)

Preisinformationen finden Sie unter Amazon DevOps Guru-Preise.

Voraussetzungen

Die Heimatregion für Quick Setup muss angegeben werden, bevor Sie die folgenden Aufgaben ausführen. Weitere Informationen finden Sie unter Konfigurieren der Heimat- AWS-Region.

Um DevOps Guru einzurichten, führen Sie die folgenden Aufgaben in der AWS Systems Manager Quick Setup Konsole aus.

Um DevOps Guru einzurichten mit Quick Setup

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der DevOps Guru-Karte die Option Erstellen aus.

> **Tip**
>
> Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Wählen Sie im Abschnitt Configuration options (Konfigurationsoptionen) die AWS -Ressourcentypen, die Sie analysieren möchten, und Ihre Benachrichtigungseinstellungen.

Wenn du die Option Alle AWS Ressourcen in allen Konten in meiner Organisation analysieren nicht auswählst, kannst du in der DevOps Guru-Konsole AWS Ressourcen auswählen, die später analysiert werden sollen. DevOpsGuru analysiert verschiedene AWS Ressourcentypen (wie Amazon Simple Storage Service (Amazon S3) -Buckets und Amazon Elastic Compute Cloud (Amazon EC2) -Instances), die in zwei Preisgruppen unterteilt sind. Sie zahlen für die analysierten AWS -Ressourcenstunden, für jede aktive Ressource. Eine Ressource ist nur aktiv, wenn sie Metriken, Ereignisse oder Protokolleinträge innerhalb einer Stunde erzeugt. Der Tarif, der Ihnen für einen bestimmten AWS Ressourcentyp berechnet wird, hängt von der Preisgruppe ab.

Wenn Sie die Option SNS-Benachrichtigungen aktivieren auswählen, wird in jeder AWS-Konto der Organisationseinheiten (OUs), auf die Sie mit Ihrer Konfiguration abzielen, ein Amazon Simple Notification Service (Amazon SNS) -Thema erstellt. DevOpsGuru verwendet das Thema, um dich über wichtige DevOps Guru-Ereignisse zu informieren, wie z. B. die Entstehung neuer Erkenntnisse. Wenn du diese Option nicht aktivierst, kannst du später in der DevOps Guru-Konsole ein Thema hinzufügen.

Wenn Sie die AWS Systems Manager OpsItems Option Aktivieren auswählen, werden operative Arbeitselemente (OpsItems) für verwandte EventBridge Amazon-Ereignisse und CloudWatch Amazon-Alarme erstellt.

5. Wählen Sie im Abschnitt Schedule aus, wie häufig Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, beheben soll. Die Standard-Option wird einmal ausgeführt. Wenn Sie nicht möchten, dass Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, wiederherstellt, wählen Sie unter Custom (Benutzerdefiniert) die Option Disabled (deaktiviert).
6. Wählen Sie im Bereich Ziele aus, ob DevOps Guru Ressourcen in einigen Ihrer Organisationseinheiten (OUs) oder in dem Konto analysieren darf, mit dem Sie gerade angemeldet sind.

   Fahren Sie mit Schritt 8 fort, wenn Sie Benutzerdefiniert wählen.

   Fahren Sie mit Schritt 9 fort, wenn Sie Custom account (Benutzerdefiniertes Konto) wählen.
7. Wählen Sie in den Abschnitten Ziel-Organisationseinheiten und Zielregionen die Kontrollkästchen der Organisationseinheiten und Regionen aus, in denen Sie DevOps Guru verwenden möchten.
8. Wählen Sie die Regionen aus, in denen Sie DevOps Guru im Girokonto verwenden möchten.

9. Wählen Sie Create (Erstellen) aus.

Distributor-Paket-Bereitstellung

Distributorist eine Fähigkeit von AWS Systems Manager. Ein Distributor-Paket ist eine Sammlung installierbarer Software oder Komponenten, das als einzelne Einheit bereitgestellt werden kann. Mit Quick Setup können Sie ein Distributor Paket in einer AWS-Konto und einer Organisation AWS-Region oder unternehmensweit in bereitstellen AWS Organizations. Derzeit können nur der EC2Launch v2-Agent, das Amazon Elastic File System (Amazon EFS) -Dienstprogrammpaket und der CloudWatch Amazon-Agent bereitgestellt werden. Quick Setup Mehr über Distributor erfahren Sie unter AWS Systems Manager Distributor.

Voraussetzungen

Die Heimatregion für Quick Setup muss angegeben werden, bevor Sie die folgenden Aufgaben ausführen. Weitere Informationen finden Sie unter Konfigurieren der Heimat- AWS-Region.

Um Distributor Pakete bereitzustellen, führen Sie die folgenden Aufgaben in der Konsole aus AWS Systems Manager Quick Setup.

Bereitstellen von Distributor-Paketen mit Quick Setup

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Distributor die Option Erstellen aus.

> **Tip**
>
> Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Wählen Sie im Abschnitt Configuration options (Konfigurationsoptionen) das Paket aus, die Sie bereitstellen wollen.
5. Wählen Sie im Abschnitt Targets (Ziele) aus, ob Sie das Paket für Ihre gesamte Organsiation, einige Ihrer Organisationseinheiten (OUs) oder das Konto, bei dem Sie angemeldet sind, bereitstellen möchten.

Fahren Sie mit Schritt 8 fort, wenn Sie Entire organization (Ganze Organisation) wählen.

Fahren Sie mit Schritt 7 fort, wenn Sie Custom (Benutzerdefiniert) wählen.

6. Aktivieren Sie im Abschnitt Target OUs die Kontrollkästchen der OUs (Organisationseinheiten) und Regionen, für die Sie das Paket bereitstellen möchten.
7. Wählen Sie Create (Erstellen) aus.

Amazon-EC2-Instance-Resource-Scheduler

Mit Quick Setup einer Funktion von können Sie Resource Scheduler so konfigurieren AWS Systems Manager, dass das Starten und Stoppen von Amazon Elastic Compute Cloud (Amazon EC2) - Instances automatisiert wird.

Diese Quick Setup-Konfiguration unterstützt Sie bei der Senkung der Betriebskosten, indem Instances gemäß dem von Ihnen festgelegten Zeitplan gestartet und beendet werden. Diese Funktion hilft Ihnen, unnötige Kosten für die Ausführung von Instances zu vermeiden, wenn diese nicht benötigt werden. So kann es beispielsweise sein, dass Sie Ihre Instances ständig ausführen lassen, obwohl sie nur 10 Stunden pro Tag und 5 Tage pro Woche verwendet werden. Stattdessen können Sie Ihre Instances so planen, dass sie jeden Tag nach den Geschäftszeiten beendet werden. Das Ergebnis wäre eine Einsparung von 70 Prozent für diese Instances, da die Ausführung von 168 Stunden auf 50 Stunden reduziert wird. Bei der Nutzung des Quick Setup-Service fallen keine Kosten an. Kosten können jedoch durch die von Ihnen eingerichteten Ressourcen und die Nutzungsbeschränkungen entstehen, wobei keine Gebühren für die Dienste anfallen, die zur Einrichtung Ihrer Konfiguration verwendet werden.

Mit Resource Scheduler können Sie festlegen, dass Instances automatisch über mehrere Instanzen hinweg und AWS-Konten nach einem von Ihnen definierten Zeitplan gestoppt AWS-Regionen und gestartet werden. Die Quick Setup-Konfiguration verwendet den von Ihnen angegebenen Tag-Schlüssel und Wert für Amazon-EC2-Instances. Nur die Instances mit einem Tag, das mit dem Wert übereinstimmt, den Sie in Ihrer Konfiguration angeben, werden vom Resource Scheduler beendet oder gestartet.

Eine individuelle Konfiguration unterstützt die Zeitplanung von bis zu 5 000 Instances pro Region. Wenn in Ihrem Fall mehr als 5 000 Instances in einer bestimmten Region geplant werden müssen, müssen Sie mehrere Konfigurationen erstellen. Kennzeichnen Sie Ihre Instances entsprechend, damit jede Konfiguration bis zu 5 000 Instances verwalten kann. Beim Erstellen mehrerer Quick Setup-Konfigurationen für Resource Scheduler müssen Sie verschiedene Tag-Schlüsselwerte

angeben. Beispielsweise kann eine Konfiguration den Tag-Schlüssel „Env“ mit dem Wert „Prod“ verwenden, während eine andere „Env“ und „Dev“ verwendet.

Wenn Sie Ihre Konfiguration löschen, werden Instances nicht mehr gemäß dem zuvor definierten Zeitplan beendet und gestartet. In seltenen Fällen werden Instances aufgrund von API-Operationsfehlern möglicherweise nicht erfolgreich beendet oder gestartet.

Resource Scheduler startet die gekennzeichneten Instances nur, wenn sich diese im `stopped`-Status befinden. Ebenso werden Instances nur dann beendet, wenn sie sich im `running`-Status befinden. Resource Scheduler arbeitet nach einem ereignisgesteuerten Modell und startet oder beendet Instances nur zu den von Ihnen festgelegten Zeiten. Sie erstellen beispielsweise einen Zeitplan, der Instances um 9:00 Uhr startet. Resource Scheduler startet alle Instances, die dem von Ihnen angegebenen Tag zugeordnet sind und sich im `stopped`-Status befinden, um 09:00 Uhr. Wenn die Instances zu einem späteren Zeitpunkt manuell angehalten werden, startet Resource Scheduler diese nicht erneut, um den `running`-Status beizubehalten. Wenn eine Instance manuell gestartet wird, nachdem sie gemäß Ihrem Zeitplan angehalten wurde, wird Resource Scheduler die Instance nicht erneut anhalten.

Wenn Sie einen Zeitplan mit einer Startzeit erstellen, die nach der Anhaltezeit liegt, geht Resource Scheduler davon aus, dass Ihre Instances über Nacht ausgeführt werden. Sie erstellen beispielsweise einen Zeitplan, der Instances um 21:00 Uhr startet und Instances um 07:00 Uhr beendet. Resource Scheduler startet alle Instances, die dem von Ihnen angegebenen Tag zugeordnet sind und sich im `stopped`-Status befinden, um 21:00 Uhr und beendet sie um 07:00 Uhr am nächsten Tag. Bei Nachtplänen gilt die Startzeit für die Tage, die Sie für Ihren Zeitplan auswählen. Die Anhaltezeit gilt jedoch für den folgenden Tag in Ihrem Zeitplan.

Voraussetzungen

Die Heimatregion für Quick Setup muss angegeben werden, bevor Sie die folgenden Aufgaben ausführen. Weitere Informationen finden Sie unter Konfigurieren der Heimat- AWS-Region.

Um die Planung für Amazon EC2 EC2-Instances einzurichten, führen Sie die folgenden Aufgaben in der AWS Systems Manager Quick Setup Konsole aus.

So richten Sie die Instance-Zeitplanung mit Quick Setup ein

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.

3. Wählen Sie auf der Karte Resource Scheduler die Option Erstellen aus.

> **Tip**
>
> Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Geben Sie im Abschnitt Instance tag (Instance-Tag) den Tag-Schlüssel und den Wert für die Instances an, die Sie Ihrem Zeitplan zuordnen möchten.
5. Geben Sie im Abschnitt Schedule options (Zeitplanoptionen) die Zeitzone, die Tage und die Uhrzeiten an, zu denen Sie Ihre Instances starten und beenden möchten.
6. Wählen Sie im Abschnitt Targets (Ziele) aus, ob Sie die Zeitplanung für eine Custom (Benutzerdefinierte) Gruppe von Organisationseinheiten (OUs) oder das Current account (Aktuelle Konto), bei dem Sie angemeldet sind, einrichten möchten:
   - Custom (Benutzerdefiniert) – Wählen Sie im Abschnitt Target OUs (Ziel-Organisationseinheiten) die Organisationseinheiten aus, in denen Sie die Zeitplanung einrichten möchten. Wählen Sie als Nächstes im Abschnitt Target Regions (Zielregionen) die Regionen aus, in denen Sie die Zeitplanung einrichten möchten.
   - Current account (aktuelles Konto)— Wählen Sie Current Region (aktuelle Region) oder Choose Regions (Regionen wählen). Wenn Sie Choose Regions (Regionen auswählen) ausgewählt haben, wählen Sie die Target Regions (Zielregionen) aus, in denen Sie die Zeitplanung einrichten möchten.
7. Überprüfen Sie die Informationen zum Zeitplan im Abschnitt Summary (Zusammenfassung).
8. Wählen Sie Create (Erstellen) aus.

AWS Ressourcen Explorer Konfiguration

Mit Quick Setup der Funktion von können Sie schnell konfigurieren AWS Systems Manager, AWS Ressourcen Explorer um Ressourcen in Ihrer Organisation AWS-Konto oder in einer gesamten AWS Organisation zu suchen und zu finden. Sie können mithilfe von Metadaten wie Namen, Tags und IDs nach Ihren Ressourcen suchen. AWS Ressourcen Explorer bietet mithilfe von Indizes schnelle Antworten auf Ihre Suchanfragen. Resource Explorer erstellt und verwaltet Indizes mithilfe einer Vielzahl von Datenquellen, um Informationen über Ressourcen in Ihrem zu sammeln. AWS-Konto

Quick Setupfür Resource Explorer automatisiert den Indexkonfigurationsprozess. Weitere Informationen zu finden Sie AWS Ressourcen Explorer unter Was ist AWS Ressourcen Explorer? im AWS Ressourcen Explorer Benutzerhandbuch.

Quick SetupWährenddessen macht Resource Explorer Folgendes:

- Erstellt AWS-Region in jedem von Ihnen einen Index AWS-Konto.
- Aktualisiert den Index in der Region, die Sie als Aggregatorindex für das Konto angeben.
- Erstellt eine Standardansicht in der Aggregator-Index-Region. Diese Ansicht hat keine Filter und gibt daher alle im Index gefundenen Ressourcen zurück.

Mindestberechtigungen

Um die Schritte im folgenden Verfahren ausführen zu können, benötigen Sie die folgenden Berechtigungen:

- Aktion: `resource-explorer-2:*` — Ressource: keine spezifische Ressource (`*`)
- Aktion: `iam:CreateServiceLinkedRole` — Ressource: keine spezifische Ressource (`*`)

Um Resource Explorer zu konfigurieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie eine Heimatregion und dann Erste Schritte.
4. Wählen Sie auf der Resource Explorer-Karte die Option Erstellen aus.
5. Wählen Sie im Abschnitt Aggregator-Index-Region aus, welche Region der Aggregatorindex enthalten soll. Sie sollten die Region auswählen, die für den geografischen Standort Ihrer Benutzer geeignet ist.
6. (Optional) Aktivieren Sie das Kontrollkästchen Vorhandene Aggregatorindizes in anderen als den oben ausgewählten Regionen ersetzen.
7. Wählen Sie im Abschnitt Ziele die Zielorganisation oder bestimmte Organisationseinheiten (OUs) aus, die die Ressourcen enthalten, die Sie ermitteln möchten.
8. Wählen Sie im Abschnitt Regionen aus, welche Regionen in die Konfiguration aufgenommen werden sollen.

9. Sehen Sie sich die Zusammenfassung der Konfiguration an und wählen Sie dann Create aus.

Auf der Resource Explorer-Seite können Sie den Konfigurationsstatus überwachen.

Fehlerbehebung von Quick Setup-Ergebnissen

Fehlgeschlagene Bereitstellung

Eine Bereitstellung schlägt fehl, wenn der Stack-Set von CloudFormation während der Erstellung fehlgeschlagen ist. Gehen Sie wie folgt vor, um einen Bereitstellungsfehler zu untersuchen.

1. Navigieren Sie zur AWS CloudFormation-Konsole.
2. Wählen Sie den Stack aus, der von der Quick Setup-Konfiguration erstellt wurde. Der Stack name (Stack-Name) beinhaltet `QuickSetup`, gefolgt von der Art der ausgewählten Konfiguration, wie etwa `SSMHostMgmt`.

   > **Note**
   >
   > Manchmal löscht CloudFormation fehlgeschlagene Stack-Bereitstellungen. Wenn der Stack in der Tabelle Stacks nicht verfügbar ist, wählen Sie Deleted (Gelöscht) in der Filterliste aus.

3. Zeigen Sie den Status und den Status reason (Statusgrund) an. Weitere Informationen zum Stack-Status finden Sie unter Stack-Statuscodes im Benutzerhandbuch von AWS CloudFormation.
4. Um nachzuvollziehen, welcher Schritt genau fehlgeschlagen ist, sehen Sie sich auf der Registerkarte Events (Ereignisse) den Status der einzelnen Ereignisse an.
5. Lesen Sie den Abschnitt Fehlerbehebung im Benutzerhandbuch von AWS CloudFormation.
6. Wenn sich der Bereitstellungsfehler mit den CloudFormation-Schritten zur Fehlerbehebung nicht beheben lässt, löschen Sie die Konfiguration und erstellen Sie sie neu.

Fehlgeschlagene Zuordnung

Die Tabelle Configuration details (Konfigurationsdetails) auf der Seite Configuration details Ihrer Konfiguration zeigt als Configuration status (Konfigurationsstatus) Failed (Fehlgeschlagen) an, wenn eine der Zuordnungen bei der Einrichtung fehlgeschlagen ist. Gehen Sie zur Fehlerbehebung einer fehlgeschlagenen Zuordnung wie folgt vor.

1. Wählen Sie in der Tabelle Configuration details (Konfigurationsdetails) die fehlgeschlagene Konfiguration und dann View Details (Details anzeigen) aus.
2. Kopieren Sie den Association name (Zuordnungsnamen).
3. Navigieren Sie zu State Manager und fügen Sie den Zuordnungsnamen in das Suchfeld ein.
4. Wählen Sie die Zuordnung und dann die Registerkarte Execution history (Ausführungsverlauf) aus.
5. Wählen Sie unter Execution ID (Ausführungs-ID) die Zuordnungsausführung aus, die fehlgeschlagen ist.
6. Auf der Seite Association execution targets (Zuordnungs-Ausführungsziele) werden alle Knoten aufgelistet, auf denen die Zuordnung ausgeführt wurde. Wählen Sie die Schaltfläche Output (Ausgabe) für eine fehlgeschlagene Ausführung aus.
7. Wählen Sie auf der Seite Output (Ausgabe) Step – Output (Schritt – Ausgabe) aus, um die Fehlermeldung für diesen Schritt in der Befehlsausführung anzuzeigen. Jeder Schritt kann eine andere Fehlermeldung anzeigen. Überprüfen Sie die Fehlermeldungen für alle Schritte, um das Problem zu beheben.

Wenn sich das Problem durch die Anzeige der Schrittausgabe nicht beheben lässt, können Sie versuchen, die Zuordnung neu zu erstellen. Um die Zuordnung neu zu erstellen, löschen Sie zunächst die fehlgeschlagene Zuordnung in State Manager. Bearbeiten Sie anschließend die Konfiguration, wählen Sie die von Ihnen gelöschte Option aus und klicken Sie auf Update (Aktualisieren).

> **Note**
>
> Um fehlgeschlagene Zuordnungen für die Konfiguration einer Organisation zu untersuchen, müssen Sie sich bei dem Konto mit der fehlgeschlagenen Zuordnung anmelden und das zuvor beschriebene Verfahren für fehlgeschlagene Zuordnungen anwenden. Die Association ID (Zuordnungs-ID) ist kein Hyperlink zum Zielkonto beim Anzeigen von Ergebnissen vom Verwaltungskonto.

Abweichungsstatus

Auf der Detailseite einer Konfiguration können Sie den Abweichungsstatus der einzelnen Bereitstellungen anzeigen. Eine Konfigurationsabweichung tritt auf, wenn Benutzer Änderungen an einem Service oder einer Funktion vornehmen, die mit der Auswahl in Quick Setup im Konflikt

stehen. Wenn sich eine Zuordnung nach der Erstkonfiguration geändert hat, zeigt die Tabelle ein Warnsymbol, das die Anzahl der abweichenden Elemente angibt. Sie können die Ursache der Abweichung feststellen, indem Sie den Mauszeiger über das Symbol bewegen.

Wenn eine Zuordnung in State Manager gelöscht wird, zeigen die zugehörigen Bereitstellungen eine Abweichungswarnung an. Bearbeiten Sie zur Behebung dieses Problems die Konfiguration und wählen Sie die Option aus, die beim Löschen der Zuordnung entfernt wurde. Wählen Sie Update (Aktualisieren) aus und warten Sie, bis die Bereitstellung abgeschlossen ist.

Verfahrensmanagement

Operations Management besteht aus einer Reihe von Funktionen, die Sie bei der Verwaltung Ihrer AWS -Ressourcen unterstützen.

Themen

- AWS Systems Manager Incident Manager
- AWS Systems Manager Explorer
- AWS Systems Manager OpsCenter
- Von Systems Manager gehostete CloudWatch Amazon-Dashboards

AWS Systems Manager Incident Manager

Verwenden Sie Incident Manager, eine Funktion von AWS Systems Manager, um Vorfälle zu verwalten, die in Ihren AWS gehosteten Anwendungen auftreten. Incident Manager kombiniert Benutzereingriffe, Eskalation, Runbooks, Reaktionspläne, Chat-Kanäle und Analysen nach Vorfällen, damit Ihr Team Vorfälle schneller einordnen und Ihre Anwendungen wieder in den Normalzustand versetzen kann. Weitere Informationen zu Incident Manager finden Sie unter Incident Manager-Benutzerhandbuch.

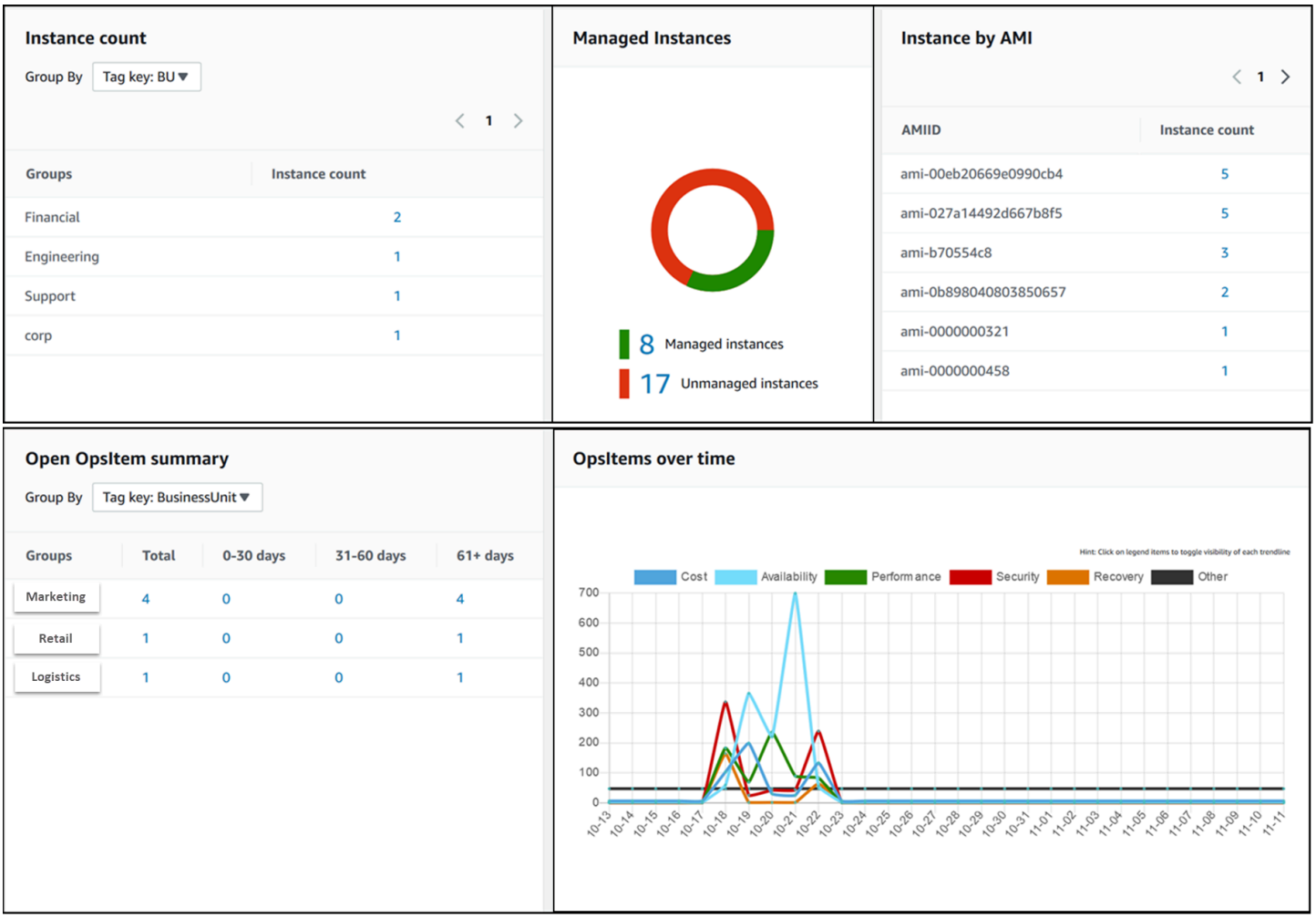
AWS Systems Manager Explorer

AWS Systems Manager-Explorer ist ein anpassbares Betriebs-Dashboard, in dem Informationen zu Ihren AWS-Ressourcen aufgeführt werden. Explorer zeigt eine Gesamtansicht der Betriebsdaten (OpsData) für Ihre AWS-Konten und für alle AWS-Regionen an. In Explorer enthält OpsData Metadaten über die verwalteten Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung. OpsData enthält auch Informationen, die von anderen Systems-Manager-Funktionen bereitgestellt werden, einschließlich Details zur Patch Manager-Patch-Compliance und zur State Manager-Zuordnungs-Compliance. Um den Zugriff auf OpsData weiter zu vereinfachen, zeigt Explorer Informationen von unterstützenden AWS-Services wie AWS Config, AWS Trusted Advisor, AWS Compute Optimizer und AWS Support (Supportfälle) an.

Um die betriebliche Sensibilisierung zu erhöhen zeigt Explorer auch operative Arbeitselemente an (OpsItems). Explorer bietet Kontext darüber, wie OpsItems über Ihre Geschäftsbereiche oder Anwendungen verteilt sind, welche Trends im Zeitverlauf bestehen und wie sie je nach Kategorie

variieren. Sie können Informationen in Explorer gruppieren und filtern, um sich auf die Elemente zu konzentrieren, die für Sie relevant sind und eine Aktion erfordern. Wenn Sie Probleme mit hoher Priorität erkennen, können Sie mit Systems Manager OpsCenter Automation-Runbooks ausführen und die Probleme schnell beheben. Um mit Explorer zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Explorer aus.

Die folgende Abbildung zeigt einige der einzelnen Berichtsfelder, Widgets, genannt, die in Explorer verfügbar sind.



Über welche Features verfügt Explorer?

Explorer umfasst die folgenden Features:

- Anpassbare Anzeige von verwertbaren Informationen: Explorer enthält Drag-and-Drop-Widgets, die automatisch verwertbare Informationen über Ihre AWS-Ressourcen anzeigen. Explorer zeigt Informationen in zwei Arten von Widgets an.

  - Informative Widgets: Diese Widgets fassen Daten aus Amazon EC2, Patch Manager ,State Manager und unterstützenden AWS-Services wie AWS Trusted Advisor, AWS-Compute Optimizer und AWS Support zusammen. Diese Widgets bieten einen wichtigen Kontext, mit dem Sie die Zustands- und Betriebsrisiken Ihrer AWS-Ressourcen verstehen können. Beispiele für Informations-Widgets sind: Instance count (Instance-Anzahl), Instance by AMI (Instance nach AMI), Total noncompliant nodes (Gesamtzahl nicht konformer Knoten) (Patching), Noncompliant associations (Nicht konforme Zuordnungen) und Support Center cases (Support-Center-Fälle).
  - OpsItem-Widgets: Ein Systems Manager-OpsItem ist ein operatives Arbeitselement, das mit mindestens einer AWS-Ressource verknüpft ist. OpsItems sind ein Feature von Systems Manager OpsCenter. Für OpsItems kann erforderlich sein, dass DevOps-Techniker ein Problem untersuchen und möglicherweise beheben müssen. Zu den Beispielen möglicher OpsItems gehören hohe CPU-Auslastung der EC2-Instance, getrennte Amazon Elastic Block Store (Amazon EBS)-Volumes, AWS CodeDeploy-Bereitstellungsfehler oder Systems Manager-Automation-Ausführungsfehler. Zu den Beispielen für OpsItem-Widgets gehören Open OpsItem summary (OpsItem-Zusammenfassung öffnen), OpsItem by status (nach Status) und OpsItems over time (im Laufe der Zeit).
- Filters: Jedes Widget bietet die Möglichkeit, Informationen basierend auf AWS-Konto, AWS-Region und Tag zu filtern. Mithilfe von Filtern können Sie die in Explorer angezeigten Informationen schnell verfeinern.
- Direktlinks zu Service-Bildschirmen: Um Probleme mit AWS-Ressourcen zu untersuchen, enthalten Explorer-Widgets Direktlinks zu verwandten Service-Bildschirmen. Filter, die auf ein Widget angewendet werden, bleiben wirksam, wenn Sie zu einem zugehörigen Service-Bildschirm navigieren.
- Gruppen: Mit einigen Widgets können Sie Daten basierend auf Konto, Region und Tag gruppieren, um die Arten von betrieblichen Problemen in Ihrer Organisation zu verstehen.
- Berichts-Tag-Schlüssel: Wenn Sie Explorer einrichten, können Sie bis zu fünf Tag-Schlüssel angeben. Diese Schlüssel helfen Ihnen, Daten in Explorer zu gruppieren und zu filtern. Wenn ein angegebener Schlüssel mit einem Schlüssel auf einer Ressource übereinstimmt, die ein OpsItem generiert, sind der Schlüssel und der Wert in OpsItems enthalten.
- Drei Modi der AWS-Konto- und AWS-Region-Anzeige: Explorer enthält die folgenden Anzeigemodi für OpsData und OpsItems in AWS-Konten und AWS-Regionen:
  - Einzelkonto/Einzelregion: Dies ist die Standardansicht. In diesem Modus können Benutzer Daten und OpsItems von ihrem eigenen Konto und der aktuellen Region anzeigen.
  - Einzelkonto/Mehrfachregion: In diesem Modus müssen Sie mithilfe der Seite Explorer-Settings (Einstellungen) mindestens eine Ressourcen-Datensynchronisierung erstellen. Eine Ressourcen-

Datensynchronisierung aggregiert OpsData aus mindestens einer Region. Nachdem Sie eine Ressourcen-Datensynchronisierung erstellt haben, können Sie umschalten, welche Synchronisierung im Explorer-Dashboard verwendet werden soll. Anschließend können Sie Daten basierend auf der Region filtern und gruppieren.

  - Mehrfachkonto/Mehrfachregion: Dieser Modus erfordert, dass Ihre Organisation oder Ihr Unternehmen AWS Organizations mit Alle Features eingeschaltet verwendet. Nachdem Sie AWS Organizations in Ihrer Computerumgebung konfiguriert haben, können Sie alle Kontodaten in einem Verwaltungskonto aggregieren. Anschließend können Sie die Ressourcen-Datensynchronisierungen erstellen, damit Sie die Daten basierend auf der Region filtern und gruppieren können. Weitere Informationen über den Modus Alle Features in Organisationen finden Sie unter Aktivieren aller Features in Ihrer Organisation.

- Berichterstellung: Sie können Explorer-Berichte als CSV-Dateien in einen Amazon Simple Storage Service (Amazon S3)-Bucket exportieren. Wenn ein Export abgeschlossen ist, erhalten Sie eine Benachrichtigung von Amazon Simple Notification Service (Amazon SNS).

In welcher Verbindung steht Explorer mit OpsCenter?

Systems Manager OpsCenter bietet einen zentralen Standort, an dem Techniker und IT-Experten OpsItems im Zusammenhang mit AWS-Ressourcen anzeigen, untersuchen und Fehler beheben können. Explorer ist ein Berichts-Hub, in dem DevOps-Manager aggregierte Zusammenfassungen ihrer Betriebsdaten, einschließlich OpsItems, über alle AWS-Regionen und -Konten hinweg anzeigen. Explorer hilft Benutzern, Trends und Muster zu erkennen und bei Bedarf Probleme mit Systems Manager-Automation-Runbooks schnell zu lösen.

Das OpsCenter-Setup ist jetzt in das Explorer-Setup integriert. Wenn Sie OpsCenter bereits eingerichtet haben,zeigt Explorer automatisch Betriebsdaten an, einschließlich aggregierter Informationen über OpsItems. Wenn Sie OpsCenter nicht eingerichtet haben, können Sie das Explorer-Setup verwenden, um mit beiden Funktionen zu beginnen. Weitere Informationen finden Sie unter Erste Schritte mit Systems Manager Explorer und OpsCenter.

Was ist OpsData?

OpsData bezeichnet alle Betriebsdaten, die im Systems-Manager-Explorer-Dashboard angezeigt werden. Explorer ruft OpsData aus den folgenden Quellen ab:

- Amazon Elastic Compute Cloud (Amazon EC2)

Die in Explorer angezeigten Daten umfassen die Gesamtanzahl der Knoten, die Gesamtanzahl der verwalteten und nicht verwalteten Knoten und die Anzahl von Knoten, die ein spezifisches Amazon Machine Image (AMI) verwenden.

- Systems Manager OpsCenter

  Die in Explorer angezeigten Daten umfassen die Anzahl von OpsItems nach Status, die Anzahl von OpsItems nach Schweregrad, die Anzahl der offenen OpsItems in allen Gruppen und für Zeiträume von 30 Tagen sowie historische Daten zu OpsItems im Zeitverlauf.

- Systems Manager Patch Manager

  Die in Explorer angezeigten Daten enthalten die Anzahl der nicht konformen und kritischen nicht konformen Knoten.

- AWS Trusted Advisor

  Die in Explorer angezeigten Daten enthalten: Status bewährter Prüfungsmethoden für EC2 Reserved Instances in den Bereichen Kostenoptimierung, Sicherheit, Fehlertoleranz, Leistung und Servicelimits.

- AWS Compute Optimizer

  In Explorer werden die folgenden Daten angezeigt: die Zahl der EC2-Instances, die Under provisioned (Zu wenig bereitgestellt) und Over provisioned (Zu viel bereitgestellt) wurden, Optimierungsergebnisse, Details zu On-Demand-Preisen und Empfehlungen für Instance-Typ und Preis.

- AWS Support zentrale Fälle

  Angezeigte Daten in Explorer umfassen: Fall-ID, Schweregrad, Status, Erstellungszeit, Betreff, Service und Kategorie.

- AWS Config

  Angezeigte Daten in Explorer umfassen: Zusammenfassung der konformen und nicht konformen AWS Config-Regeln, die Anzahl der konformen und nicht konformen Ressourcen und spezifische Details zu den einzelnen Ressourcen (wenn Sie eine nicht konforme Regel oder Ressource aufschlüsseln).

- AWS Security Hub

  Angezeigte Daten in Explorer umfassen: Gesamtübersicht der Ergebnisse des Security Hub, die Anzahl jeder Suche gruppiert nach Schweregrad und spezifische Details zur Suche.

> **Note**
>
> Um AWS Trusted Advisor und AWS Support Center-Fälle in Explorer anzeigen zu können, müssen Sie entweder ein Enterprise- oder ein Business-Konto bei AWS Support eingerichtet haben.

Sie können OpsData-Quellen auf der Explorer-Seite Explorer Settings anzeigen und verwalten. Weitere Informationen zum Einrichten und Konfigurieren von Services, die Explorer-Widgets mit OpsData füllen, finden Sie unter Einrichten von zugehörigen Services.

Entstehen Kosten für die Verwendung von Explorer?

Ja. Wenn Sie die Standardregeln für die Erstellung von OpsItems während des integrierten Setups aktivieren, starten Sie einen Prozess, der OpsItems automatisch erstellt. Ihr Konto wird basierend auf der Anzahl der pro Monat erstellten OpsItems belastet. Ihr Konto wird außerdem basierend auf der Anzahl der pro Monat getätigten `GetOpsItem`-, `DescribeOpsItem`-, `UpdateOpsItem`- und `GetOpsSummary`-API-Aufrufe belastet. Darüber hinaus können öffentliche API-Aufrufe an andere Services, die relevante Diagnoseinformationen bereitstellen, in Rechnung gestellt werden. Weitere Informationen finden Sie unter AWS Systems Manager- Preise.

Themen

- Erste Schritte mit Systems Manager Explorer und OpsCenter
- Verwenden von Systems Manager Explorer
- OpsData Aus Systems Manager exportieren Explorer
- Fehlerbehebung von Systems Manager Explorer

Erste Schritte mit Systems Manager Explorer und OpsCenter

AWS Systems Manager verwendet eine integrierte Einrichtungsumgebung, um Ihnen den Einstieg in Systems Manager Explorer und Systems Manager OpsCenter zu erleichtern. In dieser Dokumentation wird das Setup von Explorer und OpsCenter als Integriertes Setup bezeichnet. Wenn Sie OpsCenter bereits eingerichtet haben, müssen Sie das integrierte Setup trotzdem ausführen, um die Einstellungen und Optionen zu überprüfen. Wenn Sie OpsCenter nicht eingerichtet haben, können Sie mit dem integrierten Setup beide Funktionen einrichten.

> **Note**
>
> Integriertes Setup ist nur in der Systems Manager-Konsole verfügbar. Sie können Explorer oder OpsCenter nicht programmatisch einrichten.

Das integrierte Setup führt die folgenden Aufgaben aus:

- Konfiguriert Rollen und Berechtigungen: Das integrierte Setup erstellt eine AWS Identity and Access Management (IAM)-Rolle, die es Amazon EventBridge ermöglicht, OpsItems automatisch basierend auf Standardregeln zu erstellen. Nach dem Einrichten müssen Sie Benutzer-, Gruppen- oder Rollenberechtigungen für OpsCenter konfigurieren, wie in diesem Abschnitt beschrieben.
- Aktiviert Standardregeln für die OpsItem-Erstellung: Das integrierte Setup erstellt Standardregeln in EventBridge. Diese Regeln erstellen OpsItems automatisch als Reaktion auf Ereignisse. Beispiele für diese Ereignisse sind: Statusänderung für eine AWS-Ressource, Änderung der Sicherheitseinstellungen oder ein Service, der nicht mehr verfügbar ist.
- Lässt OpsData-Quellen zu: Das integrierte Setup ermöglicht Datenquellen, die Explorer-Widgets ausfüllen.
- Erlaubt die Angabe von Berichts-Tag-Schlüsseln: Mit dem integrierten Setup können Sie bis zu fünf Berichts-Tag-Schlüssel angeben, die automatisch neuen OpsItems zugewiesen werden sollen, die bestimmte Kriterien erfüllen.

Nach dem Abschluss des integrierten Setups empfehlen wir, dass Sie Set up Explorer to display data from multiple Regions and accounts (einrichten, um Daten aus mehreren Regionen und Konten anzuzeigen). Explorer und OpsCenter synchronisieren OpsData und OpsItems automatisch für das AWS-Konto und die AWS-Region, die Sie verwendet haben, als Sie das integrierte Setup abgeschlossen haben. Sie können OpsData und OpsItems aus anderen Konten und Regionen aggregieren, indem Sie eine Ressourcendatensynchronisierung erstellen.

> **Note**
>
> Sie können die Setup-Konfigurationen jederzeit auf der Seite Settings (Einstellungen) ändern.

Einrichten von zugehörigen Services

AWS Systems Manager-Explorer und AWS Systems Manager OpsCenter sammeln Informationen oder interagieren mit anderen AWS-Services und Systems-Manager-Funktionen. Wir empfehlen, diese anderen Services oder Funktionen einzurichten und zu konfigurieren, bevor Sie das integrierte Setup verwenden.

Die folgende Tabelle enthält Aufgaben, mit denen Explorer und OpsCenter ermöglicht wird, Informationen aus anderen AWS-Services und Systems-Manager-Funktionen zu sammeln und mit diesen zu interagieren.

Aufgabe	Informationen
Überprüfen der Berechtigungen in Systems Manager Automation	Explorer und OpsCenter ermöglichen es Ihnen, Probleme mit AWS-Ressourcen mithilfe von Systems Manager Automation-Runbooks zu beheben. Zur Verwendung dieser Abhilfemaßnahme, benötigen Sie die Berechtigung zur Ausführung von Systems Manager Automation-Runbooks. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
Einrichten und Konfigurieren von Systems Manager Patch Manager	Explorer enthält ein Widget, das Informationen zu Patch Compliance bereitstellt. Um diese Daten in Explorer anzuzeigen, müssen Sie das Patching konfigurieren. Weitere Informationen finden Sie unter AWS Systems Manager Patch Manager.
Einrichten und Konfigurieren von Systems Manager State Manager	Explorer enthält ein Widget, das Informationen zu Systems Manager State Manager-Association-Compliance bereitstellt. Um diese Daten in Explorer anzuzeigen, müssen Sie State Manager konfigurieren. Weitere Informationen finden Sie unter AWS Systems Manager State Manager.

Aufgabe	Informationen
Aktivieren Sie AWS Config-Configuration Recorder	Explorer verwendet Daten, die von AWS Config Configuration Recorder bereitgestellt werden, um Widgets mit Informationen zu Ihren EC2-Instances zu füllen. Um diese Daten in Explorer anzuzeigen, aktivieren Sie AWS Config-Co nfiguration Recorder. Weitere Informationen finden Sie unter Verwalten von Configuration Recorder. **Note** Nachdem Sie Configuration Recorder aktiviert haben, kann Systems Manager bis zu sechs Stunden benötigen, bis Daten in Explorer-Widgets angezeigt werden, die Informationen zu Ihren EC2-Instances anzeigen.
AWS Trusted Advisor aktivieren	Explorer verwendet von Trusted Advisor bereitgestellte Daten, um den Status bewährter Prüfungsmethoden für Amazon EC2 Reserved Instances in den Bereichen Kostenoptimierung, Sicherheit, Fehlertoleranz, Leistung und Servicelimits anzuzeigen. Um diese Daten in Explorer anzuzeigen, benötigen Sie einen Business- oder Enterprise-Supportplan. Weitere Informationen finden Sie unter AWS Support.

Aufgabe	Informationen
AWS Compute Optimizer aktivieren	Explorer verwendet von Compute Optimizer bereitgestellte Daten zur Anzeige von Details, darunter die Zahl der EC2-Instances, die Under provisioned (Zu wenig bereitgestellt) und Over provisioned (Zu viel bereitgestellt) wurden, Optimierungsergebnisse, Details zu On-Demand-Preisen und Empfehlungen für Instance-Typ und Preis. Aktivieren Sie Compute Optimizer zum Anzeigen von Daten in Explorer. Weitere Informationen finden Sie unter Erste Schritte mit AWS Compute Optimizer.
AWS Security Hub aktivieren	Explorer verwendet Daten, die von Security Hub bereitgestellt werden, um Widgets mit Informationen zu Ihren Sicherheitsergebnissen zu füllen. Um diese Daten in Explorer anzuzeigen, aktivieren Sie die Integration des Security Hub. Weitere Informationen finden Sie unter Was ist AWS Security Hub.

Konfigurieren von Rollen und Berechtigungen für Systems Manager Explorer

Das integrierte Setup erstellt und konfiguriert automatisch AWS Identity and Access Management-(IAM)-Rollen für AWS Systems Manager-Explorer und AWS Systems Manager OpsCenter. Wenn Sie das integrierte Setup abgeschlossen haben, müssen Sie keine zusätzlichen Aufgaben ausführen, um Rollen und Berechtigungen für Explorer zu konfigurieren. Sie müssen jedoch die Berechtigung für OpsCenter konfigurieren, wie weiter unten in diesem Thema beschrieben.

Inhalt

- Informationen zu den durch das integrierte Setup erstellten Rollen
- Konfigurieren von Berechtigungen für Systems Manager OpsCenter

Informationen zu den durch das integrierte Setup erstellten Rollen

Das integrierte Setup erstellt und konfiguriert die folgenden Rollen für die Arbeit mit Explorer und OpsCenter.

- `AWSServiceRoleForAmazonSSM`: Bietet Zugriff auf AWS-Ressourcen, die von Systems Manager verwaltet oder verwendet werden.
- `OpsItem-CWE-Role`: Erlaubt CloudWatch Events und EventBridge die Erstellung von OpsItems als Reaktion auf häufig vorkommende Ereignisse.
- `AWSServiceRoleForAmazonSSM_AccountDiscovery`: Ermöglicht Systems Manager, andere AWS-Services aufzurufen, um AWS-Konto-Informationen beim Synchronisieren von Daten zu entdecken. Weitere Informationen über diese Rolle finden Sie unter Informationen über die `AWSServiceRoleForAmazonSSM_AccountDiscovery`-Rolle.
- `AmazonSSMExplorerExport`: Ermöglicht Explorer den Export von OpsData in eine Datei mit durch Komma getrennten Werten (CSV).

Informationen über die **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-Rolle

Wenn Sie Explorer zur Anzeige von Daten aus mehreren Konten und Regionen mithilfe von AWS Organizations und einer Ressourcendatensynchronisierung konfigurieren, erstellt Systems Manager eine service-verknüpfte Rolle. Systems Manager verwendet diese Rolle, um Informationen über Ihren AWS-Konten in AWS Organizations zu erhalten. Die Rolle verwendet die folgende Berechtigungsrichtlinie.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "organizations:DescribeAccount",
            "organizations:DescribeOrganization",
            "organizations:ListAccounts",
            "organizations:ListAWSServiceAccessForOrganization",
            "organizations:ListChildren",
            "organizations:ListParents"
         ],
         "Resource":"*"
      }
   ]
```

```
}
```

Weitere Informationen zur `AWSServiceRoleForAmazonSSM_AccountDiscovery`-Rolle finden Sie unter Verwenden von Rollen zum Sammeln von AWS-Konto Informationen für OpsCenter und Explorer.

Konfigurieren von Berechtigungen für Systems Manager OpsCenter

Nachdem Sie die integrierte Einrichtung abgeschlossen haben, müssen Sie Benutzer-, Gruppen- oder Rollenberechtigungen konfigurieren, damit Benutzer Aktionen in OpsCenter ausführen können.

Bevor Sie beginnen

Sie können Ihr OpsCenter so konfigurieren, dass Sie OpsItems für mehrere Konten oder nur für ein einziges Konto erstellen und verwalten können. Wenn Sie OpsCenter für die Erstellung und Verwaltung von OpsItems für mehrere Konten konfigurieren, kann das AWS Organizations-Verwaltungskonto OpsItems in anderen Konten manuell erstellen, anzeigen oder bearbeiten. Bei Bedarf können Sie auch das delegierte Administratorkonto von Systems Manager auswählen, um OpsItems in Mitgliedskonten zu erstellen und zu verwalten. Wenn Sie jedoch OpsCenter für ein einzelnes Konto konfigurieren, können Sie OpsItems nur in dem Konto anzeigen oder bearbeiten, in dem OpsItems erstellt wurden. Sie können OpsItems nicht über AWS-Konten freigeben oder übertragen. Aus diesem Grund empfehlen wir Ihnen, Berechtigungen für OpsCenter in dem AWS-Konto zu konfigurieren, das für die Ausführung Ihrer AWS-Workloads verwendet wird. Anschließend können Sie in diesem Konto -Benutzer oder -Gruppen erstellen. Auf diese Weise können mehrere Betriebstechniker oder IT-Experten OpsItems in demselben AWS-Konto erstellen, einsehen und bearbeiten.

Explorer und OpsCenter verwenden die folgenden API-Operationen. Sie können alle Funktionen von Explorer und OpsCenter verwenden, wenn Ihre Benutzer, Gruppe oder Rolle über Zugriff für diese Aktionen verfügen. Sie können auch strengere Zugriffsberechtigungen erstellen, wie weiter unten in diesem Abschnitt beschrieben.

- CreateOpsItem
- CreateResourceDataSync
- DescribeOpsItems
- DeleteResourceDataSync
- GetOpsItem
- GetOpsSummary

- ListResourceDataSync
- UpdateOpsItem
- UpdateResourceDataSync

Wenn Sie möchten, können Sie eine schreibgeschützte Berechtigung angeben, indem Sie Ihrem Konto, Ihrer Gruppe oder Rolle die folgende Inline-Richtlinie hinzufügen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem",
        "ssm:GetOpsSummary",
        "ssm:DescribeOpsItems",
        "ssm:GetServiceSetting",
        "ssm:ListResourceDataSync"
      ],
      "Resource": "*"
    }
  ]
}
```

Weitere Informationen zum Erstellen von IAM-Benutzer-Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch. Weitere Informationen zum Zuweisen dieser Richtlinie zu einer IAM-Gruppe finden Sie unter Zuordnen einer Richtlinie zu einer IAM-Gruppe.

Erstellen Sie wie folgt eine Berechtigung und fügen Sie sie Ihren Benutzern, Gruppen oder Rollen hinzu:

```
{
  "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": "*"
    }
  ]
}
```

Abhängig von der Identitätsanwendung, die Sie in Ihrer Organisation verwenden, können Sie eine der folgenden Optionen auswählen, um den Benutzerzugriff zu konfigurieren.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center-Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter Erstellen einer Rolle für einen externen Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter Erstellen einer Rolle für einen IAM-Benutzer im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Einschränken des Zugriffs auf OpsItems mithilfe von Tags

Sie können auch den Zugriff auf OpsItems einschränken, indem Sie eine IAM-Inlinerichtlinie verwenden, die Tags festlegt. Hier sehen Sie ein Beispiel, das einen Tag-Schlüssel für Abteilung und einen Tag-Wert für Finanzen angibt. Mit dieser Richtlinie kann der Benutzer nur den API-Vorgang GetOpsItem aufrufen, um OpsItems einzusehen, die zuvor mit den Tags Schlüssel=Abteilung und Wert=Finanzen markiert wurden. Benutzer haben keinen Zugriff auf andere OpsItems.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem"
             ],
      "Resource": "*"
      ,
      "Condition": { "StringEquals": { "ssm:resourceTag/Department": "Finance" } }
    }
  ]
}
```

Hier sehen Sie ein Beispiel, das API-Vorgänge zum Einsehen und Aktualisieren von OpsItems angibt. Diese Richtlinie gibt auch zwei Sätze von Tag-Schlüssel-Wert-Paaren an: Abteilung-Finanzen und Projekt-Unity.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:GetOpsItem",
            "ssm:UpdateOpsItem"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "ssm:resourceTag/Department":"Finance",
               "ssm:resourceTag/Project":"Unity"
            }
         }
      }
   ]
}
```

Weitere Informationen zum Hinzufügen von Tags zu einem OpsItem finden Sie unter Manuelles Erstellen der OpsItems.

Aktivieren von Standardregeln

Das integrierte Setup konfiguriert die folgenden Standardregeln in Amazon EventBridge automatisch. Diese Regeln erstellen OpsItems in AWS Systems Manager-OpsCenter. Wenn Sie nicht möchten, dass EventBridge für die folgenden Ereignisse OpsItems erstellt, deaktivieren Sie diese Option im integrierten Setup. Wenn Sie möchten, können Sie OpsCenter als Ziel bestimmter EventBridge-Ereignisse angeben. Weitere Informationen finden Sie unter Konfigurieren von EventBridge-Regeln zum Erstellen von OpsItems. Sie können die Standardregeln auch jederzeit auf der Seite Settings (Einstellungen) deaktivieren.

> **Important**
>
> Sie können die Werte für Category (Kategorie) und Severity (Schweregrad) für Standardregeln nicht bearbeiten, Sie können diese Werte jedoch an OpsItems bearbeiten, die aus den Standardregeln erstellt wurden.

Rule	Category	Severity
CWE rules (11)		
SSMOpsItems-Autoscaling-instance-launch-failure	Availability	2-High
SSMOpsItems-Autoscaling-instance-termination-failure	Availability	2-High
SSMOpsItems-EBS-snapshot-copy-failed	Availability	2-High
SSMOpsItems-EBS-snapshot-creation-failed	Availability	2-High
SSMOpsItems-EBS-volume-performance-issue	Performance	3-Medium
SSMOpsItems-EC2-issue	Availability	2-High
SSMOpsItems-EC2-scheduled-change	Availability	3-Medium
SSMOpsItems-RDS-issue	Availability	2-High
SSMOpsItems-RDS-scheduled-change	Availability	3-Medium
SSMOpsItems-SSM-maintenance-window-execution-failed	Availability	3-Medium
SSMOpsItems-SSM-maintenance-window-execution-timedout	Availability	2-High

Konfigurieren von OpsData-Quellen

Das integrierte Setup aktiviert die folgenden Datenquellen, die Explorer-Widgets ausfüllen.

- AWS Support Center (Sie müssen entweder über einen Business- oder Enterprise-Support-Plan verfügen, um diese Quelle zu aktivieren.)
- AWS Compute Optimizer (Sie müssen entweder über einen Business- oder Enterprise-Support-Plan verfügen, um diese Quelle zu aktivieren.)
- Systems Manager State Manager Association Compliance
- AWS Config-Compliance
- Systems Manager OpsCenter
- Systems Manager Patch Manager Patch Compliance
- Amazon Elastic Compute Cloud (Amazon EC2)
- Systems Manager Inventory
- AWS Trusted Advisor (Sie müssen entweder über einen Business- oder Enterprise-Support-Plan verfügen, um diese Quelle zu aktivieren.)
- AWS Security Hub

Angeben von Tag-Schlüsseln

Beim Einrichten von AWS Systems Manager-Explorer können Sie bis zu fünf Berichts-Tag-Schlüssel angeben. Diese Tag-Schlüssel sollten bereits auf Ihren AWS-Ressourcen vorhanden sein. Dies sind keine neuen Tag-Schlüssel. Nach dem Hinzufügen der Schlüssel zum System können Sie OpsItems in Explorer unter Verwendung dieser Tag-Schlüssel filtern.

> **Note**
>
> Sie können auch auf der Seite Settings (Einstellungen) Berichts-Tag-Schlüssel angeben.

Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen

AWS Systems Manager verwendet eine integrierte Einrichtungsumgebung, um Ihnen den Einstieg in AWS Systems Manager-Explorer und AWS Systems Manager OpsCenter zu erleichtern. Nach Abschluss der integrierten Einrichtung synchronisieren Explorer und OpsCenter automatisch die Daten. Genauer gesagt synchronisieren diese Funktionen OpsData und OpsItems für das AWS-Konto und die AWS-Region, die Sie beim Abschluss der integrierten Einrichtung verwendet haben.

Wenn Sie OpsData und OpsItems von anderen Konten und Regionen aggregieren möchten, müssen Sie eine Ressourcendatensynchronisierung erstellen, wie in diesem Thema beschrieben.

> **Note**
>
> Weitere Hinweise zum integrierten Setup finden Sie unter Erste Schritte mit Systems Manager Explorer und OpsCenter.

Informationen zur Ressourcendatensynchronisierung für Explorer

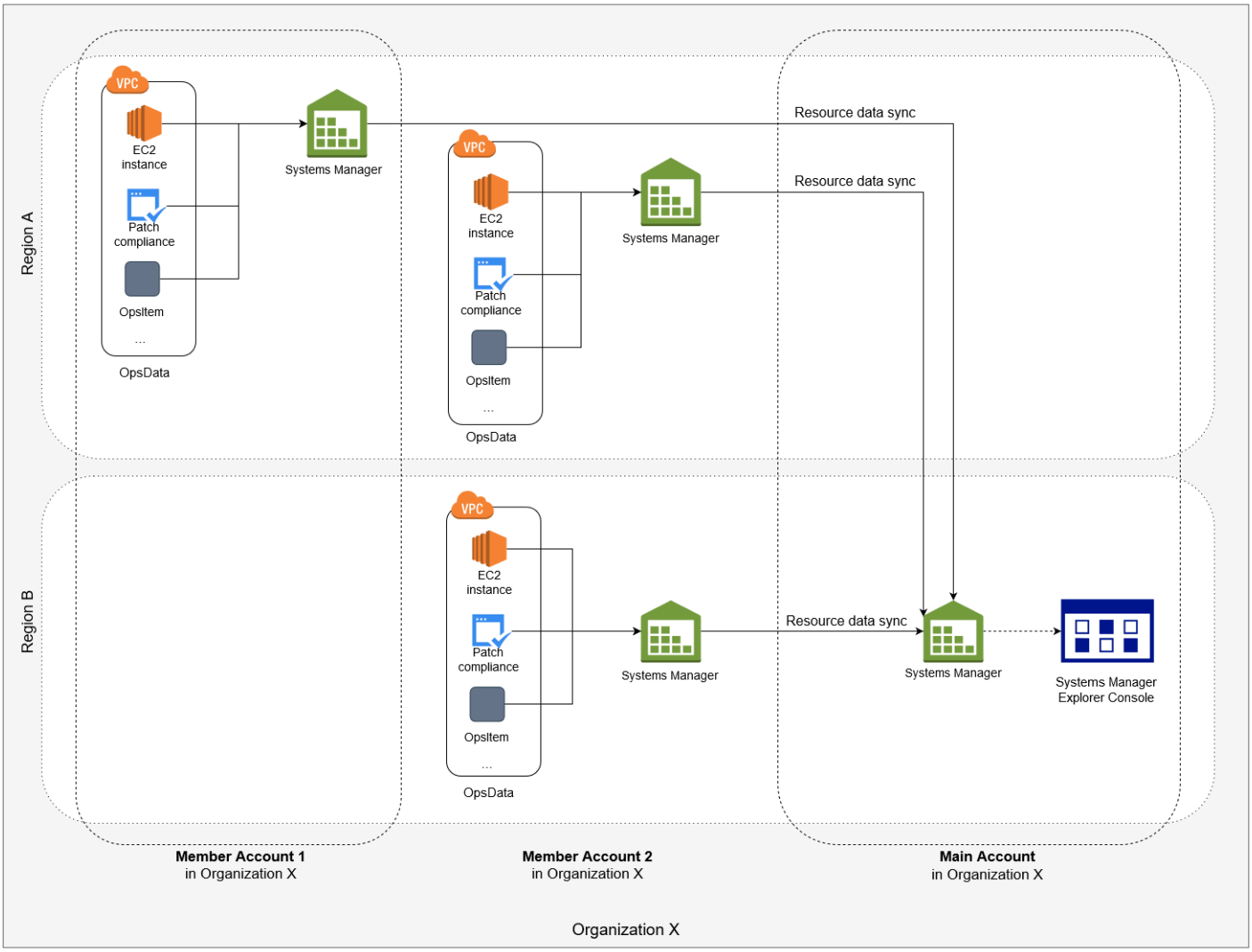
Die Ressourcendatensynchronisierung für Explorer bietet zwei Aggregierungsoptionen:

- Single-account/Multiple-regions (Ein Konto/Mehrere Regionen): Sie können Explorer so konfigurieren, dass OpsItems und OpsData aus mehreren AWS-Regionen aggregiert werden. Der Datensatz ist jedoch auf das aktuelle AWS-Konto beschränkt.
- Multiple-accounts/Multiple-regions (Mehrere Konten/Mehrere Regionen): Sie können Explorer so konfigurieren, dass Daten aus mehreren AWS-Regionen und -Konten aggregiert werden. Diese Option erfordert die Einrichtung und Konfiguration von AWS Organizations. Nachdem Sie AWS Organizations einrichten und konfigurieren, können Sie Daten in Explorer nach Organisationseinheit (OU) oder für eine ganze Organisation aggregieren. Systems Manager aggregiert die Daten in das AWS Organizations-Verwaltungskonto vor der Anzeige in Explorer. Weitere Informationen finden Sie unter Was ist AWS Organizations? im AWS Organizations-Benutzerhandbuch.

> **Warning**
>
> Wenn Sie Explorer konfigurieren, um Daten von einer Organisation in AWS Organizations zu aggregieren, aktiviert das System OpsData in allen Mitgliedskonten in der Organisation. Das Aktivieren von OpsData-Quellen in allen Mitgliedskonten erhöht die Anzahl der Anrufe an OpsCenter-APIs wie CreateOpsItem und GetOpsSummary. Aufrufe dieser API-Aktionen werden Ihnen in Rechnung gestellt.

Das folgende Diagramm zeigt eine Ressourcendatensynchronisierung, die für die Arbeit mit AWS Organizations konfiguriert ist. In diesem Szenario hat der Benutzer zwei Konten in AWS Organizations definiert. Die Ressourcendatensynchronisierung aggregiert Daten aus Konten und

mehreren AWS-Regionen in das AWS Organizations-Hauptkonto, in dem sie dann in Explorer angezeigt werden.



Über die Synchronisierung von Daten mehrerer Konto- und Regions-Ressourcendaten

In diesem Abschnitt werden wichtige Details zur Synchronisierung von mehreren Konto- und mehreren Regions-Ressourcendaten beschrieben, die AWS Organizations verwenden. Die Informationen in diesem Abschnitt gelten insbesondere, wenn Sie auf der Seite Erstellen von Ressourcendaten-Synchronisierung eine der folgenden Optionen wählen:

- Alle Konten aus meiner AWS Organizations-Konfiguration einbeziehen
- Wählen Sie Organisationseinheiten in AWS Organizations

Wenn Sie keine dieser Optionen verwenden möchten, können Sie diesen Abschnitt überspringen.

Wenn Sie eine Ressourcen-Datensynchronisierung in der SSM-Konsole erstellen und eine der AWS Organizations-Optionen gewählt haben, erlaubt Systems Manager automatisch alle OpsData-Quellen in den ausgewählten Regionen für alle AWS-Konten in Ihrer Organisation (oder in den ausgewählten Organisationseinheiten). Zum Beispiel, selbst wenn Sie Explorer in einer Region nicht aktiviert haben, wenn Sie eine AWS Organizations-Option für die Ressourcendatensynchronisierung wählen, sammelt Systems Manager automatisch OpsData aus dieser Region. Um eine Ressourcendaten-Synchronisierung zu erstellen, ohne OpsData-Quellen zuzulassen, geben Sie bei der Erstellung der Daten-Synchronisierung EnableAllOpsDataSources als „false“ an. Weitere Informationen finden Sie unter EnableAllOpsDataSources in der Amazon-EC2-Systems-Manager-API-Referenz.

Wählen Sie keine der AWS Organizations-Optionen für eine Ressourcendatensynchronisierung wählen, müssen Sie das integrierte Setup in jedem Konto und jeder Region abschließen, in der Sie Explorer Zugriff auf Daten gewähren wollen. Wenn dies nicht der Fall ist, zeigt Explorer keine OpsData und OpsItems für die Konten und Regionen an, in denen Sie das integrierte Setup nicht abgeschlossen haben.

Wenn Sie Ihrer Organisation ein untergeordnetes Konto hinzufügen, erlaubt Explorer automatisch alle OpsData Quellen für das Konto. Wenn Sie das untergeordnete Konto zu einem späteren Zeitpunkt aus Ihrer Organisation entfernen, sammelt Explorer weiterhin OpsData aus dem Konto.

Wenn Sie eine vorhandene Ressourcendatensynchronisierung aktualisieren, die eine der AWS Organizations-Optionen verwendet, werden Sie vom System aufgefordert, die Sammlung aller OpsData Quellen für alle Konten und Regionen zu genehmigen, die von der Änderung betroffen sind.

Wenn Sie einen neuen Dienst zu Ihrem AWS-Konto hinzufügen und wenn Explorer OpsData für diesen Service sammelt, konfiguriert Systems Manager automatisch Explorer, um diese OpsData zu sammeln. Wenn Ihr Unternehmen beispielsweise AWS Trusted Advisor nicht verwendet hat, als Sie zuvor eine Ressourcendaten-Synchronisierung erstellt haben, sich aber für diesen Service anmeldet, aktualisiert Explorer automatisch Ihre Ressourcendaten-Synchronisierungen, um diese OpsData zu erfassen.

Important

Beachten Sie die folgenden wichtigen Informationen über mehrere Konto- und Regions-Ressourcendatensynchronisierungen:

- Das Löschen einer Ressourcendatensynchronisierung deaktiviert keine OpsData Quelle in Explorer.

- Um OpsData und OpsItems von mehreren Konten aus anzuzeigen, müssen Sie den AWS Organizations-Modus Alle Features aktiviert haben und beim AWS Organizations-Verwaltungskonto angemeldet sein.

Erstellen einer Ressourcendatensynchronisierung

Beachten Sie die folgenden Details, bevor Sie Ressourcendatensynchronisierung für Explorer konfigurieren.

- Explorer unterstützt maximal fünf Ressourcendatensynchronisierungen.
- Nachdem Sie eine Ressourcendatensynchronisierung für eine Region erstellt haben, können Sie die Kontooptionen für diese Synchronisierung nicht ändern. Wenn Sie beispielsweise eine Synchronisierung in der Region us-east-2 (Ohio) erstellen und die Option Include only the current account (Nur das aktuelle Konto einschließen) auswählen, können Sie diese Synchronisierung später nicht bearbeiten und die Option Include all accounts from my AWS Organizations configuration (Alle Konten aus meiner -Konfiguration einschließen) auswählen. Stattdessen müssen Sie die erste Ressourcendatensynchronisierung löschen und eine neue erstellen. Weitere Informationen finden Sie unter Löschen einer Systems-Manager-Explorer-Ressourcendatensynchronisierung
- OpsData, die in Explorer angezeigt werden, sind schreibgeschützt.

Gehen Sie folgendermaßen vor, um eine Ressourcendatensynchronisierung für Explorer zu erstellen.

Erstellen einer Resource Data Sync

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Settings (Einstellungen) aus.
4. Wählen Sie im Abschnitt Configure resource data sync (Ressourcendatensynchronisierung konfigurieren) die Option Create resource data sync (Ressourcendatensynchronisierung erstellen) aus.
5. Geben Sie unter Resource data sync name (Name der Ressourcen-Datensynchronisierung) einen Namen ein.
6. Wählen Sie im Abschnitt Add accounts (Konten hinzufügen) eine Option aus.

> **Note**
>
> Um eine der AWS Organizations-Optionen verwenden zu können, müssen Sie an dem AWS Organizations-Verwaltungskonto oder an einem delegierten Explorer-Administratorkonto angemeldet sein. Weitere Informationen zu dem delegierten Administratorkonto finden Sie unter Konfigurierung eines delegierten Administrators.

7. Wählen Sie im Abschnitt Regions to include (Einzubeziehende Regionen) eine der folgenden Optionen aus.
   - Wählen Sie All current and future regions (Alle aktuellen und zukünftigen Regionen) aus, um automatisch Daten aus allen aktuellen AWS-Regionen und neuen Regionen zu synchronisieren, die zukünftig online sind,.
   - Wählen Sie All regions (Alle Regionen) aus, um Daten aus allen aktuellen AWS-Regionen automatisch zu synchronisieren.
   - Wählen Sie Regionen, die Sie einbeziehen möchten, einzeln aus.
8. Wählen Sie Create resource data sync (Ressourcen-Datensynchronisierung erstellen).

Nach dem Erstellen einer Ressourcendatensynchronisierung kann das System einige Minuten benötigen, bis Explorer mit Daten gefüllt wird. Sie können die Synchronisierung anzeigen, indem Sie sie aus der Liste Select a resource data sync (Ressourcen-Datensynchronisierung auswählen) in Explorer auswählen.

Konfigurierung eines delegierten Administrators

Wenn Sie AWS Systems Manager-Explorer-Daten von mehreren AWS-Regionen und Konten zusammenfassen, indem Sie die Ressourcendaten-Synchronisierung mit AWS Organizations verwenden, empfehlen wir, dass Sie einen delegierten Administrator für Explorer konfigurieren.

Ein delegierter Administrator kann die folgenden APIs zur Synchronisierung von Explorer-Ressourcendaten über die Konsole, das SDK, AWS Command Line Interface (AWS CLI) oder AWS Tools for Windows PowerShell verwenden:

- CreateResourceDataSync
- DeleteResourceDataSync
- ListResourceDataSync

- UpdateResourceDataSync

Ein delegierter Administrator kann maximal fünf Ressourcendaten-Synchronisierungen für eine gesamte Organisation oder eine Untergruppe von Organisationseinheiten erstellen. Ressourcendatensynchronisierungen, die von einem delegierten Administrator erstellt wurden, sind nur in dem delegierten Administratorkonto verfügbar. Sie können die Synchronisierung und die aggregierten Daten auch nicht im AWS Organizations-Verwaltungskonto anzeigen.

Informationen zur Ressourcendatensynchronisierung finden Sie unter Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen. Weitere Informationen zu AWS Organizations, finden Sie unter Was ist AWS Organizations? im AWS Organizations-Benutzerhandbuch.

Themen

- Konfigurieren eines delegierten Explorer-Administrators
- Registrieren eines delegierten Explorer-Administrators

Konfigurieren eines delegierten Explorer-Administrators

Gehen Sie wie folgt vor, um einen delegierten Explorer-Administrator zu registrieren.

So registrieren Sie einen delegierten Explorer-Administrator

1. Melden Sie sich bei Ihrem AWS Organizations-Verwaltungskonto an.
2. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Explorer aus.
4. Wählen Sie Settings (Einstellungen) aus.
5. Überprüfen Sie im Abschnott Delegierter Administrator für Explorer, ob Sie die erforderlichen dienstverknüpften Rollen- und Servicezugriffsoptionen konfiguriert haben. Wählen Sie bei Bedarf die Schaltflächen Create role (Rolle erstellen) und Enable access (Zugriff gewähren) aus, um diese Optionen zu konfigurieren.
6. Geben Sie für Konto-ID die AWS-Konto-ID ein. Dieses Konto muss ein Mitgliedskonto in AWS Organizations sein.
7. Wählen Sie Register delegated administrator (Delegierten Administrator registrieren).

Der delegierte Administrator hat nun Zugriff auf die Optionen Include all accounts from my AWS Organizations configuration (Alle Konten aus meiner -Konfiguration einschließen) und Select organization units in AWS Organizations (Organisationseinheiten in auswählen) auf der Seite Create resource data sync (Synchronisierung von Ressourcendaten erstellen).

Registrieren eines delegierten Explorer-Administrators

Gehen Sie wie folgt vor, um die Registrierung eines delegierten Explorer-Administrators aufzuheben. Die Registrierung eines delegierten Administratorkonto kann nur vom AWS Organizations-Verwaltungskonto aufgehoben werden. Wenn ein delegiertes Administratorkonto aufgehoben wird, löscht das System alle von dem delegierten Administrator erstellten AWS Organizations-Ressourcendatensynchronisierungen.

So heben Sie die Registrierung eines delegierten Explorer-Administrators auf

1. Melden Sie sich bei Ihrem AWS Organizations-Verwaltungskonto an.
2. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Explorer aus.
4. Wählen Sie Settings (Einstellungen) aus.
5. Wählen Sie Abmelden im Abschnitt Delegierter Administrator für Explorer. Das System zeigt eine Warnung an.
6. Geben Sie die Konto-ID ein und wählen Sie Remove (Entfernen).

Das Konto hat jetzt keinen Zugriff mehr auf die API-Vorgänge zur AWS Organizations-Ressourcendatensynchronisierung. Das System löscht alle AWS Organizations-Ressourcendatensynchronisierungen, die von dem Konto erstellt wurden.

Verwenden von Systems Manager Explorer

Dieser Abschnitt enthält Informationen zum Anpassen von AWS Systems Manager-Explorer durch Ändern des Widget-Layouts und durch Ändern der im Dashboard angezeigten Daten.

Inhalt

- Bearbeiten von Standardregeln für OpsItems
- Bearbeiten von Systems-Manager-Explorer-Datenquellen
- Anpassen der Anzeige und Verwenden von Filtern

- Löschen einer Systems-Manager-Explorer-Ressourcendatensynchronisierung
- Empfangen von Ergebnissen von AWS Security Hub in Explorer

Bearbeiten von Standardregeln für OpsItems

Wenn Sie das integrierte Setup abgeschlossen haben, werden mehr als ein Dutzend Regeln in Amazon EventBridge aktiviert. Diese Regeln erstellen automatisch OpsItems in AWS Systems Manager-OpsCenter. AWS Systems Manager-Explorer zeigt anschließend aggregierte Informationen über OpsItems an.

Jede Regel enthält einen voreingestellten Wert für Category (Kategorie) und Severity (Schweregrad). Wenn das System OpsItems aus einem Ereignis erstellt, weist es automatisch die voreingestellten Werte für Category (Kategorie) und Severity (Schweregrad) zu.

> **Important**
>
> Sie können die Werte für Category (Kategorie) und Severity (Schweregrad) für Standardregeln nicht bearbeiten, Sie können diese Werte jedoch an OpsItems bearbeiten, die aus den Standardregeln erstellt wurden.

Rule	Category	Severity
CWE rules (11)		
SSMOpsItems-Autoscaling-instance-launch-failure	Availability	2-High
SSMOpsItems-Autoscaling-instance-termination-failure	Availability	2-High
SSMOpsItems-EBS-snapshot-copy-failed	Availability	2-High
SSMOpsItems-EBS-snapshot-creation-failed	Availability	2-High
SSMOpsItems-EBS-volume-performance-issue	Performance	3-Medium
SSMOpsItems-EC2-issue	Availability	2-High
SSMOpsItems-EC2-scheduled-change	Availability	3-Medium
SSMOpsItems-RDS-issue	Availability	2-High
SSMOpsItems-RDS-scheduled-change	Availability	3-Medium
SSMOpsItems-SSM-maintenance-window-execution-failed	Availability	3-Medium
SSMOpsItems-SSM-maintenance-window-execution-timedout	Availability	2-High

Bearbeiten von Standardregeln zum Erstellen von OpsItems

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Settings (Einstellungen) aus.
4. Wählen Sie im Abschnitt OpsItems rules (OIS-Regeln) Edit (Bearbeiten).
5. Erweitern Sie CWE rules (CWE-Regeln).
6. Deaktivieren Sie das Kontrollkästchen neben den Regeln, die Sie nicht verwenden möchten.
7. Verwenden Sie die Listen Category (Kategorie) und Severity (Schweregrad), um diese Informationen für eine Regel zu ändern.
8. Wählen Sie Save (Speichern).

Ihre Änderungen werden wirksam, wenn das System das nächste Mal einen OpsItem erstellt.

Bearbeiten von Systems-Manager-Explorer-Datenquellen

AWS Systems Manager-Explorer zeigt Daten aus den folgenden Quellen an. Sie können Explorer-Einstellungen zum Hinzufügen oder Entfernen von Datenquellen bearbeiten:

- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Systems Manager OpsCenter
- AWS Systems Manager Patch Manager Patch Compliance
- AWS Systems Manager State Manager Informationen zu Zuordnungs-Compliance
- AWS Trusted Advisor
- AWS Compute Optimizer
- AWS Support Center Fälle
- AWS Config Rege- und Ressource-Compliance
- AWS Security Hub-Erkenntnisse

> **Note**
>
> - Um AWS Support Center-Fälle inExplorer anzeigen zu können, müssen Sie entweder ein Enterprise- oder ein Business-Konto beim AWS Support-Support eingerichtet haben.

- Sie können Explorer nicht so konfigurieren, dass OpsCenter-OpsItem-Daten nicht mehr angezeigt werden.

Bevor Sie beginnen

Stellen Sie sicher, dass Sie Services eingerichtet und konfiguriert haben, die Explorer-Widgets mit Daten füllen. Weitere Informationen finden Sie unter Einrichten von zugehörigen Services.

So bearbeiten Sie Datenquellen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Settings (Einstellungen) aus.
4. Wählen Sie im Abschnitt OpsData sources (OpsData-Quellen) die Option Edit (Bearbeiten).
5. Erweitern Sie OpsData sources (OpsData-Quellen).
6. Fügen Sie eine oder mehrere Quellen hinzu oder entfernen Sie sie.
7. Wählen Sie Save (Speichern).

Anpassen der Anzeige und Verwenden von Filtern

Sie können das Widget-Layout in AWS Systems Manager-Explorer mithilfe einer Drag-&-Drop-Funktion anpassen. Sie können OpsData und OpsItems, die in Explorer angezeigt werden, auch mithilfe von Filtern anpassen, wie in diesem Thema beschrieben.

Bevor Sie beginnen

Vergewissern Sie sich vor dem Anpassen des Widget-Layouts, dass die Widgets, die Sie anzeigen möchten, derzeit in Explorer angezeigt werden. Um einige Widgets in Explorer (z. B. das AWS Config-Compliance-Widget) anzuzeigen, müssen Sie sie auf der Seite Configure dashboard (Dashboard konfigurieren) aktivieren.

So aktivieren Sie die Anzeige von Widgets in Explorer

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

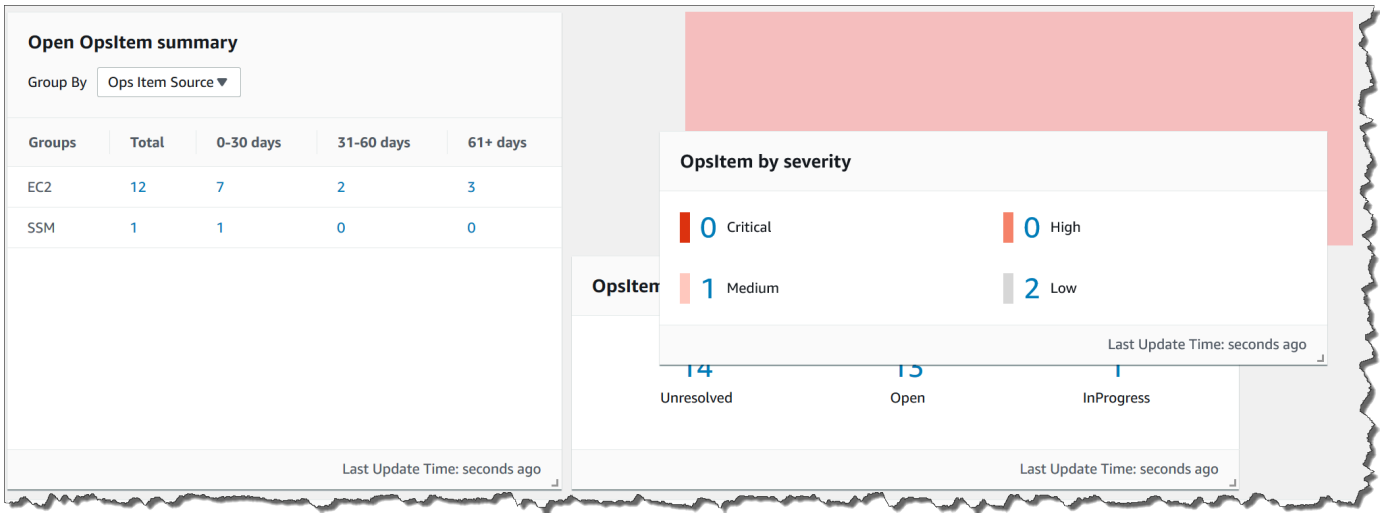
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Dashboard actions (Dashboard-Aktionen), Configure dashboard (Dashboard konfigurieren).
4. Wählen Sie die Registerkarte Configure Dashboard (Dashboard konfigurieren).
5. Wählen Sie entweder Enable all (Alle aktivieren) oder aktivieren Sie ein einzelnes Widget oder eine einzelne Datenquelle.
6. Wählen Sie Explorer, um Ihre Änderungen anzuzeigen.

Anpassen des Widget-Layouts

Gehen Sie wie folgt vor, um das Widget-Layout in Explorer anzupassen.

Anpassen des Widget-Layouts

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie ein Widget, das Sie verschieben möchten.
4. Klicken Sie auf den Namen des Widgets, halten Sie es und ziehen Sie es dann an seine neue Position.



5. Wiederholen Sie diesen Vorgang für jedes Widget, das Sie neu positionieren möchten.

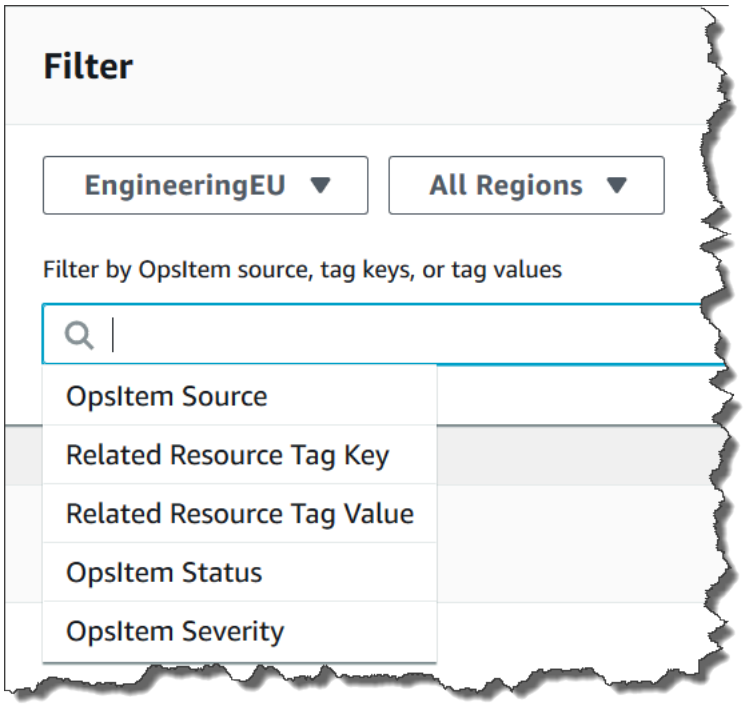
Wenn Sie sich entscheiden, dass Ihnen das neue Layout nicht gefällt, wählen Sie Reset layout (Layout zurücksetzen), um alle Widgets wieder an ihren ursprünglichen Speicherort zu verschieben.

Verwenden von Filtern zum Ändern der angezeigten Daten in Explorer

Standardmäßig zeigt Explorer Daten für das aktuelle AWS-Konto und die aktuelle Region an. Wenn Sie mindestens eine Ressourcen-Datensynchronisierung erstellen, können Sie mithilfe von Filtern ändern, welche Synchronisierung aktiv ist. Sie können dann wählen, ob Daten für eine bestimmte Region oder für alle Regionen angezeigt werden sollen. Sie können die Suchleiste auch verwenden, um nach verschiedenen OpsItem- und Schlüssel-Tag-Kriterien zu filtern.

Ändern der in Explorer angezeigten Daten mithilfe von Filtern

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Verwenden Sie im Abschnitt Filter die Liste Select a resource data sync (Ressourcen-Datensynchronisierung auswählen), um eine Synchronisierung auszuwählen.
4. Verwenden Sie die Liste Regions (Regionen), um entweder eine bestimmte AWS-Region oder All Regions (Alle Regionen) auszuwählen.
5. Wählen Sie die Suchleiste und dann die Kriterien aus, nach denen die Daten gefiltert werden sollen.



6. Drücken Sie die Eingabetaste.

Explorer behält die ausgewählten Filteroptionen bei, wenn Sie die Seite schließen und erneut öffnen.

Löschen einer Systems-Manager-Explorer-Ressourcendatensynchronisierung

In AWS Systems Manager-Explorer können Sie OpsData und OpsItems aus anderen Konten und Regionen zusammenfassen, indem Sie eine Ressourcendaten-Synchronisierung erstellen.

Sie können die Kontooptionen für eine Ressourcen-Datensynchronisierung nicht ändern. Wenn Sie beispielsweise eine Synchronisierung in der Region us-east-2 (Ohio) erstellt haben und die Option Include only the current account (Nur das aktuelle Konto einschließen) ausgewählt haben, können Sie diese Synchronisierung später nicht bearbeiten und die Option Include all accounts from my AWS Organizations configuration (Alle Konten aus meiner -Konfiguration einschließen) auswählen. Stattdessen müssen Sie die Ressourcen-Datensynchronisierung löschen und eine neue erstellen, wie im folgenden Verfahren beschrieben.

Löschen einer Resource Data Sync

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Settings (Einstellungen) aus.
4. Wählen Sie im Abschnitt Configure resource data sync (Ressourcen-Datensynchronisierung konfigurieren) die Ressourcen-Datensynchronisierung aus, die Sie löschen möchten.
5. Wählen Sie Delete (Löschen).

Empfangen von Ergebnissen von AWS Security Hub in Explorer

AWS Security Hubbietet einen umfassenden Überblick über Ihren Sicherheitsstatus in AWS. Der Dienst sammelt Sicherheitsdaten, so genannte Erkenntnisse, aus allen AWS-Konten, Diensten und unterstützten Produkten von Drittanbietern. Die Erkenntnisse von Security Hub können Ihnen dabei helfen, Ihre Umgebung anhand von Branchenstandards und bewährten Methoden zu überprüfen, Ihre Sicherheitstrends zu analysieren und die Sicherheitsprobleme mit der höchsten Priorität zu identifizieren.

Security Hub sendet Ergebnisse an Amazon EventBridge, das die Ergebnisse mithilfe einer Ereignisregel an Amazon sendetExplorer. Nachdem Sie die Integration wie hier beschrieben aktiviert haben, können Sie die Erkenntnisse von Security Hub in einem Explorer-Widget und die Details zu

den Erkenntnissen in OpsCenter OpsItemsanzeigen. Das Widget bietet eine Zusammenfassung aller Security-Hub-Erkenntnisse nach Schweregrad. Neue Erkenntnisse in Security Hub sind normalerweise innerhalb von Sekunden nach ihrer Erstellung in Explorer sichtbar.

> **Warning**
>
> Beachten Sie die folgenden wichtigen Informationen:
>
> - Explorer ist integriert mit OpsCenter, einer Funktion von Systems Manager. Nachdem Sie die Explorer-Integration mit Security Hub aktiviert haben, erstellt OpsCenter automatisch OpsItems für Security-Hub-Erkenntnisse. Abhängig von Ihrer AWS Umgebung kann die Aktivierung der Integration zu einer großen Anzahl von Daten führenOpsItems, was mit Kosten verbunden ist.
>
> Bevor Sie fortfahren, lesen Sie über die OpsCenter-Integration mit Security Hub. Das Thema enthält spezifische Details darüber, wie Änderungen und Aktualisierungen von Erkenntnissen und OpsItems Ihrem Konto berechnet werden. Weitere Informationen finden Sie unter AWS Security Hub. OpsCenter-Preisinformationen finden Sie unter AWS Systems Manager -Preise.
> - Wenn Sie eine Ressourcendaten-Synchronisierung in Explorer erstellen, während Sie im Administratorkonto angemeldet sind, wird die Security-Hub-Integration automatisch für den Administrator und alle Mitgliedskonten in der Synchronisierung aktiviert. Sobald diese Funktion aktiviert ist, erstellt OpsCenter automatisch OpsItems für Security-Hub-Erkenntnisse, was mit Kosten verbunden ist. Weitere Informationen zum Erstellen einer Ressourcendaten-Synchronisierung finden Sie unter Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen.

Ergebnisarten, die Explorer empfängt

Explorer empfängt alle Ergebnisse von Security Hub Wenn Sie die Standardeinstellungen für Security Hub aktivieren, können Sie im Explorer-Widget alle Erkenntnisse nach Schweregrad sortiert sehen. Standardmäßig erstellt Explorer OpsItems für kritische und hochgradig schwerwiegende Befunde. Sie können Explorer manuell konfigurieren, um OpsItems für Erkenntnisse mit mittlerem und niedrigem Schweregrad zu erstellen.

ExplorerEs wird zwar nicht OpsItems für informative Ergebnisse erstellt, Sie können jedoch informative Betriebsdaten (OpsData) im Security Hub Hub-Widget mit der Zusammenfassung der

Ergebnisse anzeigen. Explorererstellt OpsData für alle Ergebnisse unabhängig vom Schweregrad. Weitere Informationen zu den Schweregraden von Security Hub finden Sie unter Schweregrad in der AWS Security Hub -API-Referenz.

Aktivieren der Integration

In diesem Abschnitt wird beschrieben, wie Sie Explorer für den Empfang von Security-Hub-Erkenntnissen aktivieren und konfigurieren.

Bevor Sie beginnen

Führen Sie die folgenden Aufgaben aus, bevor Sie Explorer für den Empfang von Security Hub-Ergebnissen konfigurieren.

- Aktivieren und konfigurieren von Security Hub. Weitere Informationen finden Sie unter Setting up Security Hub (Einrichten von Security Hub) im AWS Security Hub -Benutzerhandbuch.
- Loggen Sie sich in das AWS Organizations Verwaltungskonto ein. Systems Manager erfordert Zugriff auf AWS Organizations zum Erstellen von OpsItems aus Security Hub-Ergebnissen. Nachdem Sie sich beim Verwaltungskonto angemeldet haben, werden Sie aufgeforderter die Schaltfläche Enable access (Zugriff erlauben) auf der Registerkarte Explorer Configure Dashboard zu wählen, wie nachfolgend beschrieben: Wenn Sie sich nicht beim AWS Organizations Verwaltungskonto anmelden, können Sie keinen Zugriff gewähren und Explorer keine Ergebnisse OpsItems aus Security Hub erstellen.

Security Hub-Ergebnisse erhalten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Klicken Sie auf Settings (Einstellungen).
4. Wählen Sie die Registerkarte Configure dashboard.
5. Wählen Sie AWS Security Hub.
6. Wählen Sie den Schieberegler Disabled, um AWS Security Hub zu aktivieren.

   Kritische und schwerwiegende Erkenntnisse werden standardmäßig angezeigt. Um Erkenntnisse mit mittlerem und niedrigem Schweregrad anzuzeigen, wählen Sie den Schieberegler Deaktiviert neben Mittel, Niedrig.

7. Wählen Sie im Abschnitt OpsItems created by Security Hub findings die Option Enable access (Zugriff gewähren). Wenn Sie diese Schaltfläche nicht sehen, melden Sie sich beim AWS Organizations Verwaltungskonto an und kehren Sie zu dieser Seite zurück, um die Schaltfläche auszuwählen.

Security Hub-Ergebnisse anzeigen

Im folgenden Verfahren wird beschrieben, wie Sie Security Hub-Ergebnisse anzeigen.

Security Hub-Ergebnisse anzeigen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Suchen Sie das Widget AWS Security Hub findings summary (Ergebniszusammenfassung). Hier werden Ihre Security Hub-Ergebnisse angezeigt. Sie können einen Schweregrad auswählen, um eine detaillierte Beschreibung der entsprechenden OpsItem auszuwählen.

Empfangen von Ergebnisse stoppen

Im folgenden Verfahren wird beschrieben, wie Sie das Empfangen von Security Hub-Ergebnissen stoppen.

Erhalt von Security Hub-Ergebnissen stoppen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Klicken Sie auf Settings (Einstellungen).
4. Wählen Sie die Registerkarte Configure dashboard.
5. Wählen Sie den Schieberegler Enabled (aktiviert), um AWS Security Hub zu deaktivieren..

> **Important**
>
> Wenn die Option zum Deaktivieren der Security Hub Hub-Ergebnisse in der Konsole ausgegraut ist, können Sie diese Einstellung deaktivieren, indem Sie den folgenden Befehl in

der AWS CLI ausführen. Sie müssen den Befehl ausführen, während Sie entweder mit dem AWS Organizations Verwaltungskonto oder dem delegierten Administratorkonto von Systems Manager angemeldet sind. Geben Sie für den `region` Parameter an, AWS-Region wo Sie den Empfang von Security Hub Hub-Ergebnissen beenden möchtenExplorer.

```
aws ssm update-service-setting --setting-id /ssm/opsdata/SecurityHub --setting-
value Disabled --region AWS-Region
```

Ein Beispiel:

```
aws ssm update-service-setting --setting-id /ssm/opsdata/SecurityHub --setting-
value Disabled --region us-east-1
```

OpsData Aus Systems Manager exportieren Explorer

Sie können 5.000 OpsData Artikel als Datei mit kommagetrennten Werten (.csv) aus AWS Systems Manager dem Explorer in einen Amazon Simple Storage Service (Amazon S3) -Bucket exportieren. Der Explorer verwendet das AWS-ExportOpsDataToS3Automations-Runbook für den Export. OpsData Beim Exportieren zeigt das System die Automations-Runbook-Seite an OpsData, auf der Sie Details wie AssumeRole, den Amazon S3 S3-Bucket-Namen, den SNS-Thema-ARN und die zu exportierenden Felder angeben können.

Um zu exportieren: OpsData

- Schritt 1: Festlegen eines SNS-Themas
- Schritt 2: (Optional) Datenexport konfigurieren
- Schritt 3: Exportieren OpsData

Schritt 1: Festlegen eines SNS-Themas

Wenn Sie den Datenexport konfigurieren, müssen Sie ein Amazon Simple Notification Service (Amazon SNS) -Thema angeben, das in derselben AWS-Region Datei vorhanden ist, in die Sie die Daten exportieren möchten. Wenn ein Export abgeschlossen ist, sendet Systems Manager eine Benachrichtigung an das Amazon SNS-Thema. Informationen zum Erstellen eines Amazon-SNS-Themas finden Sie unter Erstellen eines Amazon-SNS-Themas.

Schritt 2: (Optional) Datenexport konfigurieren

Sie können die Einstellungen für den Datenexport auf der Seite „Einstellungen“ oder „Ops-Daten in S3-Bucket exportieren“ konfigurieren.

So konfigurieren Sie den Datenexport von Explorer

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Settings (Einstellungen) aus.
4. Wählen Sie im Abschnitt Configure data export (Datenexport konfigurieren) die Option Edit (Bearbeiten).
5. Um die Datenexport-Datei in einen vorhandenen Amazon-S3-Bucket hochzuladen, Wählen Sie einen vorhandenen S3-Bucket aus und wählen Sie den Bucket aus der Liste aus.

   Um die Datenexportdatei in einen neuen Amazon-S3-Bucket hochzuladen, wählen Sie Neuen S3-Bucket erstellen und geben den Namen ein, den Sie für den neuen Bucket verwenden möchten.

   > **Note**
   >
   > Sie können den Namen des Amazon-S3-Buckets und den Amazon-SNS-Themen-ARN nur auf der Seite bearbeiten, auf der Sie diese Einstellungen zum ersten Mal in Explorer konfiguriert haben. Wenn Sie den Amazon-S3-Bucket und den Amazon-SNS-Themen-ARN auf der Seite Einstellungen eingerichtet haben, können Sie diese Einstellungen nur auf der Seite Einstellungen ändern.

6. Wählen Sie unter Amazon-SNS-Themen-ARN auswählen das Thema aus, das Sie benachrichtigen möchten, wenn der Export abgeschlossen ist.
7. Wählen Sie Erstellen.

Schritt 3: Exportieren OpsData

Wenn Sie Explorer Daten exportieren, erstellt Systems Manager eine AWS Identity and Access Management (IAM-) Rolle mit dem Namen`AmazonSSMExplorerExportRole`. Diese Rolle verwendet die folgende IAM-Richtlinie.

```
{
      "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": [
                  "{{SnsTopicArn}}"
              ]
          },
          {
              "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": [
                "*"
            ]
        }
    ]
}
```

Die Rolle umfasst die folgende Vertrauenseinheit.

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

Um zu exportieren OpsData aus Explorer

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Tabelle exportieren.

> **Note**
>
> Wenn Sie OpsData zum ersten Mal exportieren, erstellt das System eine Rolle für den Export. Sie können die standardmäßig angenommene Rolle nicht ändern.

4. Wählen Sie für Amazon-S3-Bucket-Name einen bestehenden Bucket. Sie können Erstellen wählen, um einen Amazon-S3-Bucket zu erstellen, falls erforderlich. Wenn Sie den Namen des S3-Buckets nicht ändern können, bedeutet dies, dass Sie den Bucket-Namen auf der Seite Einstellungen konfiguriert haben. Sie können den Bucket-Namen nur auf der Seite Einstellungen ändern.

> **Note**
>
> Sie können den Namen des Amazon-S3-Buckets und den Amazon-SNS-Themen-ARN nur auf der Seite bearbeiten, auf der Sie diese Einstellungen zum ersten Mal in Explorer konfiguriert haben.

5. Wählen Sie für SNS-Themen-ARN einen bestehenden Amazon-SNS-Themen-ARN, der benachrichtigt werden soll, wenn der Download abgeschlossen ist.

   Wenn Sie den ARN des Amazon SNS-Themas nicht ändern können, bedeutet dies, dass Sie den ARN des Amazon SNS-Themas auf der Seite Einstellungen konfiguriert haben. Sie können den Themen-ARN nur auf der Seite Einstellungen ändern.

6. (Optional) Geben Sie für SNS-Erfolgsmeldung eine Erfolgsmeldung an, die angezeigt werden soll, wenn der Export erfolgreich abgeschlossen wurde.
7. Wählen Sie Absenden aus. Das System navigiert zur vorherigen Seite und zeigt die Meldung Hier klicken, um den Status des Exportvorgangs anzuzeigen. Details anzeigen.

Sie können Details anzeigen wählen, um den Status des Runbooks und den Fortschritt in Systems Manager Automation anzuzeigen.

Sie können jetzt OpsData aus dem Explorer angegebenen Amazon S3 S3-Bucket exportieren.

Wenn Sie mit diesem Verfahren keine Daten exportieren können, stellen Sie sicher, dass Ihr Benutzer, Ihre Gruppe oder Rolle die `iam:CreatePolicyVersion`- und `iam:DeletePolicyVersion`-Aktionen einschließt. Weitere Informationen zum Hinzufügen dieser Aktionen zu Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle finden Sie unter Bearbeiten von IAM-Richtlinien im IAM-Benutzerhandbuch.

Fehlerbehebung von Systems Manager Explorer

Dieses Thema enthält Informationen zum Beheben gängiger Probleme mit AWS Systems Manager-Explorer.

Kann AWS-Ressourcen in Explorer nach dem Aktualisieren von Tags auf der Seite Settings (Einstellungen) nicht filtern

Wenn Sie Tag-Schlüssel oder andere Dateneinstellungen im Explorer aktualisieren, kann das System bis zu sechs Stunden benötigen, um Daten basierend auf Ihren Änderungen zu synchronisieren.

Die AWS Organizations-Optionen auf der Seite Create resource data sync (Ressourcendatensynchronisierung erstellen) sind ausgegraut

Die Optionen Include all accounts from my AWS Organizations configuration (Alle Konten von meiner AOlong-Konfiguration einbeziehen) und Select organization units in AWS Organizations(Organisationseinheiten in AOlong auswählen auf der Seite Create resource data sync (Ressourcendatensynchronisierung erstellen) sind nur verfügbar, wenn Sie AWS Organizations eingerichtet und konfiguriert haben. Wenn Sie AWS Organizations eingerichtet und konfiguriert haben, kann das AWS Organizations-Verwaltungskonto oder ein delegierter Explorer-Administrator Ressourcendatensynchronisierungen erstellen, die diese Optionen verwenden.

Weitere Informationen finden Sie unter Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen und Konfigurierung eines delegierten Administrators.

Explorer zeigt überhaupt keine Daten an

- Stellen Sie sicher, dass Sie das integrierte Setup in jedem Konto und jeder Region abgeschlossen haben, in der Explorer Daten aufrufen und anzeigen soll. Wenn dies nicht der Fall ist, zeigt Explorer

keine OpsData und OpsItems für die Konten und Regionen an, in denen Sie das integrierte Setup nicht abgeschlossen haben. Weitere Informationen finden Sie unter Erste Schritte mit Systems Manager Explorer und OpsCenter.

- Wenn Sie Explorer aus mehreren Konten und Regionen anzeigen, stellen Sie sicher, dass Sie am AWS Organizations-Verwaltungskonto angemeldet sind. Zum Anzeigen von OpsData und OpsItems von mehreren Konten und Regionen müssen Sie bei diesem Konto angemeldet sein.

Widgets zu Amazon-EC2-Instances zeigen keine Daten an

Wenn Widgets zu Amazon Elastic Compute Cloud (Amazon EC2)-Instances, z. B. Instance count (Anzahl der Instances), Managed Instances (Verwaltete Instances) und Instance by AMI (Instance nach AMI) keine Daten anzeigen, überprüfen Sie Folgendes:

- Stellen Sie sicher, dass Sie mehrere Minuten gewartet haben. OpsData kann einige Minuten benötigen, bis es in Explorer angezeigt wird, nachdem Sie das integrierte Setup abgeschlossen haben.
- Stellen Sie sicher, dass Sie AWS Config Configuration Recorder konfiguriert haben. Explorer verwendet Daten, die von AWS Config Configuration Recorder bereitgestellt werden, um Widgets mit Informationen zu Ihren EC2-Instances zu füllen. Weitere Informationen finden Sie unter Verwalten von Configuration Recorder.
- Stellen Sie sicher, dass die Amazon EC2-OpsData-Quelle auf der Seite Settings (Einstellungen) aktiviert ist. Stellen Sie außerdem sicher, dass mehr als 6 Stunden vergangen sind, seit Sie den Konfigurationsrekorder aktiviert oder seit Sie Änderungen an Ihren Instances vorgenommen haben. Systems Manager kann bis zu sechs Stunden benötigen, um Daten von AWS Config in Explorer EC2-Widgets anzuzeigen, nachdem Sie den Konfigurationsrekorder aktiviert oder Änderungen an Ihren Instance vorgenommen haben.
- Wenn eine Instance angehalten oder beendet wird, beachten Sie, dass Explorer diese Instances nach 24 Stunden nicht mehr anzeigt.
- Stellen Sie sicher, dass Sie sich in der richtigen AWS-Region befinden, in der Sie Ihre Amazon–EC2-Instances konfiguriert haben. Explorer zeigt keine Daten über On-Premises-Instances an.
- Wenn Sie eine Ressourcen-Datensynchronisierung für mehrere Konten und Regionen konfiguriert haben, stellen Sie sicher, dass Sie beim Organizations-Verwaltungskonto angemeldet sind.

Das Patch-Widget zeigt keine Daten an

Das Widget Non-compliant instances for patching (Nicht konforme Instances für das Patchen) zeigt nur Daten über Patch-Instances an, die nicht kompatibel sind. Dieses Widget zeigt keine Daten an, wenn Ihre Instances konform sind. Wenn Sie vermuten, dass Sie nicht konforme Instances haben, stellen Sie sicher, dass Sie Systems Manager-Patches eingerichtet und konfiguriert haben, und verwenden Sie AWS Systems Manager Patch Manager, um die Patch-Compliance zu überprüfen. Weitere Informationen finden Sie unter AWS Systems Manager Patch Manager.

Sonstige Probleme

Explorer lässt Sie OpsItems nicht bearbeiten und keine Fehler beheben: Über Konten oder Regionen hinweg angezeigte OpsItems sind schreibgeschützt. Sie können nur über ihr Heimatkonto oder ihre Region aktualisiert und korrigiert werden.

AWS Systems Manager OpsCenter

OpsCenter, eine Funktion von AWS Systems Manager, bietet einen zentralen Ort, an dem Betriebsingenieure und IT-Experten betriebliche Arbeitsaufgaben (OpsItems) im Zusammenhang mit AWS Ressourcen verwalten können. Ein OpsItem ist ein betriebliches Problem oder eine Unterbrechung, die untersucht und behoben werden muss. Mithilfe von OpsCenter können Sie kontextbezogene Untersuchungsdaten zu jedem OpsItem anzeigen, einschließlich zugehöriger OpsItems und zugehöriger Ressourcen. Sie können auch Systems-Manager-Automation-Runbooks ausführen, um OpsItems zu beheben.

Jede OpsItem enthält die relevanten Informationen, wie den Namen und die ID der AWS Ressource, die die generiert hatOpsItem, die zur Behebung eines Ereignisses erforderlich sind. Wenn Sie es einrichten OpsCenter und mit anderen integrieren AWS-Services, kann es OpsItems automatisch erstellt werden. Wenn in diese Dienste integriert, OpsCenter werden Informationen von AWS Config, und Amazon angezeigt AWS CloudTrail, EventBridge um Ihnen bei der Untersuchung eines zu helfenOpsItem. Auf diese Weise müssen Sie für Ihre Untersuchung nicht mehr zwischen den Konsolenseiten navigieren.

Sie können OpsCenter verwenden, um Probleme mit Ihren verwalteten On-Premises-Knoten zu untersuchen und zu beheben, die für Systems Manager konfiguriert sind. Weitere Informationen zum Einrichten und Konfigurieren On-Premises-Server und virtueller Computer für Systems Manager finden Sie unter Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen.

Sie können mit OpsCenter der Systems Manager Manager-Konsole AWS Command Line Interface (AWS CLI) oder dem AWS SDK Ihrer Wahl arbeiten. AWS Tools for PowerShell Mithilfe von AWS

Identity and Access Management (IAM-) Richtlinien können Sie entscheiden, welche Mitglieder Ihrer Organisation Daten erstellen, anzeigen, auflisten und aktualisieren OpsItems können. Sie können OpsItems auch Tags zuweisen und anschließend IAM-Richtlinien erstellen, die Benutzern und Gruppen basierend auf Tags Zugriff gewähren.

> **Note**
>
> Für die Verwendung des OpsCenter fallen Gebühren an. Weitere Informationen finden Sie unter AWS Systems Manager -Preise.
> Sie können Kontingente für alle Systems-Manager-Funktionen unter Service Quotas für Systems Manager in der Allgemeine Amazon Web Services-Referenz. anzeigen. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region.

OpsCenter-Workflow

Führen Sie die folgenden Schritte aus, um OpsCenter zur Behebung von OpsItems einzurichten und damit zu arbeiten:

1. OpsCenter einrichten. Sie können auch OpsCenter einrichten, sodass OpsItems in allen Konten zentral verwaltet werden.
2. Integrieren Sie OpsCenter mit anderen AWS-Services. OpsCenterkann in Amazon CloudWatch, Amazon CloudWatch Application Insights, Amazon EventBridge, Amazon DevOps Guru, AWS Config AWS Security Hub, und integriert AWS Systems Manager Incident Manager werden.
3. Erstellen von OpsItems. Sie können OpsItems automatisch oder manuell erstellen.
4. Verwalten Sie OpsItems, indem Sie Kontext zu zugehörigen Ressourcen, zugehörigen OpsItems und Betriebsdaten hinzufügen und OpsItems-Duplikate entfernen.
5. Beheben Sie OpsItems mithilfe von Systems-Manager-Automation-Runbooks.

Einrichten von OpsCenter

AWS Systems Manager verwendet ein integriertes Setup-ErlebnisExplorer, um Ihnen den Einstieg in die Funktionen von Systems Manager zu erleichtern. OpsCenter Explorerist ein anpassbares Operations-Dashboard, das Informationen über Ihre AWS Ressourcen enthält. In dieser Dokumentation wird die Einrichtung von Explorer und OpsCenter als Integrierte Einrichtung bezeichnet.

Für die Einrichtung müssen Sie die integrierte Einrichtung verwenden, um OpsCenter mit Explorer einzurichten. Das integrierte Setup ist nur in der AWS Systems Manager Konsole verfügbar. Sie können Explorer und OpsCenter nicht programmatisch einrichten. Weitere Informationen finden Sie unter Erste Schritte mit Systems Manager Explorer und OpsCenter.

Durch das Setup aktivierte Standardregeln

Bei der Einrichtung OpsCenter aktivieren Sie Standardregeln in Amazon EventBridge , die automatisch erstellt werdenOpsItems. In der folgenden Tabelle werden die EventBridge Standardregeln beschrieben, die automatisch erstellt werdenOpsItems. Sie können EventBridge Regeln auf der Seite OpsCenter Einstellungen unter OpsItemRegeln deaktivieren.

> **Important**
>
> Wir stellen Ihnen OpsItems in Rechnung, die mit Standardregeln erstellt werden. Weitere Informationen finden Sie unter AWS Systems Manager -Preisgestaltung.

Regelname	Beschreibung
SSMOpsItems-Autoscaling-instance-launch-failure	Diese Regel erstellt OpsItems, wenn der Start einer Auto-Scaling-EC2-Instance fehlschlägt.
SSMOpsItems-Autoscaling-instance-termination-failure	Diese Regel erstellt OpsItems, wenn das Beenden einer Auto-Scaling-EC2-Instance fehlschlägt.
SSMOpsItems-EBS-snapshot-copy-failed	Diese Regel erstellt OpsItems, wenn das System einen Snapshot des Amazon Elastic Block Store (Amazon EBS) nicht erfolgreich kopieren konnte.
SSMOpsItems-EBS-snapshot-creation-failed	Diese Regel erstellt OpsItems, wenn das System einen Amazon-EBS-Snapshot nicht erfolgreich erstellen konnte.
SSMOpsItems-EBS-volume-performance-issue	Diese Regel entspricht einer AWS Health Tracking-Regel. Die Regel erstellt OpsItems, wenn es ein Leistungsproblem mit einem

Regelname	Beschreibung
	Amazon-EBS-Volume gibt (Zustandsereignis = `AWS_EBS_DEGRADED_EBS_VOLUME_PERFORMANCE`).
SSMOpsItems-EC2-issue	Diese Regel entspricht einer AWS Health Verfolgungsregel für unerwartete Ereignisse, die sich auf AWS Dienste oder Ressourcen auswirken. Die Regel erstellt OpsItems, wenn beispielsweise ein Dienst Mitteilungen über betriebliche Probleme sendet, die zu einer Beeinträchtigung des Dienstes führen, oder um auf lokale Probleme auf Ressourcenebene aufmerksam zu machen. Diese Regel erstellt beispielsweise ein OpsItem für das folgende Ereignis: `AWS_EC2_OPERATIONAL_ISSUE` .
SSMOpsItems-EC2-scheduled-change	Diese Regel entspricht einer AWS Health Verfolgungsregel. AWS kann Ereignisse für Ihre Instances planen, z. B. das Neustarten, Stoppen oder Starten von Instances. Die Regel erstellt OpsItems für geplante Ereignisse in EC2. Weitere Informationen zu geplanten Ereignissen finden Sie unter Geplante Ereignisse für Ihre Instances im Amazon EC2 EC2-Benutzerhandbuch.

Regelname	Beschreibung
SSMOpsItems-RDS-issue	Diese Regel entspricht einer AWS Health Verfolgungsregel für unerwartete Ereignisse, die sich auf AWS Dienste oder Ressourcen auswirken. Die Regel erstellt OpsItems, wenn beispielsweise ein Dienst Mitteilungen über betriebliche Probleme sendet, die zu einer Beeinträchtigung des Dienstes führen, oder um auf lokale Probleme auf Ressourcenebene aufmerksam zu machen. Diese Regel erstellt beispielsweise ein OpsItem für die folgenden Ereignisse: `AWS_RDS_MYSQL_DATABASE_CRASHING_REPEATEDLY` , `AWS_RDS_EXPORT_TASK_FAILED` und `AWS_RDS_CONNECTIVITY_ISSUE` .
SSMOpsItems-RDS-scheduled-change	Diese Regel entspricht einer AWS Health Verfolgungsregel. Die Regel erstellt OpsItems für geplante Ereignisse in Amazon RDS. Geplante Ereignisse bieten Informationen über bevorstehende Änderungen an Ihren Amazon-RDS-Ressourcen. Bei einigen Ereignissen wird Ihnen empfohlen, Maßnahmen zu ergreifen, um Unterbrechungen des Dienstes zu vermeiden. Andere Ereignisse treten automatisch auf, ohne dass Sie etwas tun müssen. Ihre Ressource ist während der geplanten Änderungsaktivität möglicherweise vorübergehend nicht verfügbar. Diese Regel erstellt beispielsweise ein OpsItem für die folgenden Ereignisse: `AWS_RDS_SYSTEM_UPGRADE_SCHEDULED` und `AWS_RDS_MAINTENANCE_SCHEDULED` . Weitere Informationen zu geplanten Ereignissen finden Sie im AWS Health -Benutzerhandbuch unter Ereignistypkategorien.

Regelname	Beschreibung
SSMOpsItems-SSM-maintenance-window-execution-failed	Diese Regel erstellt OpsItems, wenn die Verarbeitung des Systems-Manager-Wartungsfensters fehlschlägt.
SSMOpsItems-SSM-maintenance-window-execution-timedout	Diese Regel erstellt OpsItems, wenn beim Start des Systems-Manager-Wartungsfensters eine Zeitüberschreitung auftritt.

Einrichten von OpsCenter

Führen Sie die folgenden Schritte aus, um OpsCenter einzurichten.

So konfigurieren Sie OpsCenter:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie auf der OpsCenter-Startseite Erste Schritte aus.
4. Wählen Sie auf der OpsCenter Einrichtungsseite die Option Diese Option aktivieren aus, damit Explorer konfigurierte AWS Config und von Amazon automatisch CloudWatch erstellte Ereignisse auf der OpsItems Grundlage häufig verwendeter Regeln und Ereignisse erstellt werden. Wenn Sie diese Option nicht auswählen, bleibt OpsCenter deaktiviert.

   > **Note**
   >
   > Amazon EventBridge (ehemals Amazon CloudWatch Events) bietet alle Funktionen von CloudWatch Events und einige neue Funktionen, wie benutzerdefinierte Event-Busse, Eventquellen von Drittanbietern und Schemaregistrierung.

5. Wählen Sie Enable (Aktivieren)OpsCenter aus.

Nachdem Sie OpsCenter aktiviert haben, können Sie Folgendes in den Einstellungen vornehmen:

- Mit der Schaltfläche „ CloudWatch Konsole öffnen“ können Sie CloudWatch Alarme erstellen. Weitere Informationen finden Sie unter Konfigurieren von CloudWatch-Alarmen zum Erstellen von OpsItems.
- Aktivieren Sie betriebliche Einblicke. Weitere Informationen finden Sie unter Analyse betrieblicher Einblicke zur Reduzierung von OpsItems.
- Aktivieren Sie Alarme für AWS Security Hub Ergebnisse. Weitere Informationen finden Sie unter AWS Security Hub.

Inhalt

- (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems
- (Optional) Einrichten von Amazon SNS für den Empfang von Benachrichtigungen zu OpsItems

(Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems

Sie können den OpsCenter-Systemmanager verwenden, um OpsItems über mehrere AWS-Konten in einer ausgewählten AWS-Region zentral zu verwalten. Dieses Feature ist verfügbar, nachdem Sie Ihre Organisation in AWS Organizations. eingerichtet haben. AWS Organizations ist ein Kontoverwaltungsservice, mit dem Sie mehrere AWS-Konten in einer von Ihnen erstellten und zentral verwalteten Organisation konsolidieren können. AWS Organizations umfasst Funktionen zur Kontoverwaltung und konsolidierten Rechnungsstellung, mit denen Sie die Budget-, Sicherheits- und Compliance-Anforderungen Ihres Unternehmens besser erfüllen können. Weitere Informationen finden Sie unter Was ist AWS Organizations? im AWS Organizations-Benutzerhandbuch

Benutzer, die zum AWS Organizations-Verwaltungskonto gehören, können ein delegiertes Administratorkonto für Systems Manager einrichten. Im Kontext von OpsCenter können delegierte Administratoren OpsItems in Mitgliederkonten erstellen, bearbeiten und anzeigen. Der delegierte Administrator kann auch Systems-Manager-Automatisierungs-Runbooks verwenden, um in großem Umfang Probleme mit OpsItems zu beheben oder Probleme mit AWS-Ressourcen zu beheben, die OpsItems erzeugen.

> **Note**
>
> Sie können nur ein Konto als delegierten Administrator für Systems Manager zuweisen. Weitere Informationen finden Sie unter Einen AWS Organizations delegierten Administrator für Systems Manager erstellen.

Systems Manager bietet die folgenden Methoden für die Einrichtung von OpsCenter, um OpsItems zentral über mehrere AWS-Konten zu verwalten.

- Quick Setup: Quick Setup, eine Funktion von Systems Manager, vereinfacht die Einrichtungs- und Konfigurationsaufgaben für Systems-Manager-Funktionen. Weitere Informationen finden Sie unter AWS Systems Manager Quick Setup.

  Quick Setup für OpsCenter hilft Ihnen bei der Ausführung der folgenden Aufgaben für die kontenübergreifende Verwaltung von OpsItems:

  - Ein Konto als delegierter Administrator registrieren (wenn der delegierte Administrator nicht bereits benannt wurde)
  - Erstellung der erforderlichen AWS Identity and Access Management (IAM)-Richtlinien und -Rollen
  - Angeben einer AWS Organizations-Organisation oder -Organisationseinheiten (OUs), in denen ein delegierter Administrator OpsItems kontenübergreifend verwalten kann

  Weitere Informationen finden Sie unter (Optional) Konfigurieren Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup.

  > **Note**
  >
  > Quick Setup ist nicht in allen AWS-Regionen verfügbar, in denen Systems Manager derzeit verfügbar ist. Wenn Quick Setup in einer Region, in der Sie OpsCenter für die zentrale Verwaltung von OpsItems über mehrere Konten hinweg konfigurieren möchten, nicht verfügbar ist, müssen Sie die manuelle Methode verwenden. Eine Liste der AWS-Regionen, in denen Quick Setup verfügbar ist, finden Sie unter Verfügbarkeit von Quick Setup in AWS-Regionen.

- Manuelle Einrichtung: Wenn Quick Setup in der Region, in der Sie OpsCenter für die zentrale Verwaltung von OpsItems über Konten hinweg konfigurieren möchten, nicht verfügbar ist, können

Sie dafür den manuellen Vorgang verwenden. Weitere Informationen finden Sie unter (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

(Optional) Konfigurieren Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup

Quick Setup, eine Funktion von AWS Systems Manager, vereinfacht Einrichtungs- und Konfigurationsaufgaben für Systems Manager Manager-Funktionen. Quick Setupfor OpsCenter unterstützt Sie bei der Ausführung der folgenden Aufgaben für die OpsItems kontenübergreifende Verwaltung:

- Angeben des delegierten Administratorkontos
- Erstellung der erforderlichen AWS Identity and Access Management (IAM-) Richtlinien und Rollen
- Angabe einer AWS Organizations Organisation oder einer Teilmenge von Mitgliedskonten, die ein delegierter Administrator kontenübergreifend verwalten kann OpsItems

Wenn Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup konfigurieren, erstellt Quick Setup die folgenden Ressourcen in den angegebenen Konten. Diese Ressourcen gewähren den angegebenen Konten die Erlaubnis, mit Automation-Runbooks zu arbeiten OpsItems und diese zu verwenden, um Probleme bei AWS der Generierung von Ressourcen zu beheben. OpsItems

Ressourcen	Konten
`AWSServiceRoleForAmazonSSM_AccountDiscovery` AWS Identity and Access Management (IAM) serviceverknüpfte Rolle Weitere Informationen über diese Rolle finden Sie unter Verwenden von Rollen zum Sammeln von AWS-Konto Informationen für OpsCenter und Explorer.	AWS Organizations Verwaltungskonto und delegiertes Administratorkonto
`OpsItem-CrossAccountManagementRole` -IAM-Rolle	Delegiertes Administratorkonto

Ressourcen	Konten
`AWS-SystemsManager-AutomationAdministrationRole` -IAM-Rolle	
`OpsItem-CrossAccountExecutionRole` -IAM-Rolle `AWS-SystemsManager-AutomationExecutionRole` -IAM-Rolle `AWS::SSM::ResourcePolicy` Systems-Manage-Ressourcenrichtlinie für die standardmäßige OpsItem-Gruppe (`OpsItemGroup`)	Alle AWS Organizations Mitgliedskonten

> **Note**
>
> Wenn Sie zuvor für die manuelle Verwaltung OpsItems von Konten konfiguriert OpsCenter haben, müssen Sie die in den Schritten 4 und 5 dieses Vorgangs erstellten AWS CloudFormation Stapel oder Stack-Sets löschen. Wenn diese Ressourcen in Ihrem Konto vorhanden sind, wenn Sie das folgende Verfahren ausführen, kann Quick Setup die kontenübergreifende Verwaltung von OpsItem nicht ordnungsgemäß konfigurieren.

So konfigurieren Sie OpsCenter, um OpsItems kontenübergreifend mithilfe von Quick Setup zu verwalten

1. Melden Sie sich AWS Management Console mit dem AWS Organizations Verwaltungskonto bei an.
2. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Quick Setup aus.
4. Wählen Sie die Registerkarte Bibliothek.
5. Scrollen Sie nach unten und finden Sie die OpsCenter-Konfigurationskachel. Wählen Sie Erstellen.

6. Geben Sie auf der Seite Quick Setup OpsCenter im Bereich Delegierter Administrator eine Konto-ID ein. Wenn Sie dieses Feld nicht bearbeiten können, wurde bereits ein delegiertes Administratorkonto für Systems Manager angegeben.
7. Wählen Sie im Abschnitt Targets (Ziele) eine Option aus. Wenn Sie Benutzerdefiniert wählen, wählen Sie die Organisationseinheiten (OU) aus, die OpsItems kontenübergreifend verwalten sollen.
8. Wählen Sie Erstellen.

Quick Setup erstellt die OpsCenter-Konfiguration und stellt die erforderlichen AWS -Ressourcen für die angegebenen Organisationseinheiten bereit.

> **Note**
>
> Wenn Sie OpsItems nicht über mehrere Konten hinweg verwalten möchten, können Sie die Konfiguration von Quick Setup löschen. Wenn Sie die Konfiguration löschen, löscht Quick Setup die folgenden IAM-Richtlinien und -Rollen, die bei der ursprünglichen Bereitstellung der Konfiguration erstellt wurden:
>
> - `OpsItem-CrossAccountManagementRole` aus dem delegierten Administratorkonto
> - `OpsItem-CrossAccountExecutionRole` und `SSM::ResourcePolicy` aus allen Organizations-Mitgliedskonten
>
> Quick Setup entfernt die Konfiguration aus allen Organisationseinheiten und AWS-Regionen , in denen die Konfiguration ursprünglich bereitgestellt wurde.

Behebung von Problemen mit einer Quick Setup-Konfiguration für OpsCenter

Dieser Abschnitt enthält Informationen zur Behebung von Problemen bei der Konfiguration der kontenübergreifenden OpsItem-Verwaltung mithilfe von Quick Setup.

Themen

- Die Bereitstellung für folgende Geräte StackSets ist fehlgeschlagen: DelegatedAdmin
- Quick Setup-Konfigurationsstatus zeigt Fehlgeschlagen

Die Bereitstellung für folgende Geräte StackSets ist fehlgeschlagen: DelegatedAdmin

Wenn Sie eine OpsCenter-Konfiguration erstellen, stellt Quick Setup zwei AWS CloudFormation -Stack-Sets im Verwaltungskonto Organizations bereit. Die Stack-Sets verwenden das folgende Präfix: `AWS-QuickSetup-SSMOpsCenter`. Wenn Quick Setup den folgenden Fehler anzeigt: `Deployment to these StackSets failed: delegatedAdmin`, gehen Sie wie folgt vor, um dieses Problem zu beheben.

Um einen failed:DelegatedAdmin-Fehler StackSets zu beheben

1. Wenn Sie den `Deployment to these StackSets failed: delegatedAdmin` Fehler in einem roten Banner in der Quick Setup Konsole erhalten haben, melden Sie sich mit dem delegierten Administratorkonto und der AWS-Region als Heimatregion angegebenen Region an. Quick Setup
2. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie den Stack aus, der von der Quick Setup-Konfiguration erstellt wurde. Der Stack-Name beinhaltet Folgendes: AWS- QuickSetup -SSM OpsCenter.

   > **Note**
   >
   > CloudFormation Löscht manchmal fehlgeschlagene Stack-Bereitstellungen. Wenn der Stack in der Tabelle Stacks nicht verfügbar ist, wählen Sie Deleted (Gelöscht) in der Filterliste aus.

4. Zeigen Sie den Status und den Status reason (Statusgrund) an. Weitere Informationen zum Stack-Status finden Sie unter Stack-Statuscodes im Benutzerhandbuch von AWS CloudFormation .
5. Um nachzuvollziehen, welcher Schritt genau fehlgeschlagen ist, sehen Sie sich auf der Registerkarte Events (Ereignisse) den Status der einzelnen Ereignisse an. Weitere Informationen finden Sie unter Fehlerbehebung im AWS CloudFormation -Benutzerhandbuch.

> **Note**
>
> Wenn Sie den Bereitstellungsfehler nicht mithilfe der Schritte CloudFormation zur Fehlerbehebung beheben können, löschen Sie die Konfiguration und versuchen Sie es erneut.

Quick Setup-Konfigurationsstatus zeigt Fehlgeschlagen

Wenn in der Tabelle mit den Konfigurationsdetails auf der Seite mit den Konfigurationsdetails der Konfigurationsstatus angezeigt wird`Failed`, melden Sie sich in der AWS-Konto Region an, in der der Fehler aufgetreten ist.

Einen Quick Setup-Fehler beim Erstellen einer OpsCenter-Konfiguration beheben

1. Melden Sie sich bei der AWS-Konto und in der Region AWS-Region an, in der der Fehler aufgetreten ist.
2. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie den Stack aus, der von der Quick Setup-Konfiguration erstellt wurde. Der Stack-Name beinhaltet Folgendes: AWS- QuickSetup -SSM OpsCenter.

   > **Note**
   >
   > CloudFormation Löscht manchmal fehlgeschlagene Stack-Bereitstellungen. Wenn der Stack in der Tabelle Stacks nicht verfügbar ist, wählen Sie Deleted (Gelöscht) in der Filterliste aus.

4. Zeigen Sie den Status und den Status reason (Statusgrund) an. Weitere Informationen zum Stack-Status finden Sie unter Stack-Statuscodes im Benutzerhandbuch von AWS CloudFormation .
5. Um nachzuvollziehen, welcher Schritt genau fehlgeschlagen ist, sehen Sie sich auf der Registerkarte Events (Ereignisse) den Status der einzelnen Ereignisse an. Weitere Informationen finden Sie unter Fehlerbehebung im AWS CloudFormation -Benutzerhandbuch.

Die Konfiguration des Mitgliedskontos zeigt ResourcePolicyLimitExceededException

Wenn der Stack-Status `ResourcePolicyLimitExceededException` anzeigt, wurde das Konto zuvor über die manuelle Methode in die kontoübergreifende OpsCenter-Verwaltung aufgenommen. Um dieses Problem zu beheben, müssen Sie die AWS CloudFormation Stacks oder Stack-Sets löschen, die Sie in den Schritten 4 und 5 des manuellen Onboarding-Prozesses erstellt haben. Weitere Informationen finden Sie unter Löschen eines Stack-Sets und Löschen eines Stacks auf der AWS CloudFormation Konsole im AWS CloudFormation Benutzerhandbuch.

(Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems

In diesem Abschnitt wird beschrieben, wie Sie OpsCenter manuell für die kontoübergreifende Verwaltung von OpsItem konfigurieren. Dieser Prozess wird zwar weiterhin unterstützt, wurde jedoch durch einen neueren Prozess ersetzt, der Systems Manager Quick Setup verwendet. Weitere Informationen finden Sie unter (Optional) Konfigurieren Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup.

Sie können ein zentrales Konto einrichten, um manuelle OpsItems für Mitgliedskonten zu erstellen und diese OpsItems zu verwalten und zu beheben. Das zentrale Konto kann das AWS Organizations Verwaltungskonto oder sowohl das AWS Organizations Verwaltungskonto als auch das delegierte Administratorkonto von Systems Manager sein. Wir empfehlen, dass Sie das delegierte Administratorkonto von Systems Manager als zentrales Konto verwenden. Sie können dieses Feature erst verwenden, nachdem Sie AWS Organizations konfiguriert haben.

Mit AWS Organizations können Sie mehrere zu einer Organisation AWS-Konten zusammenfassen, die Sie zentral erstellen und verwalten. Der Benutzer des zentralen Kontos kann OpsItems für alle ausgewählten Mitgliedskonten gleichzeitig Konten erstellen und diese OpsItems verwalten.

Verwenden Sie den Prozess in diesem Abschnitt, um den Systems Manager Manager-Dienstprinzipal in Organizations zu aktivieren und AWS Identity and Access Management (IAM) -Berechtigungen für die OpsItems kontenübergreifende Arbeit zu konfigurieren.

Themen

- Bevor Sie beginnen
- Schritt 1: Erstellen einer Ressourcen-Datensynchronisierung
- Schritt 2: Aktivieren des Systems Manager Manager-Dienstprinzipals in AWS Organizations
- Schritt 3: Erstellen der AWSServiceRoleForAmazonSSM_AccountDiscovery-serviceverknüpften Rolle
- Schritt 4: Konfigurieren von Berechtigungen für die kontenübergreifende Arbeit mit OpsItems

- Schritt 5: Konfigurieren von Berechtigungen für das kontenübergreifende Arbeiten mit zugehörigen Ressourcen

> **Note**
>
> Nur OpsItems vom Typ `/aws/issue` werden bei der kontenübergreifenden Arbeit in OpsCenter unterstützt.

Bevor Sie beginnen

Bevor Sie OpsCenter für die Arbeit mit OpsItems kontenübergreifend einrichten, stellen Sie sicher, dass Sie Folgendes eingerichtet haben:

- Ein delegiertes Administratorkonto für Systems Manager. Weitere Informationen finden Sie unter Konfigurierung eines delegierten Administrators.
- Eine Organisation, die in Organizations eingerichtet und konfiguriert wurde. Weitere Informationen finden Sie unter Erstellen und Verwalten einer Organisation im AWS Organizations - Benutzerhandbuch.
- Sie haben Systems Manager Automation so konfiguriert, dass Automatisierungs-Runbooks für mehrere AWS-Regionen AWS Konten ausgeführt werden. Weitere Informationen finden Sie unter Ausführen von Automatisierungen in mehreren AWS-Regionen-Regionen und -Konten.

Schritt 1: Erstellen einer Ressourcen-Datensynchronisierung

Nach der Einrichtung und Konfiguration können Sie die Daten OpsCenter für eine gesamte Organisation zusammenfassen AWS OrganizationsOpsItems, indem Sie eine Ressourcendatensynchronisierung erstellen. Weitere Informationen finden Sie unter Erstellen einer Ressourcendatensynchronisierung. Achten Sie beim Erstellen der Synchronisierung darauf, im Abschnitt Konten hinzufügen die Option Alle Konten aus meiner AWS Organizations Konfiguration einbeziehen auszuwählen.

Schritt 2: Aktivieren des Systems Manager Manager-Dienstprinzipals in AWS Organizations

Damit ein Benutzer OpsItems kontenübergreifend arbeiten kann, muss der Systems Manager Manager-Dienstprinzipal aktiviert sein AWS Organizations. Wenn Sie Systems Manager zuvor mit anderen Funktionen für Szenarios mit mehreren Konten konfiguriert haben, ist der Systems-

Manager-Service-Prinzipal möglicherweise bereits in Organizations konfiguriert. Führen Sie zur Überprüfung die folgenden Befehle von AWS Command Line Interface (AWS CLI) aus. Wenn Sie Systems Manager nicht für andere Szenarien mit mehreren Konten konfiguriert haben, fahren Sie mit dem nächsten Schritt fort: So aktivieren Sie den Systems-Manager-Service-Prinzipal in AWS Organizations.

So überprüfen Sie, ob der Systems Manager Manager-Dienstprinzipal aktiviert ist in AWS Organizations

1. Laden Sie die neueste Version von AWS CLI auf Ihren lokalen Computer herunter.
2. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen und eine anzugeben AWS-Region.

   ```
   aws configure
   ```

   Sie werden aufgefordert, Folgendes anzugeben: Ersetzen Sie im folgenden Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

   ```
   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. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Systems-Manager-Service-Prinzipal für AWS Organizations aktiviert ist.

   ```
   aws organizations list-aws-service-access-for-organization
   ```

   Der Befehl gibt ähnliche Informationen wie im folgenden Beispiel zurück.

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

So aktivieren Sie den Systems Manager Manager-Dienstprinzipal in AWS Organizations

Wenn Sie den Systems-Manager-Service-Prinzipal für Organizations noch nicht konfiguriert haben, verwenden Sie dazu das folgende Verfahren. Weitere Informationen zu diesem Befehl finden Sie enable-aws-service-accessin der AWS CLI Befehlsreferenz.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben. Informationen finden Sie unter Installation der CLI und Konfiguration der CLI.
2. Laden Sie die neueste Version von AWS CLI auf Ihren lokalen Computer herunter.
3. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen und eine anzugeben AWS-Region.

   ```
   aws configure
   ```

   Sie werden aufgefordert, Folgendes anzugeben: Ersetzen Sie im folgenden Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

   ```
   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. Führen Sie den folgenden Befehl aus, um den Systems-Manager-Service-Prinzipal für AWS Organizations zu aktivieren.

   ```
   aws organizations enable-aws-service-access --service-principal "ssm.amazonaws.com"
   ```

Schritt 3: Erstellen der **AWSServiceRoleForAmazonSSM_AccountDiscovery**-serviceverknüpften Rolle

Eine dienstbezogene Rolle wie die `AWSServiceRoleForAmazonSSM_AccountDiscovery` Rolle ist ein einzigartiger Typ von IAM-Rolle, die direkt mit einer verknüpft ist AWS-Service, z. B. Systems Manager. Mit Diensten verknüpfte Rollen sind vom Dienst vordefiniert und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS-Services in Ihrem Namen anzurufen. Weitere Informationen zur serviceverknüpften `AWSServiceRoleForAmazonSSM_AccountDiscovery`-Rolle finden Sie unter Berechtigungen von serviceverknüpften Rollen für Systems Manager-Kontoerkennung.

Verwenden Sie das folgende Verfahren, um die serviceverknüpfte `AWSServiceRoleForAmazonSSM_AccountDiscovery`-Rolle mithilfe der AWS CLI zu erstellen. Weitere Informationen zu dem in diesem Verfahren verwendeten Befehl finden Sie create-service-linked-rolein der AWS CLI Befehlsreferenz.

So erstellen Sie die serviceverknüpfte **AWSServiceRoleForAmazonSSM_AccountDiscovery**-Rolle

1. Melden Sie sich beim AWS Organizations Verwaltungskonto an.
2. Führen Sie den folgenden Befehl aus, während Sie mit dem Organizations-Verwaltungskonto angemeldet sind.

```
aws iam create-service-linked-role \
    --aws-service-name accountdiscovery.ssm.amazonaws.com \
    --description "Systems Manager account discovery for AWS Organizations service-
linked role"
```

Schritt 4: Konfigurieren von Berechtigungen für die kontenübergreifende Arbeit mit OpsItems

Verwenden Sie AWS CloudFormation Stacksets, um eine `OpsItemGroup` Ressourcenrichtlinie und eine IAM-Ausführungsrolle zu erstellen, mit OpsItems denen Benutzer kontenübergreifend arbeiten können. Laden Sie zunächst die `OpsCenterCrossAccountMembers.zip`-Datei herunter und entpacken Sie sie. Diese Datei enthält die Vorlagendatei. `OpsCenterCrossAccountMembers.yaml` AWS CloudFormation Wenn Sie mithilfe dieser Vorlage ein Stack-Set erstellen, CloudFormation werden automatisch die `OpsItemCrossAccountResourcePolicy` Ressourcenrichtlinie und die `OpsItemCrossAccountExecutionRole` Ausführungsrolle im Konto erstellt. Weitere Informationen

zum Erstellen eines Stack-Sets finden Sie unter Erstellen eines Stack-Sets im AWS CloudFormation - Benutzerhandbuch.

> **Important**
>
> Berücksichtigen Sie für diese Aufgabe die folgenden wichtigen Informationen:
>
> - Sie müssen das Stackset bereitstellen, während Sie beim AWS Organizations - Verwaltungskonto angemeldet sind.
> - Sie müssen dieses Verfahren wiederholen, während Sie bei jedem Konto angemeldet sind, das Sie für die Arbeit mit OpsItems kontenübergreifend festlegen möchten, einschließlich des delegierten Administratorkontos.
> - Wenn Sie die kontoübergreifende OpsItems Verwaltung in verschiedenen Bereichen aktivieren möchten AWS-Regionen, wählen Sie im Abschnitt Regionen angeben der Vorlage die Option Alle Regionen hinzufügen aus. Die kontoübergreifende OpsItem-Verwaltung wird für Opt-in-Regionen nicht unterstützt.

Schritt 5: Konfigurieren von Berechtigungen für das kontenübergreifende Arbeiten mit zugehörigen Ressourcen

Eine OpsItem kann detaillierte Informationen über betroffene Ressourcen wie Amazon Elastic Compute Cloud (Amazon EC2)-Instances oder Amazon Simple Storage Service (Amazon S3)-Buckets enthalten. Die `OpsItemCrossAccountExecutionRole`-Ausführungsrolle, die Sie im vorherigen Schritt 4 erstellt haben, stellt OpsCenter mit schreibgeschützten Berechtigungen für Mitgliedskonten zum Anzeigen zugehöriger Ressourcen bereit. Sie müssen auch eine IAM-Rolle erstellen, um Verwaltungskonten die Berechtigung zum Anzeigen und Interagieren mit verwandten Ressourcen zu gewähren. Dies werden Sie in dieser Aufgabe durchführen.

Laden Sie zunächst die `OpsCenterCrossAccountManagementRole.zip`-Datei herunter und entpacken Sie sie. Diese Datei enthält die `OpsCenterCrossAccountManagementRole.yaml` AWS CloudFormation Vorlagendatei. Wenn Sie mithilfe dieser Vorlage einen Stack erstellen, CloudFormation wird automatisch die `OpsCenterCrossAccountManagementRole` IAM-Rolle im Konto erstellt. Weitere Informationen zum Erstellen eines Stacks finden Sie im AWS CloudFormation Benutzerhandbuch unter Erstellen eines Stacks auf der AWS CloudFormation Konsole.

Important

Berücksichtigen Sie für diese Aufgabe die folgenden wichtigen Informationen:

- Wenn Sie beabsichtigen, ein Konto als delegierter Administrator für anzugebenOpsCenter, müssen Sie dies AWS-Konto bei der Erstellung des Stacks angeben.
- Sie müssen dieses Verfahren ausführen, während Sie beim AWS Organizations -Verwaltungskonto angemeldet sind, und erneut, während Sie beim delegierten Administratorkonto angemeldet sind.

(Optional) Einrichten von Amazon SNS für den Empfang von Benachrichtigungen zu OpsItems

Sie können OpsCenter so konfigurieren, dass Benachrichtigungen an ein Amazon Simple Notification Service (Amazon SNS)-Thema gesendet werden, wenn das System eine neues OpsItem erstellt oder ein vorhandenes OpsItem aktualisiert.

Führen Sie die folgenden Schritte aus, um Benachrichtigungen für OpsItems zu erhalten.

- Schritt 1: Erstellen und Abonnieren eines Amazon-SNS-Themas
- Schritt 2: Aktualisieren der Amazon SNS-Zugriffsrichtlinie
- Schritt 3: Aktualisieren der AWS KMS -Zugriffsrichtlinie

  Note

  Wenn Sie in Schritt 2 die serverseitige Verschlüsselung AWS Key Management Service (AWS KMS) aktivieren, müssen Sie Schritt 3 abschließen. Andernfalls können Sie Schritt 3 überspringen.

- Schritt 4: Aktivieren Sie OpsItems-Standardregeln zum Senden von Benachrichtigungen für neue OpsItems

Schritt 1: Erstellen und Abonnieren eines Amazon-SNS-Themas

Um Benachrichtigungen zu erhalten, müssen Sie ein Amazon SNS-Thema erstellen und abonnieren. Weitere Informationen finden Sie unter Erstellen eines Amazon-SNS-Themas und Abonnieren eines Amazon-SNS-Themas im Entwicklerhandbuch für Amazon Simple Notification Service.

> **Note**
>
> Wenn Sie mehrere Konten verwenden, müssen Sie OpsCenter in jeder Region AWS-Regionen oder jedem Konto, für das Sie OpsItem Benachrichtigungen erhalten möchten, ein Amazon SNS SNS-Thema erstellen und abonnieren.

Schritt 2: Aktualisieren der Amazon SNS-Zugriffsrichtlinie

Sie müssen ein Amazon-SNS-Thema mit OpsItems verknüpfen. Gehen Sie wie folgt vor, um eine Amazon-SNS-Zugriffsrichtlinie einzurichten, damit OpsItems-Benachrichtigungen von Systems Manager für das Amazon-SNS-Thema veröffentlichen kann, das Sie in Schritt 1 erstellt haben.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon SNS SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home.
2. Wählen Sie im Navigationsbereich Themen aus.
3. Wählen Sie das Thema aus, das Sie in Schritt 1 erstellt haben, und klicken Sie dann auf Bearbeiten.
4. Erweitern Sie die Option Zugriffsrichtlinie.
5. Fügen Sie der vorhandenen Richtlinie den folgenden `Sid`-Block hinzu. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

```
{
      "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**
>
> Der `aws:SourceAccount` globale Bedingungsschlüssel schützt vor dem Szenario Confused Deputy. Um diesen Bedingungsschlüssel zu verwenden, setzen Sie den Wert auf die Konto-ID des OpsItem-Eigentümers. Weitere Informationen finden Sie unter Confused Deputy im IAM-Benutzerhandbuch.

6. Wählen Sie Änderungen speichern aus.

Das System sendet jetzt Benachrichtigungen an das Amazon SNS-Thema, wenn OpsItems erstellt oder aktualisiert werden.

> **Important**
>
> Wenn Sie das Amazon SNS SNS-Thema in Schritt 2 mit einem AWS Key Management Service (AWS KMS) serverseitigen Verschlüsselungsschlüssel konfigurieren, führen Sie Schritt 3 aus. Andernfalls können Sie Schritt 3 überspringen.

Schritt 3: Aktualisieren der AWS KMS -Zugriffsrichtlinie

Wenn Sie die AWS KMS serverseitige Verschlüsselung für Ihr Amazon SNS SNS-Thema aktiviert haben, müssen Sie auch die Zugriffsrichtlinie aktualisieren AWS KMS key , die Sie bei der Konfiguration des Themas ausgewählt haben. Gehen Sie wie folgt vor, um die Zugriffsrichtlinie zu aktualisieren, damit Systems Manager OpsItem-Benachrichtigungen für das Amazon-SNS-Thema veröffentlichen kann, das Sie in Schritt 1 erstellt haben.

> **Note**
>
> OpsCenter unterstützt keine Veröffentlichung von OpsItems in einem Amazon-SNS-Thema, das mit einem Von AWS verwalteter Schlüssel konfiguriert ist.

1. Öffnen Sie die AWS KMS Konsole unter https://console.aws.amazon.com/kms.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.

3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Wählen Sie die ID des KMS-Schlüssels aus, den Sie bei der Erstellung des Themas ausgewählt haben.
5. Wählen Sie im Abschnitt Key policy (Schlüsselrichtlinie) die Option Switch to policy view (Zur Richtlinienansicht wechseln) aus.
6. Wählen Sie Bearbeiten aus.
7. Fügen Sie der vorhandenen Richtlinie den folgenden `Sid`-Block hinzu. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

```
{
      "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"
    }
```

Im folgenden Beispiel wird der neue Block in Zeile 14 eingegeben.

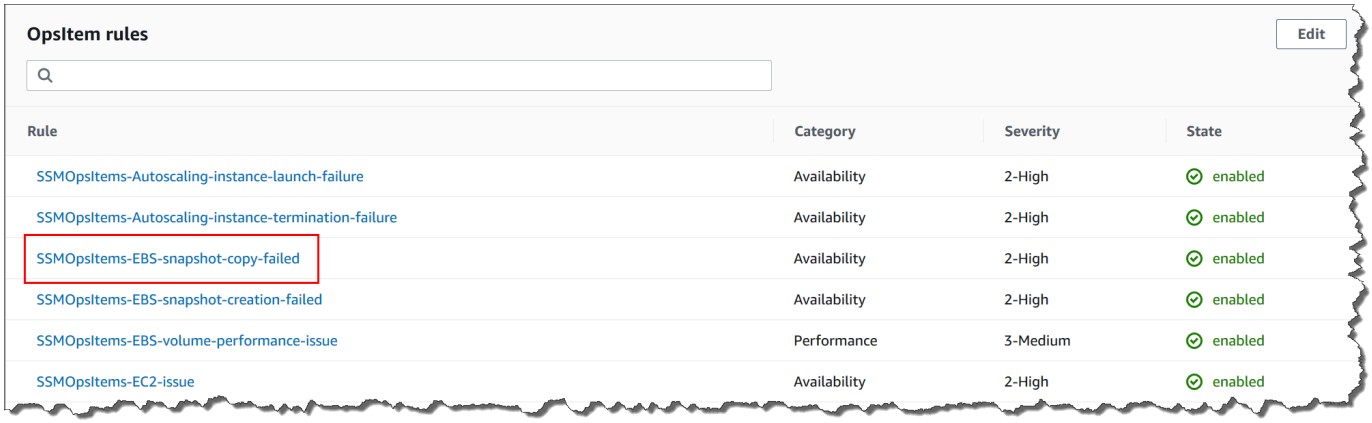
8. Wählen Sie Änderungen speichern aus.

Schritt 4: Aktivieren Sie OpsItems-Standardregeln zum Senden von Benachrichtigungen für neue OpsItems

OpsItemsStandardregeln in Amazon EventBridge sind nicht mit einem Amazon-Ressourcennamen (ARN) für Amazon SNS-Benachrichtigungen konfiguriert. Gehen Sie wie folgt vor, um eine Regel zu bearbeiten EventBridge und einen `notifications` Block einzugeben.

So fügen Sie einer OpsItem-Standardregel einen Benachrichtigungsblock hinzu

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie die Registerkarte OpsItems und dann Configure sources (Quellen konfigurieren) aus.
4. Wählen Sie den Namen der Quellregel aus, die Sie mit einem `notifications`-Block konfigurieren möchten, wie im folgenden Beispiel gezeigt.



   Die Regel wird in Amazon geöffnet EventBridge.
5. Wählen Sie auf der Regeldetailseite auf der Registerkarte Targets (Ziele) die Option Edit (Bearbeiten) aus.
6. Wählen Sie im Bereich Additional settings (Zusätzliche Einstellungen) die Option Configure input transformer (Eingabetransformator konfigurieren).
7. Fügen Sie im Feld Vorlage einen `notifications`-Block im folgenden Format hinzu.

```
"notifications":[{"arn":"arn:aws:sns:region:account ID:topic name"}],
```

   Ein Beispiel:

```
"notifications":[{"arn":"arn:aws:sns:us-west-2:1234567890:MySNSTopic"}],
```

Geben Sie den Benachrichtigungsblock vor dem `resources` Block ein, wie im folgenden Beispiel für die Region USA West (Oregon) (us-west-2) gezeigt.

```
{
    "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. Wählen Sie Bestätigen aus.

9. Wählen Sie Weiter.
10. Wählen Sie Weiter.
11. Wählen Sie Regel aktualisieren aus.

Wenn das System das nächste Mal ein OpsItem für die Standardregel erstellt, wird eine Benachrichtigung an das Amazon-SNS-Thema veröffentlicht.

Integrieren von OpsCenter in anderen AWS-Services

OpsCenter, eine Fähigkeit von AWS Systems Manager, lässt sich in mehrere integrieren, AWS-Services um Probleme mit AWS Ressourcen zu diagnostizieren und zu beheben. Sie müssen das AWS-Service einrichten, bevor Sie es mit OpsCenter integrieren.

Standardmäßig AWS-Services sind die folgenden Komponenten integriert OpsCenter und können OpsItems automatisch erstellt werden:

- Amazon CloudWatch
- Einblicke in CloudWatch Amazon-Anwendungen
- Amazon EventBridge
- AWS Config
- AWS Systems Manager Incident Manager

Sie müssen die folgenden Services mit OpsCenter integrieren, um OpsItems automatisch zu erstellen:

- DevOpsAmazon-Guru
- AWS Security Hub

Wenn einer dieser Services ein OpsItem erstellt, können Sie das OpsItem von OpsCenter verwalten und beheben. Weitere Informationen finden Sie unter Verwalten von OpsItems und Beheben von OpsItem-Problemen.

Weitere Informationen zu den einzelnen AWS-Service Optionen und deren Integration OpsCenter finden Sie in den folgenden Themen.

Themen

- Amazon CloudWatch
- Einblicke in CloudWatch Amazon-Anwendungen
- DevOpsAmazon-Guru
- Amazon EventBridge
- AWS Config
- AWS Security Hub
- Incident Manager

Amazon CloudWatch

Amazon CloudWatch überwacht Ihre AWS Ressourcen und Services und zeigt Kennzahlen zu allen Ressourcen an AWS-Service , die Sie nutzen. CloudWatch erzeugt eineOpsItem, wenn ein Alarm in den Alarmzustand übergeht. Sie können beispielsweise einen Alarm konfigurieren, um automatisch eine OpsItem zu erstellen, wenn ein Anstieg der HTTP-Fehler von Ihrem Application Load Balancer generiert wird.

In der folgenden Liste OpsItems sind einige Alarme aufgeführt CloudWatch , die Sie so konfigurieren können, dass sie erstellt werden:

- Amazon DynamoDB: Lese- und Schreibaktionen in der Datenbank erreichen einen Schwellenwert
- Amazon EC2: CPU-Auslastung erreicht einen Schwellenwert
- AWS Abrechnung: Die geschätzten Gebühren erreichen einen bestimmten Schwellenwert
- Amazon EC2: Eine Instance schlägt keine Statusprüfung vor
- Amazon Elastic Block Store (EBS): Die Festplattenspeichernutzung erreicht einen Schwellenwert

Sie können eine Warnung erstellen oder eine vorhandene Warnung bearbeiten, um ein OpsItem erstellen. Weitere Informationen finden Sie unter Konfigurieren von CloudWatch-Alarmen zum Erstellen von OpsItems.

Wenn Sie die OpsCenter Verwendung des integrierten Setups aktivieren, wird es in CloudWatch integriertOpsCenter.

Einblicke in CloudWatch Amazon-Anwendungen

Mit Amazon CloudWatch Application Insights können Sie die am besten geeigneten Monitore für Ihre Anwendungsressourcen einrichten, um Daten kontinuierlich auf Anzeichen von Problemen mit

Ihren Anwendungen zu analysieren. Wenn Sie Anwendungsressourcen in CloudWatch Application Insights konfigurieren, können Sie wählen, ob das System automatisch erstellt OpsItems werden sollOpsCenter. Für jedes Problem, das in der Anwendung entdeckt wird, wird ein OpsItem auf der OpsCenter-Konsole erstellt. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch unter Einrichtung, Konfiguration und Verwaltung Ihrer Anwendung für die Überwachung.

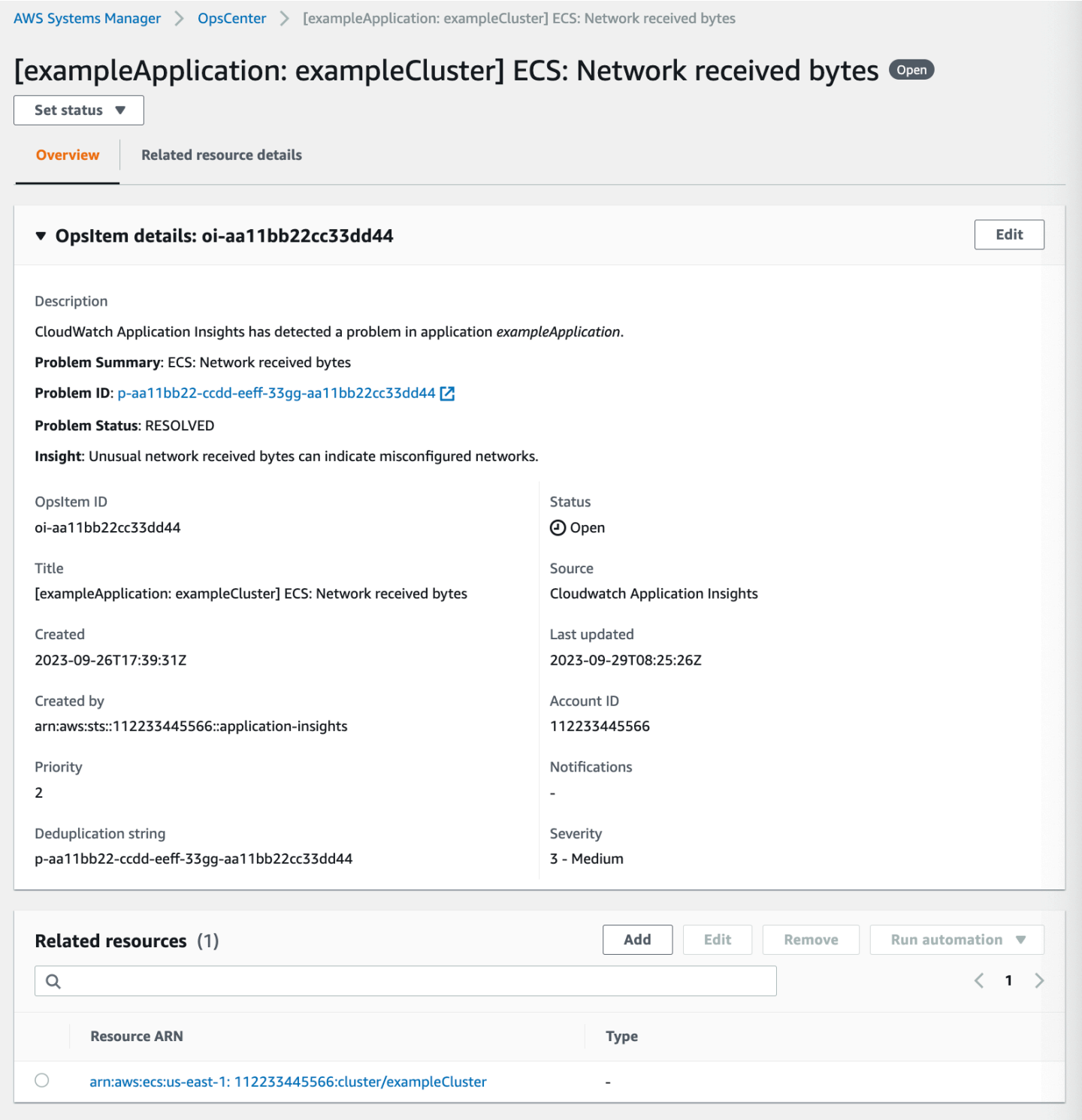
> **Note**
>
> Ab dem 16. Oktober 2023 verwenden der Titel und die Beschreibung von OpsItems Created by CloudWatch Application Insights nun das folgende verbesserte Format:

```
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>
```

Ein Beispiel:



DevOpsAmazon-Guru

Amazon DevOps Guru verwendet maschinelles Lernen, um Ihre Betriebsdaten, Anwendungsmetriken und Anwendungsereignisse zu analysieren und Verhaltensweisen zu identifizieren, die von normalen Betriebsmustern abweichen. Wenn Sie DevOps Guru die Generierung einer Eingabe OpsItem

ermöglichenOpsCenter, generiert jede Erkenntnis eine neueOpsItem. Sie können OpsCenter verwenden, um Ihre OpsItems zu verwalten.

DevOpsGuru erstellt automatischOpsItems. Sie können Amazon DevOps Guru mithilfe OpsItems vonQuick Setup, einer Funktion von Systems Manager, zum Erstellen aktivieren. Das System erstellt OpsItems mithilfe der serviceverknüpften Rolle AWSServiceRoleForDevOpsGuru AWS Identity and Access Management (IAM).

Um sich mit Guru zu integrieren OpsCenter DevOps

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Seite „ DevOpsGuru-Konfigurationsoptionen anpassen“ die Registerkarte „Bibliothek“.
4. Wählen Sie im DevOpsGuru-Bereich die Option Erstellen aus.
5. Wählen Sie für Konfigurationsoptionen die Option Aktivieren aus AWS Systems Manager OpsItems.
6. Wählen Sie nach Abschluss der Einrichtung Erstellen aus.

Amazon EventBridge

Amazon EventBridge liefert eine Reihe von Ereignissen, die Änderungen an AWS Ressourcen beschreiben. Wenn Sie die OpsCenter Verwendung von Integrated Setup aktivieren, integriert es EventBridge sich OpsCenter in die EventBridge Standardregeln und aktiviert diese. EventBridge Erstellt auf der Grundlage dieser RegelnOpsItems. Mithilfe von Regeln können Sie Ereignisse filtern und zur Untersuchung und Behebung an OpsCenter weiterleiten.

> **Note**
>
> Amazon EventBridge (ehemals Amazon CloudWatch Events) bietet alle Funktionen von CloudWatch Events und einige neue Funktionen, wie benutzerdefinierte Event-Busse, Eventquellen von Drittanbietern und Schemaregistrierung.

Im Folgenden finden Sie einige Regeln, die Sie konfigurieren können EventBridge , um eine zu erstellenOpsItem:

- Security Hub: Sicherheitswarnung ausgegeben
- Amazon DynamoDB: ein Drosselungsereignis
- Amazon Elastic Compute Cloud Auto Scaling: Instance konnte nicht gestartet werden
- Systems Manager: Fehler beim Ausführen einer Automatisierung
- AWS Health: eine Warnung für geplante Wartungsarbeiten
- Amazon EC2: Instance-Status wurde von „Ausführung“ auf „Stopp“ geändert

Basierend auf Ihren Anforderungen können Sie entweder eine Regel erstellen oder eine vorhandene Regel bearbeiten, um ein OpsItems zu erstellen. Anweisungen zum Bearbeiten einer Regel zum Erstellen eines OpsItem finden Sie unter Konfigurieren von EventBridge-Regeln zum Erstellen von OpsItems.

AWS Config

AWS Config bietet einen detaillierten Überblick über die Konfiguration der AWS Ressourcen in Ihrem AWS-Konto.

AWS Config integriert sich nicht direkt inOpsCenter. Stattdessen erstellen Sie eine AWS Config Regel, die ein Ereignis an Amazon sendet EventBridge, z. B. wenn eine nicht konforme Instance AWS Config erkannt wird. Dieses Ereignis wird dann EventBridge anhand einer von Ihnen erstellten EventBridge Regel bewertet. Wenn die Regel zutrifft, wird EventBridge das Ereignis in ein Ereignis umgewandelt OpsItem und an dieses OpsCenter als Zielziel übertragen.

Mit diesem OpsItem können Sie Details der nicht konformen Ressource nachverfolgen, Untersuchungsmaßnahmen aufzeichnen und Zugriff auf konsistente Korrekturmaßnahmen gewähren.

Verwandte Informationen

Konfigurieren von EventBridge-Regeln zum Erstellen von OpsItems

Verwendung von AWS Systems ManagerOpsCenter und AWS Config für die Überwachung der Einhaltung von Vorschriften

AWS Security Hub

AWS Security Hub sammelt Sicherheitsdaten, sogenannte Erkenntnisse, von Across AWS-Konten und Services. Mithilfe einer Reihe von Regeln zur Erkennung und Generierung von Erkenntnissen hilft Ihnen Security Hub, Sicherheitsprobleme für die von Ihnen verwalteten Ressourcen zu

identifizieren, zu priorisieren und zu beheben. Nachdem Sie die Integration, wie in diesem Thema beschrieben, konfiguriert haben, erstellt Systems Manager OpsItems für Security Hub Erkenntnisse in OpsCenter.

> **Note**
>
> OpsCenter verfügt über eine bidirektionale Integration mit Security Hub. Das heißt, wenn Sie das Feld Status oder Schweregrad für ein OpsItem im Zusammenhang mit einer Erkenntnis aktualisieren, synchronisiert das System die Änderungen mit Security Hub. Ebenso werden alle Änderungen an einem Erkenntnis automatisch in den entsprechenden OpsItems in OpsCenter aktualisiert.
>
> Wenn aus einem Security Hub-Befund ein erstellt OpsItem wird, werden Security Hub-Metadaten automatisch zum Betriebsdatenfeld von hinzugefügtOpsItem. Wenn diese Metadaten gelöscht werden, funktionieren die bidirektionalen Updates nicht mehr.

Standardmäßig erstellt Systems Manager OpsItems für kritische und hochgradig schwerwiegende Befunde. Sie können OpsCenter für die Erstellung von OpsItems für Erkenntnisse mit mittlerem und niedrigem Schweregrad manuell konfigurieren. OpsCenter erstellt kein OpsItems zu einem informativen Ergebnis, da keine Behebung erforderlich ist. Weitere Informationen zu den Schweregraden von Security Hub finden Sie unter Schweregrad in der AWS Security Hub -API-Referenz.

Bevor Sie beginnen

Bevor Sie OpsCenter konfigurieren, um OpsItems auf der Grundlage der Erkenntnisse von Security Hub zu erstellen, überprüfen Sie, ob Sie die Aufgaben zur Einrichtung von Security Hub abgeschlossen haben. Weitere Informationen finden Sie unter Setting up Security Hub (Einrichten von Security Hub) im AWS Security Hub -Benutzerhandbuch.

Wenn Sie Security Hub mit OpsCenter integrieren, erstellt das System OpsItems mithilfe der serviceverknüpften `AWSServiceRoleForSystemsManagerOpsDataSync`-IAM-Rolle. Weitere Informationen über diese Rolle finden Sie unter Verwenden von Rollen zum Erstellen OpsData und OpsItems für Explorer.

> **Warning**
>
> Berücksichtigen Sie für die OpsCenter-Integration mit Security Hub die folgenden wichtigen Informationen über die Preisgestaltung:

- Wenn Sie bei der Konfiguration von OpsCenter und der Security-Hub-Integration im Security-Hub-Administratorkonto angemeldet sind, erstellt das System OpsItems für Erkenntnisse im Administrator- und allen Mitgliedskonten. Die OpsItems werden alle im Administratorkonto erstellt. Abhängig von einer Vielzahl von Faktoren kann dies zu einer unerwartet hohen Rechnungssumme führen. AWS

  Wenn Sie bei der Konfiguration der Integration in einem Mitgliedskonto angemeldet sind, erstellt das System nur OpsItems für Erkenntnisse in diesem individuellen Konto. Weitere Informationen zum Security Hub-Administratorkonto, zu Mitgliedskonten und deren Beziehung zum EventBridge Ereignis-Feed für Ergebnisse finden Sie unter Typen der Security Hub Hub-Integration mit EventBridge im AWS Security Hub Benutzerhandbuch.
- Für jede Erkenntnis, das ein OpsItem erstellt, wird Ihnen der reguläre Preis für die Erstellung des OpsItem berechnet. Ihnen wird auch berechnet, wenn Sie das OpsItem bearbeiten oder die entsprechende Erkenntnis im Security Hub aktualisiert wird (was ein OpsItem-Update auslöst).

So konfigurieren Sie OpsCenter, um OpsItems für Security-Hub-Erkenntnisse zu erstellen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Settings (Einstellungen) aus.
4. Wählen Sie im Abschnitt Security Hub Erkenntnisse Bearbeiten.
5. Wählen Sie den Schieberegler, um Deaktiviert zu Aktiviert zu ändern.
6. Wenn Sie möchten, dass das System OpsItems für Erkenntnisse mit mittlerem oder niedrigem Schweregrad erstellt, schalten Sie diese Optionen ein.
7. Wählen Sie Save (Speichern) aus, um die Konfiguration zu speichern.

Gehen Sie wie folgt vor, wenn Sie nicht mehr möchten, dass das System OpsItems für Security-Hub-Erkenntnisse erstellt.

Erhalt von OpsItems für Security-Hub-Erkenntnisse stoppen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Settings (Einstellungen) aus.
4. Wählen Sie im Abschnitt Security Hub Erkenntnisse Bearbeiten.
5. Wählen Sie den Schieberegler, um Aktiviert zu Deaktiviert zu ändern. Wenn Sie den Schieberegler nicht umschalten können, wurde Security Hub nicht für Ihr AWS-Konto aktiviert.
6. Wählen Sie Speichern, um Ihre Konfiguration zu speichern. OpsCenter erstellt nicht mehr OpsItems auf der Grundlage von Security-Hub-Erkenntnissen.

> **Important**
>
> Ein von Systems Manager delegierter Administrator oder das AWS Organizations Verwaltungskonto können Security Hub Hub-Ergebnisse OpsCenter für mehrere Konten aktivieren und AWS-Regionen eine Ressourcendatensynchronisierung in Explorer erstellen. Wenn die Security Hub Hub-Quelle aktiviert ist Explorer und eine Ressourcendatensynchronisierung existiert, die auf das Mitgliedskonto abzielt, in dem Sie die Security Hub Hub-Integration deaktiviert haben, haben die von Ihrem Administrator ausgewählten Einstellungen Vorrang. OpsCentererstellt OpsItems weiterhin die Ergebnisse von Security Hub. Wenn Sie die Erstellung von Ergebnissen OpsItems für Security Hub in einem Mitgliedskonto beenden möchten, für das eine Ressourcendatensynchronisierung vorgesehen ist, wenden Sie sich an Ihren Administrator und bitten Sie ihn, Ihr Konto aus der Ressourcendatensynchronisierung zu entfernen oder die Security Hub Hub-Quelle in zu deaktivierenExplorer. Informationen zum Ändern von Einstellungen in Explorer finden Sie unterBearbeiten von Systems-Manager-Explorer-Datenquellen.

Incident Manager

Incident Manager, eine Funktion von AWS Systems Manager, bietet eine Incident-Management-Konsole, mit der Sie Vorfälle, die sich auf Ihre AWS gehosteten Anwendungen auswirken, abmildern und beheben können. Ein Vorfall ist jede Art von ungeplanter Unterbrechung oder Beeinträchtigung der Qualität von Services. Nachdem Sie Incident Manager eingerichtet und konfiguriert haben, erstellt das System automatisch OpsItems in OpsCenter.

Wenn das System einen Vorfall in Incident Manager erstellt, erstellt es auch ein OpsItem in OpsCenter und zeigt den Vorfall als zugehöriges Element an. Wenn das OpsItem bereits vorhanden ist, erstellt Incident Manager kein OpsItem. Das erste OpsItem ist als übergeordnetes OpsItem bekannt. Wenn ein Vorfall an Größe und Umfang zunimmt, können Sie Vorfälle zu einem vorhandenen OpsItem hinzufügen. Bei Bedarf können Sie manuell einen Vorfall für ein OpsItem erstellen. Nachdem ein Vorfall abgeschlossen ist, können Sie in Incident Manager eine Analyse erstellen, um den Behebungsprozess für ähnliche Probleme zu überprüfen und zu verbessern.

Standardmäßig ist OpsCenter mit Incident Manager integriert. Wenn Incident Manager nicht eingerichtet ist, wird auf der OpsCenter Seite eine Meldung zur Einrichtung von Incident Manager angezeigt. Wenn Incident Manager ein OpsItem erstellt, können Sie das OpsItem von OpsCenter verwalten und korrigieren. Anweisungen zum Erstellen eines Vorfalls für ein OpsItem finden Sie unter Erstellen eines Vorfalls für ein OpsItem.

Geben Sie einen Namen für den Benutzer ein und klicken Sie dann auf OpsItems

Nachdem Sie OpsCenter, eine Fähigkeit von AWS Systems Manager, eingerichtet und in Ihre AWS-Services integriert haben, erstellen Ihre AWS-Services automatisch OpsItems basierend auf Standardregeln, Ereignissen oder Warnungen.

Sie können die Status und Schweregrade der standardmäßigen Amazon EventBridge-Regeln anzeigen. Falls erforderlich, können Sie diese Regeln in Amazon EventBridge erstellen oder bearbeiten. Sie können auch Warnungen von Amazon CloudWatch anzeigen und Warnungen erstellen oder bearbeiten. Mit Regeln und Warnungen können Sie Ereignisse konfigurieren, für die Sie OpsItems automatisch generieren möchten.

Wenn das System ein OpsItem erstellt, befindet es sich im Status Offen. Sie können den Status zu In Bearbeitung ändern, wenn Sie mit der Untersuchung des OpsItem beginnen und in Gelöst, nachdem Sie das OpsItem behoben haben. Weitere Informationen zum Konfigurieren von Alarmen und Regeln in AWS-Services zum Erstellen von OpsItems und zum manuellen Erstellen von OpsItems finden Sie in den folgenden Themen.

Themen

- Konfigurieren von EventBridge-Regeln zum Erstellen von OpsItems
- Konfigurieren von CloudWatch-Alarmen zum Erstellen von OpsItems
- Manuelles Erstellen der OpsItems

Konfigurieren von EventBridge-Regeln zum Erstellen von OpsItems

Wenn Amazon EventBridge ein Ereignis empfängt, erstellt es basierend auf Standardregeln ein neues OpsItem. Sie können eine Regel erstellen oder eine vorhandene Regel bearbeiten, um OpsCenter als Ziel eines EventBridge-Ereignisses festzulegen. Weitere Informationen zum Erstellen einer neuen Ereignisregel finden Sie unter Erstellen einer Regel für ein AWS-Service im Amazon-EventBridge-Benutzerhandbuch.

So konfigurieren Sie EventBridge-Regeln zum Erstellen von OpsItems in OpsCenter

1. Öffnen Sie die Amazon EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.
3. Auf der Seite Regeln wählen Sie für Event Bus die Option Standard.
4. Wählen Sie für Regeln eine Regel aus, indem Sie das Kontrollkästchen neben dessen Namen aktivieren.
5. Wählen Sie den Namen der Regel aus, um die Detailseite zu öffnen. Stellen Sie im Abschnitt Regeldetails sicher, dass der Status auf Aktiviert festgelegt ist.

   > **Note**
   >
   > Falls erforderlich, können Sie den Status mit Bearbeiten in der oberen rechten Ecke der Seite aktualisieren.

6. Wählen Sie die Registerkarte Targets (Ziele).
7. Klicken Sie in der Registerkarte Targets (Ziele) auf Edit (Bearbeiten).
8. Wählen Sie für Zieltypen aus AWS-Service.
9. Für Select a target (Ziel auswählen), wählen Sie Systems Manager OpsItem.
10. Für viele Zieltypen benötigt EventBridge die Berechtigung, Ereignisse an das Ziel zu senden. In diesen Fällen kann EventBridge die AWS Identity and Access Management-(IAM)-Rolle erstellen, die zum Ausführen Ihrer Regel erforderlich ist:

    - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Create a new role for this specific resource (Eine neue Rolle für diese spezifische Ressource erstellen).
    - Um eine von Ihnen erstellte IAM-Rolle zu verwenden, um EventBridge die Berechtigung zum Erstellen von OpsItems in OpsCenter zu erteilen, wählen Sie Use existing role (Vorhandene Rolle verwenden) aus.

11. Wählen Sie unter Zusätzliche Einstellungen für Zieleingabe konfigurieren die Option Eingabe-Transformator aus.

    Sie können die Option Eingabe-Transformer verwenden, um eine Deduplizierungszeichenfolge und andere wichtige Informationen für OpsItems anzugeben, z. B. Titel und Schweregrad.

12. Wählen Sie Configure input transformer (Eingabetransformator konfigurieren).

13. Geben Sie unter Zieleingabe-Transformator für Eingabepfad die Werte an, die aus dem auslösenden Ereignis analysiert werden sollen. Um beispielsweise die Startzeit, die Endzeit und andere Details des Ereignisses zu analysieren, das die Regel auslöst, verwenden Sie den folgenden JSON.

```
{
    "end-time": "$.detail.EndTime",
    "failure-cause": "$.detail.cause",
    "resources": "$.resources",
    "source": "$.detail.source",
    "start-time": "$.detail.StartTime"
}
```

14. Geben Sie für Template (Vorlage) die Informationen an, die an das Ziel gesendet werden sollen. Verwenden Sie beispielsweise den folgenden JSON, um Informationen an OpsCenter zu übergeben. Die Informationen werden verwendet, um eine OpsItem zu erstellen.

> **Note**
>
> Wenn die Eingabevorlage im JSON-Format vorliegt, darf der Objektwert in der Vorlage keine Anführungszeichen enthalten. Beispielsweise dürfen die Werte für Ressourcen, Fehlerursache, Quelle, Startzeit und Endzeit nicht in Anführungszeichen stehen.

```
{
    "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",
    "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>
        }
    }
}
```

Weitere Informationen zu diesen Feldern finden Sie unter Transforming target input (Zielaufgabe transformieren) im Amazon EventBridge-Benutzerhandbuch.

15. Wählen Sie Bestätigen aus.
16. Wählen Sie Next (Weiter).
17. Wählen Sie Next (Weiter).
18. Wählen Sie Update rule (Regel aktualisieren) aus.

Nachdem ein OpsItem aus einem Ereignis erstellt wurde, können Sie die Ereignisdetails einsehen. Öffnen Sie hierzu das OpsItem und blättern Sie nach unten zum Abschnitt Private operational data (Private Betriebsdaten). Weitere Informationen zum Konfigurieren der Optionen in einem OpsItem finden Sie unter Verwalten von OpsItems.

Konfigurieren von CloudWatch-Alarmen zum Erstellen von OpsItems

Während der integrierten Einrichtung von OpsCenter, einer Funktion von AWS Systems Manager, aktivieren Sie Amazon CloudWatch, um automatisch OpsItems basierend auf allgemeinen

Warnungen zu erstellen. Sie können eine Warnungerstellen oder eine vorhandenen Warnung bearbeiten, um OpsItems in OpsCenter zu erstellen.

CloudWatch erstellt automatisch eine neue serviceverknüpfte Rolle in AWS Identity and Access Management (IAM), wenn Sie eine Warnung konfigurieren, um OpsItems zu erstellen. Der Name der neuen Rolle lautet `AWSServiceRoleForCloudWatchAlarms_ActionSSM`. Weitere Informationen zu serviceverknüpften CloudWatch-Rollen finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch im Amazon-CloudWatch-Benutzerhandbuch.

Wenn eine CloudWatch-Warnung ein OpsItem generiert, zeigt OpsItem die CloudWatch-Warnung an – '***alarm_name***' befindet sich im ALARM-Status.

Um Details zu einem bestimmten OpsItem anzuzeigen, wählen Sie das OpsItem und dann die Registerkarte Details zu zugehörigen Ressourcen aus. Sie können OpsItems manuell ändern, um Details wie den Schweregrad oder die Kategorie zu ändern. Wenn Sie jedoch den Schweregrad oder die Kategorie einer Warnung bearbeiten, kann Systems Manager den Schweregrad oder die Kategorie von OpsItems, die bereits aus der Warnung erstellt wurden, nicht aktualisieren. Wenn ein Alarm ein OpsItem erstellt hat und Sie eine Deduplizierungszeichenfolge angegeben haben, wird der Alarm keine weiteren OpsItems erstellen, selbst wenn Sie den Alarm in CloudWatch bearbeiten. Wenn das OpsItem in OpsCenter gelöst ist, erstellt CloudWatch ein neues OpsItem.

Weitere Informationen zur Konfiguration von CloudWatch-Warnungen finden Sie in den folgenden Themen.

Themen

- Konfiguration einer CloudWatch-Warnung zum Erstellen von OpsItems
- Konfiguration einer vorhandenen CloudWatch-Warnung zum Erstellen von OpsItems (programmgesteuert)

Konfiguration einer CloudWatch-Warnung zum Erstellen von OpsItems

Sie können manuell eine Warnung erstellen oder eine vorhandene Warnung aktualisieren, um OpsItems über Amazon CloudWatch zu erstellen.

Bearbeiten einer vorhandenen Warnung und Konfigurieren von Systems Managers als Ziel dieser Warnung

1. Führen Sie die Schritte 1–9 durch, wie unter Erstellen einer CloudWatch-Warnung basierend auf einem statischen Schwellenwert im Benutzerhandbuch für Amazon CloudWatch angegeben.

2. Wählen Sie im Abschnitt Systems-Manager-Aktion die Option Systems-Manager-OpsCenter-Aktion hinzufügen aus.
3. Wählen Sie OpsItems.
4. Wählen Sie für Schweregrad eine Zahl von 1 bis 4 aus.
5. (Optional) Wählen Sie für Kategorie eine Kategorie für das OpsItem aus.
6. Führen Sie die Schritte 11–13 aus, wie unter Erstellen einer CloudWatch-Warnung basierend auf einem statischen Schwellenwert im Amazon-CloudWatch-Benutzerhandbuch angegeben.
7. Klicken Sie auf Next (Weiter) und schließen Sie den Assistenten ab.

Bearbeiten eines vorhandenen Alarms und Konfigurieren des Systems Managers als Ziel dieses Alarms

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Klicken Sie im Navigationsbereich auf Alarme.
3. Wählen Sie den Alarm aus, wählen Sie dann Actions und anschließend Edit.
4. (Optional) Ändern Sie Einstellungen in den Bereichen Metrics (Metriken) und Conditions (Bedingungen) und wählen Sie dann Next (Weiter).
5. Wählen Sie im Bereich Systems Manager Add Systems Manager OpsCenter action.
6. Wählen Sie für Schweregrad eine Zahl aus.

> **Note**
>
> Der Schweregrad ist ein benutzerdefinierter Wert. Sie oder Ihre Organisation bestimmen, was jeder Schweregrad bedeutet und welche Service-Level-Vereinbarungen mit jedem Schweregrad verknüpft sind.

7. (Optional) Wählen Sie für Category eine Option aus.
8. Klicken Sie auf Next (Weiter) und schließen Sie den Assistenten ab.

Konfiguration einer vorhandenen CloudWatch-Warnung zum Erstellen von OpsItems (programmgesteuert)

Sie können Amazon CloudWatch-Warnungen programmgesteuert konfigurieren, um OpsItems mit AWS Command Line Interface (AWS CLI), AWS CloudFormation-Vorlagen oder Java-Code-Snippets zu verwenden.

Themen

- Bevor Sie beginnen
- Konfigurieren von CloudWatch-Warnungen zum Erstellen von OpsItems (AWS CLI)
- Konfiguration von CloudWatch-Warnungen zum Erstellen oder Aktualisieren von OpsItems (CloudFormation)
- Konfigurieren von CloudWatch-Warnungen zum Erstellen oder Aktualisieren von OpsItems (Java)

Bevor Sie beginnen

Wenn Sie eine vorhandenen Warnung programmgesteuert bearbeiten oder eine neuen Warnung erstellen, der OpsItems erstellt, müssen Sie einen Amazon-Ressourcennamen (ARN) angeben. Dieser ARN identifiziert Systems Manager OpsCenter als Ziel für OpsItems, die aus dem Alarm erstellt wurden. Sie können den ARN so anpassen, dass OpsItems, die aus dem Alarm erstellt wurden, bestimmte Informationen wie Schweregrad oder Kategorie enthalten. Jede ARN enthält die in der folgenden Tabelle beschriebenen Informationen.

Parameter	Details
`Region` (Erforderlich)	Die AWS-Region, in der sich der Alarm befindet. Beispiel: `us-west-2` . Weitere Informationen zu AWS-Regionen, in denen Sie OpsCenter verwenden können, finden Sie unter AWS Systems Manager-Endpunkte und Kontingente.
`account_ID` (Erforderlich)	Die gleiche AWS-Konto-ID, die zum Erstellen des Alarms verwendet wird. Beispiel: `123456789012` . Der Konto-ID muss ein Doppelpunkt (`:`) und der Parameter `opsitem` folgen, wie in den folgenden Beispielen gezeigt.
`severity` (Erforderlich)	Ein benutzerdefinierter Schweregrad für OpsItems wurde aus dem Alarm erstellt. Zulässige Werte: `1`, `2`, `3`, `4`
`Category` (Optional)	Eine Kategorie für OpsItems, die aus dem Alarm erstellt wurde. Gültige Werte:

Parameter	Details
	`Availability` , `Cost`, `Performance` , `Recovery` und `Security`.

Erstellen Sie den ARN mit der folgenden Syntax. Dieser ARN enthält nicht den optionalen `Category`-Parameter.

```
arn:aws:ssm:Region:account_ID:opsitem:severity
```

Im Folgenden sehen Sie ein Beispiel.

```
arn:aws:ssm:us-west-2:123456789012:opsitem:3
```

Verwenden Sie die folgende Syntax, um einen ARN zu erstellen, der den optionalen `Category`-Parameter verwendet.

```
arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name
```

Im Folgenden sehen Sie ein Beispiel.

```
arn:aws:ssm:us-west-2:123456789012:opsitem:3#CATEGORY=Security
```

Konfigurieren von CloudWatch-Warnungen zum Erstellen von OpsItems (AWS CLI)

Für diesen Befehl müssen Sie einen ARN für den `alarm-actions`-Parameter angeben. Informationen zum Erstellen des ARN finden Sie unter Bevor Sie beginnen.

Konfigurieren von CloudWatch-Warnungen zum Erstellen von OpsItems (AWS CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um Informationen über den Alarm zu sammeln, den Sie konfigurieren möchten.

```
aws cloudwatch describe-alarms --alarm-names "alarm name"
```

3. Führen Sie den folgenden Befehl aus, um einen Alarm zu aktualisieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
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
```

Ein Beispiel:

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
```

Konfiguration von CloudWatch-Warnungen zum Erstellen oder Aktualisieren von OpsItems (CloudFormation)

Dieser Abschnitt enthält AWS CloudFormation-Vorlagen, die Sie verwenden können, um CloudWatch-Warnungen so zu konfigurieren, dass sie automatisch OpsItems erstellen oder aktualisieren. Für jede Vorlage muss ein ARN für den `AlarmActions`-Parameter angegeben werden. Informationen zum Erstellen des ARN finden Sie unter Bevor Sie beginnen.

Metrik-Warnung – Verwenden Sie die folgenden CloudFormation-Vorlage, um eine CloudWatch-Metrik-Warnung zu erstellen oder zu aktualisieren. Der in dieser Vorlage angegebene Alarm überwacht die Statusprüfungen der Amazon Elastic Compute Cloud (Amazon EC2)-Instance. Wenn der Alarm in den `ALARM`-Zustand wecchselt, erstellt es ein OpsItem in 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"}}]
```

```
          }
        }
      }
    }
```

Zusammengesetzte Warnung – Verwenden Sie die folgende CloudFormation-Vorlage um zusammengesetzte Warnungen zu erstellen oder zu aktualisieren. Ein zusammengesetzter Alarm besteht aus mehreren Metrikalarmen. Wenn der Alarm in den ALARM-Zustand wecchselt, erstellt es ein OpsItem in 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"
         }
      }
   }
```

Konfigurieren von CloudWatch-Warnungen zum Erstellen oder Aktualisieren von OpsItems (Java)

Dieser Abschnitt enthält Java-Codeausschnitte, die Sie verwenden können, um CloudWatch-Warnungen so zu konfigurieren, dass sie automatisch OpsItems erstellen oder aktualisieren. Für jeden Ausschnitt müssen Sie einen ARN für den `validSsmActionStr`-Parameter angeben. Informationen zum Erstellen des ARN finden Sie unter Bevor Sie beginnen.

Eine spezifischer Warnung – Verwenden Sie den folgenden Java-Codeausschnitt, um eine CloudWatch-Warnung zu erstellen oder zu aktualisieren. Der in dieser Vorlage angegebene Alarm überwacht die Statusprüfungen der Amazon-EC2-Instance. Wenn der Alarm in den `ALARM`-Zustand wecchselt, erstellt es ein OpsItem in 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);
    }
```

Alle Warnungen aktualisieren – Verwenden Sie den folgenden Java-Codeausschnitt, um alle CloudWatch-Warnungen in Ihrem AWS-Konto so zu aktualisieren, dass OpsItems erstellt wird, wenn eine Warnung in den ALARM-Status wechselt.

```
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;
            }
        }
    }
```

Manuelles Erstellen der OpsItems

Wenn Sie ein Betriebsproblem feststellen, können Sie manuell ein OpsItem von OpsCenter, eine Funktion von AWS Systems Manager, erstellen, um das Problem zu verwalten und zu lösen.

Wenn Sie eine OpsCenter für eine kontoübergreifende Verwaltung einrichten, kann ein delegierter Administrator von Systems Manager oder einem AWS Organizations-Verwaltungskonto OpsItems für Mitgliederkonten erstellen. Weitere Informationen finden Sie unter (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

Sie können OpsItems mit der AWS Systems Manager-Konsole, der AWS Command Line Interface (AWS CLI), oder der AWS Tools for Windows PowerShell erstellen.

Themen

- Manuelles Erstellen von OpsItems (Konsole)
- Manuelles Erstellen von OpsItems (AWS CLI)
- Manuelles Erstellen von OpsItems (PowerShell)

Manuelles Erstellen von OpsItems (Konsole)

Sie können OpsItems manuell mit der AWS Systems Manager-Konsole erstellen. Wenn Sie ein OpsItem erstellen, wird es in Ihrem OpsCenter-Konto angezeigt. Wenn Sie OpsCenter für die kontoübergreifende Verwaltung einrichten, bietet OpsCenter dem delegierten Administrator oder dem Verwaltungskonto die Möglichkeit, OpsItems für ausgewählte Mitgliedskonten zu erstellen. Weitere Informationen finden Sie unter (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

So erstellen Sie ein OpsItem mit der AWS Systems Manager-Konsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Create (Erstellen)OpsItem aus. Wenn diese Schaltfläche nicht angezeigt wird, wählen Sie die Registerkarte OpsItems und dann Create OpsItem aus.
4. (Optional) Wählen Sie Anderes Konto und dann das Konto aus, für das Sie das OpsItem erstellen möchten.

   > **Note**
   >
   > Dieser Schritt ist erforderlich, wenn Sie OpsItems für ein Mitgliedskonto erstellen.

5. Geben Sie unter Title (Titel) einen aussagekräftigen Namen ein, um den Zweck des OpsItem zu verstehen.
6. Geben Sie unter Source (Quelle) den Typ der betroffenen AWS-Ressource oder andere Informationen der Quelle ein, damit Benutzer die Herkunft des OpsItem verstehen.

> **Note**
>
> Es ist nicht möglich, das Feld Source (Quelle) nach dem Erstellen des OpsItem zu bearbeiten.

7. (Optional) Wählen Sie unter Priority (Priorität) die Priorität aus.
8. (Optional) Wählen Sie für Severity (Schweregrad) den Schweregrad aus.
9. (Optional) Wählen Sie für Category (Kategorie) eine Kategorie aus.
10. Geben Sie unter Description (Beschreibung) Informationen zu diesem OpsItem ein, einschließlich der Schritte, um das Problem zu reproduzieren (falls zutreffend).

> **Note**
>
> Die Konsole unterstützt die meisten Markdown-Formate im OpsItem-Beschreibungsfeld. Weitere Informationen finden Sie unter Verwenden von Markdown in der Konsole im Einsteiger-Handbuch Erste Schritte mit der AWS Management Console.

11. Geben Sie für Deduplizierungszeichenfolge Wörter ein, die das System bei der Überprüfung auf OpsItems-Duplikate überprüfen soll. Weitere Informationen zu Deduplizierungszeichenfolgen finden Sie unter Verwalten von OpsItems-Duplikaten.
12. (Optional) Geben Sie unter Benachrichtigungen den Amazon-Ressourcenname (ARN) des Amazon-SNS-Themas an, an das Benachrichtigungen gesendet werden sollen, wenn dieses OpsItem aktualisiert wird. Sie müssen einen Amazon SNS-ARN angeben, der sich in derselben AWS-Region wie das OpsItem befindet.
13. (Optional) Wählen Sie unter Zugehörige Ressourcen die Option Hinzufügen zur Angabe der ID oder des ARN der betroffenen Ressource und aller zugehörigen Ressourcen.
14. Wählen Sie Create (Erstellen)OpsItem aus.

Bei Erfolg wird auf der Seite das OpsItem angezeigt. Wenn ein delegiertes Administrator- oder Verwaltungskonto ein OpsItem für ausgewählte Mitglieder erstellt, werden die neuen OpsItems im OpsCenter des Administrator- und Mitgliedskontos angezeigt. Weitere Informationen zum Konfigurieren der Optionen in einem OpsItem finden Sie unter Verwalten von OpsItems.

Manuelles Erstellen von OpsItems (AWS CLI)

Im folgenden Verfahren wird das Erstellen eines OpsItem über die AWS Command Line Interface (AWS CLI) beschrieben.

Erstellen eines OpsItem mithilfe der AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Öffnen Sie die AWS CLI und führen Sie den folgenden Befehl aus, um ein OpsItem zu erstellen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   ```
   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"
   ```

   Angabe von Betriebsdaten aus einer Datei

   Wenn Sie ein OpsItem erstellen, können Sie Betriebsdaten aus einer Datei angeben. Die Datei muss eine JSON-Datei sein, und der Inhalt der Datei muss das folgende Format aufweisen.

   ```
   {
     "key_name": {
       "Type": "SearchableString",
       "Value": "Up to 20 KB of data"
     }
   }
   ```

   Ein Beispiel.

   ```
   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**
>
> Informationen zur Eingabe von JSON-formatierten Parametern in der Befehlszeile verschiedener lokaler Betriebssysteme finden Sie unter Verwenden von Anführungszeichen mit Zeichenfolgen in der AWS CLI im AWS Command Line Interface-Benutzerhandbuch.

Das System gibt unter anderem folgende Informationen zurück

```
{
    "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

3. Führen Sie den folgenden Befehl aus, um Details zu dem von Ihnen erstellten OpsItem anzuzeigen.

```
aws ssm get-ops-item --ops-item-id ID
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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. Führen Sie den folgenden Befehl aus, um das OpsItem zu aktualisieren. Dieser Befehl ändert den Status von `Open` (Standardwert) zu `InProgress`.

```
aws ssm update-ops-item --ops-item-id ID --status InProgress
```

   Der Befehl hat keine Ausgabe.

5. Führen Sie den folgenden Befehl erneut aus, um zu überprüfen, ob der Status zu `InProgress` geändert wurde.

```
aws ssm get-ops-item --ops-item-id ID
```

Beispiele für die Erstellung eines OpsItem

Die folgenden Beispiele zeigen, wie Sie ein OpsItem mit dem Linux-Verwaltungsportal, macOS oder Windows erstellen und verwalten.

Linux-Verwaltungsportal oder macOS

Der folgende Befehl erstellt ein OpsItem, wenn eine Instance-Festplatte von Amazon Elastic Compute Cloud (Amazon EC2) voll ist.

```
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"
```

Der folgende Befehl verwendet den `/aws/resources`-Schlüssel in `OperationalData`, um ein OpsItem mit einer Amazon-DynamoDB-bezogenen Ressource zu erstellen.

```
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"
```

Der folgende Befehl verwendet den `/aws/automations`-Schlüssel in `OperationalData`, um ein OpsItem zu erstellen, das das `AWS-ASGEnterStandby`-Dokument als zugeordnetes Automation-Runbook angibt.

```
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

Der folgende Befehl erstellt ein OpsItem wenn eine Amazon Relational Database Service (Amazon RDS)-Instance nicht reagiert.

```
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"
```

Der folgende Befehl verwendet den `/aws/resources`-Schlüssel in `OperationalData`, um ein OpsItem mit einer Amazon-EC2-Instance-bezogenen Ressource zu erstellen.

```
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\"}}
```

Der folgende Befehl verwendet den `/aws/automations`-Schlüssel in `OperationalData`, um ein OpsItem zu erstellen, das das `AWS-RestartEC2Instance`-Runbook als zugeordnetes Automation-Runbook angibt.

```
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\"}}
```

Manuelles Erstellen von OpsItems (PowerShell)

Im folgenden Verfahren wird das Erstellen eines OpsItem über AWS Tools for Windows PowerShell beschrieben.

Erstellen eines OpsItem mithilfe von AWS Tools for Windows PowerShell

1. Öffnen Sie AWS Tools for Windows PowerShell und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen anzugeben.

   ```
   Set-AWSCredentials -AccessKey key-name -SecretKey key-name
   ```

2. Führen Sie den folgenden Befehl aus, um die AWS-Region für Ihre PowerShell-Sitzung festzulegen.

```
Set-DefaultAWSRegion -Region Region
```

3. Führen Sie den folgenden Befehl aus, um ein neues OpsItem zu erstellen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Dieser Befehl gibt ein Systems Manager Automation-Runbook zum Beheben dieses OpsItem an.

```
$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
```

   Im Erfolgsfall gibt der Befehl die ID des neuen OpsItem aus.

Das folgende Beispiel gibt den Amazon-Ressourcennamen (ARN) einer beeinträchtigten Amazon Elastic Compute Cloud (Amazon EC2)-Instance an.

```
$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
```

Verwalten von OpsItems

OpsCenter, eine Funktion von AWS Systems Manager, überwacht OpsItems von dessen Erstellung bis zur Auflösung. Wenn Sie eine kontoübergreifende Verwaltung einrichten für OpsCenter, kann ein delegierter Administrator oder ein Verwaltungskonto OpsItems von seinem Konto aus verwalten. Weitere Informationen finden Sie unter (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

Sie können OpsItems anzeigen und verwalten, indem Sie die folgenden Seiten in der Systems-Manager-Konsole nutzen:

- Zusammenfassung – Zeigt die Anzahl offener und in Bearbeitung befindlicher OpsItems, die Anzahl der OpsItems nach Quelle und Alter sowie betriebliche Einblicke an. Sie könnenOpsItems nach Quelle und OpsItems-Status filtern.

- OpsItems – Zeigt eine Liste von OpsItems mit mehreren Informationsfeldern an, z. B. Titel, ID, Priorität, Beschreibung, Quelle des OpsItem, sowie Datum und Uhrzeit der letzten Aktualisierung. Auf dieser Seite können Sie OpsItems manuell erstellen, Quellen konfigurieren, den Status eines OpsItem ändern und OpsItems nach neuen Vorfällen filtern. Sie können ein OpsItem auswählen, um dessen OpsItems-Detail-Seite anzuzeigen.

- OpsItem-Details – Bietet detaillierte Einblicke und Tools, mit denen Sie ein OpsItem verwalten können. Die OpsItems-Detailseite hat die folgenden Tabs:
  - Übersicht – Zeigt zugehörige Ressourcen, Runbooks, die in den letzten 30 Tagen ausgeführt wurden, und eine Liste der verfügbaren Runbooks an, die Sie ausführen können. Sie können auch ähnliche OpsItems anzeigen, Betriebsdaten hinzufügen und verbundene OpsItems hinzufügen.
  - Details der zugehörigen Ressource – Zeigt Informationen über die Ressource aus mehreren AWS-Services an. Erweitern Sie den Abschnitt Resource details (Ressourcen-Details), um Informationen über diese Ressource so anzuzeigen, wie sie von dem AWS-Service, auf dem sie gehostet wird, bereitgestellt werden. Sie können auch durch andere verwandte Ressourcen schalten, die mit diesem OpsItem verknüpft sind, indem Sie die Liste Related resources (verwandte Ressourcen) verwenden.

Weitere Informationen über die Verwaltung von OpsItems finden Sie in den folgenden Themen.

Themen

- Anzeigen von Details zu einem OpsItem
- Bearbeitung eines OpsItem
- Hinzufügen zugehöriger Ressourcen zu einem OpsItem
- Hinzufügen zugehöriger OpsItems zu einem OpsItem
- Hinzufügen von Betriebsdaten in ein OpsItem
- Erstellen eines Vorfalls für ein OpsItem
- Verwalten von OpsItems-Duplikaten
- Analyse betrieblicher Einblicke zur Reduzierung von OpsItems
- Anzeigen von OpsCenter-Protokollen und Berichten

Anzeigen von Details zu einem OpsItem

Um einen umfassenden Überblick über ein OpsItem zu erhalten, verwenden Sie die Seite OpsItem-Details in der OpsCenter-Konsole. Die Seite Übersicht zeigt die folgenden Informationen an:

- OpsItems-Details – Zeigt allgemeine Informationen für das ausgewählte OpsItem an.
- Zugehörige Ressourcen – Bei einer zugehörigen Ressource handelt es sich um die betroffene Ressource oder die Ressource, die das Ereignis ausgelöst hat, die das OpsItem erstellt hat.
- Automatisierungsausführungen in den letzten 30 Tagen – Eine Liste der Runbooks, die in den letzten 30 Tagen ausgeführt wurden.
- Runbooks – Sie können ein Runbook aus einer Liste verfügbarer Runbooks auswählen.
- Ähnliche OpsItems – Dies ist eine vom System generierte Liste von OpsItems, die für Sie relevant oder von Interesse sein könnten. Zum Generieren der Liste scannt das System die Titel und Beschreibungen aller OpsItems und gibt OpsItems, die ähnliche Wörter verwenden, zurück.
- Betriebsdaten – Bei Betriebsdaten handelt es sich um benutzerdefinierte Daten, die nützliche Details über das OpsItem bereitstellen. Sie können beispielsweise Protokolldateien, Fehlerzeichenfolgen, Lizenzschlüssel, Tipps zur Fehlerbehebung oder andere relevante Daten angeben.
- Zugehörige OpsItems – Sie können die IDs von OpsItemsangeben, die in irgendeiner Weise mit dem aktuellen OpsItem zusammenhängen.
- Details zu zugehörigen Ressourcen – Zeigt Datenanbieter, einschließlich Amazon-CloudWatch-Metriken und -Warnungen, AWS CloudTrail -Protokolle und Details aus AWS Config.

So zeigen Sie Details zu einer OpsItem an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie ein OpsItem aus, um dessen Details anzuzeigen.

Bearbeitung eines OpsItem

Der Abschnitt OpsItem-Details enthält Informationen über ein OpsItem, einschließlich der Beschreibung, des Titels, der Quelle, der OpsItem-ID und des Status.
Sie können OpsItem einzeln bearbeiten oder mehrere OpsItems auswählen und eines der folgenden Felder bearbeiten: Status, Priorität, Schweregrad, Kategorie.

Wenn Amazon eine EventBridge erstelltOpsItem, füllt es die Felder Titel, Quelle und Beschreibung aus. Sie können die Felder Titel und Beschreibung bearbeiten, jedoch nicht das Feld Quelle.

> **Note**
>
> Die Konsole unterstützt die meisten Markdown-Formatierungen im OpsItem Beschreibungsfeld. Weitere Informationen finden Sie unter Verwenden von Markdown in der Konsole im Handbuch Erste Schritte mit dem Handbuch AWS Management Console Erste Schritte.

Im Allgemeinen können Sie die folgenden konfigurierbaren Daten für ein OpsItem bearbeiten:

- Titel – Name des OpsItem. Die Quelle erstellt den Titel des OpsItem.
- Beschreibung – Informationen zu OpsItem, einschließlich (falls zutreffend) Schritte zur Reproduktion des Problems.
- Status – Der Status eines OpsItem kann Offen, In Bearbeitung oder Aufgelöst lauten.
- Priorität – Die Priorität eines OpsItem kann zwischen 1 und 5 liegen. Wir empfehlen, dass Ihre Organisation festlegt, was jede Prioritätsstufe bedeutet, und eine entsprechende Service-Level-Vereinbarung für jede Stufe erstellt.
- Schweregrad – Der Schweregrad eines OpsItems kann zwischen 1 und 4 liegen, wobei 1 für kritisch, 2 für hoch, 3 für mittel und 4 für niedrig steht.

- Kategorie – Die Kategorie eines OpsItems kann Verfügbarkeit, Kosten, Leistung, Wiederherstellung oder Sicherheit sein.
- Benachrichtigungen – Wenn Sie ein OpsItem bearbeiten, können Sie den Amazon-Ressourcennamen (ARN) eines Themas von Amazon Simple Notification Service im Feld Benachrichtigungen angeben. Indem Sie einen ARN angeben, stellen Sie sicher, dass alle Beteiligte eine Benachrichtigung erhalten, wenn das OpsItem bearbeitet wird, z. B. eine Statusänderung. Weitere Informationen finden Sie im Amazon Simple Notification Service-Entwicklerhandbuch.

> **Important**
>
> Das Amazon SNS SNS-Thema muss genauso existieren AWS-Region wie dasOpsItem. Wenn das Thema und das OpsItem sich in verschiedenen Regionen befinden, wird vom System ein Fehler zurückgegeben.

OpsCenterhat eine bidirektionale Integration mit. AWS Security Hub Wenn Sie den OpsItem-Status und den Schweregrad eines Sicherheitsbefunds aktualisieren, werden diese Änderungen automatisch an Security Hub gesendet, damit Sie immer die neuesten und richtigen Informationen sehen.

Wenn aus einem Security Hub-Befund ein erstellt OpsItem wird, werden Security Hub-Metadaten automatisch zum Betriebsdatenfeld von hinzugefügtOpsItem. Wenn diese Metadaten gelöscht werden, funktionieren die bidirektionalen Updates nicht mehr.

Bearbeiten von OpsItem-Details

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie eine OpsItem-ID, um die Detailseite zu öffnen oder mehrere OpsItems auszuwählen. Wenn Sie mehrere OpsItems wählen, können Sie nur den Status, die Priorität, den Schweregrad oder die Kategorie bearbeiten. Wenn Sie mehrere OpsItems bearbeiten, aktualisiert und speichert OpsCenter Ihre Änderungen, sobald Sie den neuen Status, die Priorität, den Schweregrad oder die Kategorie ausgewählt haben.
4. Wählen Sie im OpsItem Detailbereich die Option Bearbeiten aus.

5. Bearbeiten Sie die Details des OpsItem entsprechend den Vorschriften und Richtlinien Ihrer Organisation.
6. Wenn Sie fertig sind, wählen Sie Speichern.

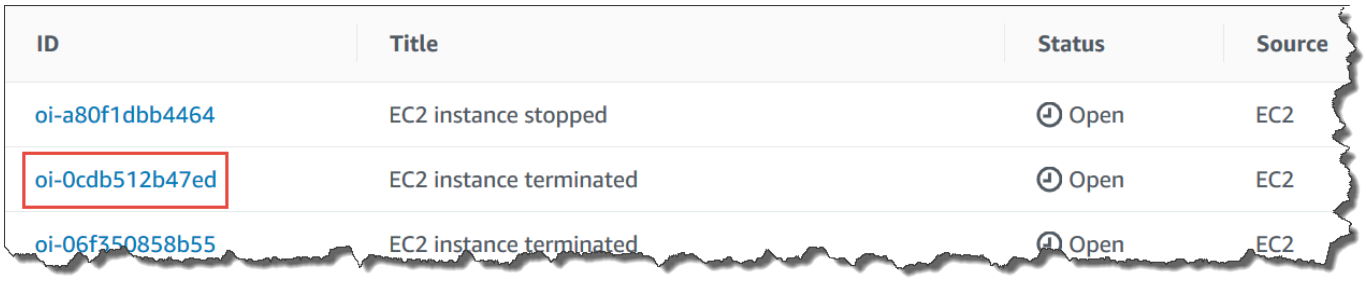
Hinzufügen zugehöriger Ressourcen zu einem OpsItem

Jedes OpsItem enthält einen Abschnitt Zugehörige Ressourcen, der den Amazon-Ressourcennamen (ARN) der zugehörigen Ressource auflistet. Eine zugehörige Ressource ist die betroffene AWS-Ressource, die untersucht werden muss.

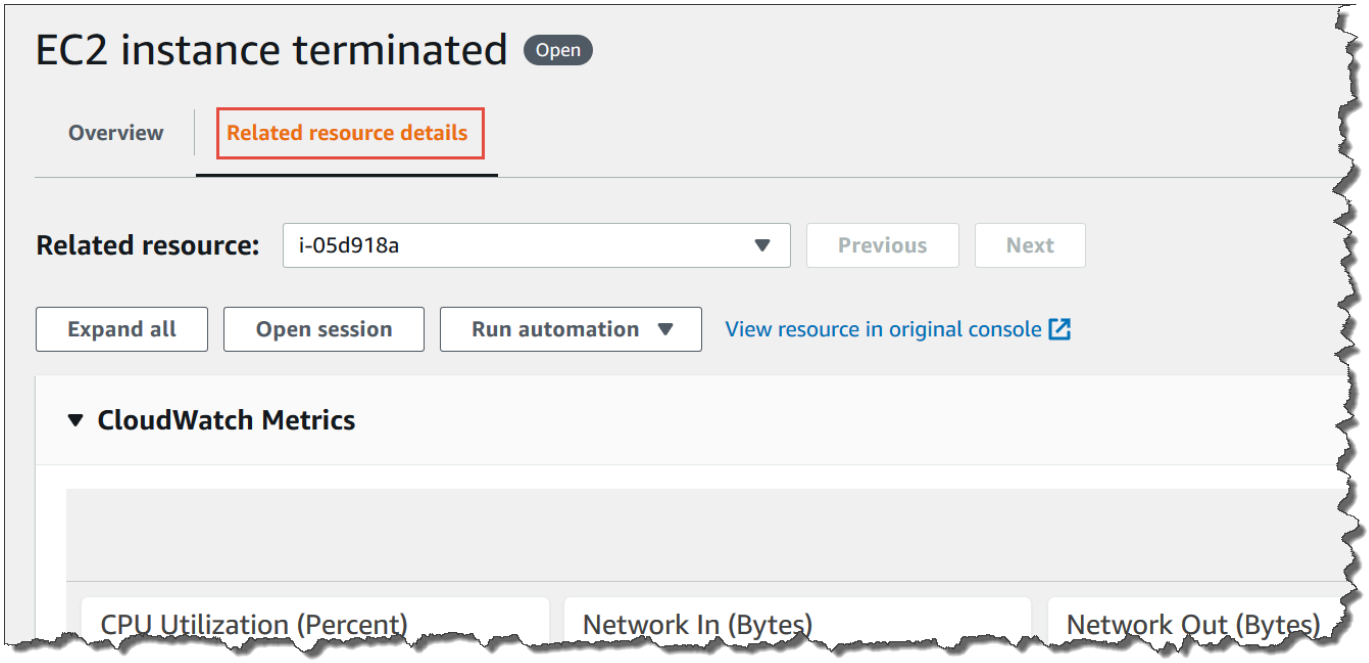
Wenn Amazon EventBridge das OpsItem erstellt, füllt das System automatisch das OpsItem mit dem ARN der Ressource. Sie können auch ARNs zugehöriger Ressourcen manuell festlegen. Bei einigen ARN-Typen erstellt OpsCenter automatisch einen Deep-Link, der Details zur Ressource direkt in der OpsCenter-Konsole anzeigt. Wenn Sie beispielsweise den ARN einer Instance der Amazon Elastic Compute Cloud (Amazon EC2) als zugehörige Ressource angeben, ruft OpsCenter Details zu dieser EC2-Instance ab. Auf diese Weise können Sie detaillierte Informationen zu Ihren betroffenen AWS-Ressourcen einsehen, ohne OpsCenter verlassen zu müssen.

So zeigen Sie zugehörige Ressourcen an und fügen sie einem OpsItem hinzu

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie die Registerkarte OpsItems aus.
4. Wählen Sie eine OpsItem-ID aus.



5. Zum Einsehen von Informationen über die betroffene Ressource wählen Sie die Registerkarte Related resources details (Details zugehörige Ressourcen) aus.



Diese Registerkarte zeigt Informationen über die Ressource aus mehreren AWS-Services an. Erweitern Sie den Abschnitt Resource details (Ressourcen-Details), um Informationen über diese Ressource so anzuzeigen, wie sie von dem AWS-Service, auf dem sie gehostet wird, bereitgestellt werden. Sie können auch durch andere verwandte Ressourcen schalten, die mit diesem OpsItem verknüpft sind, indem Sie die Liste Related resources (verwandte Ressourcen) verwenden.

6. Zum Hinzufügen zugehöriger Ressourcen wählen Sie die Registerkarte Overview (Übersicht) aus.
7. Wählen Sie im Abschnitt Related resources (Zugehörige Ressourcen die Option Hinzufügen aus.
8. Wählen Sie für Resource type (Ressourcentyp) eine Ressource aus der Liste aus.
9. Geben Sie für Resource ID entweder die ID oder den Amazon-Ressourcennamen (ARN) ein. Die Art der ausgewählten Informationen hängt von der Ressource ab, die Sie im vorherigen Schritt ausgewählt haben.

> **Note**
>
> Sie können die ARNs weiterer zugehöriger Ressourcen manuell hinzufügen. Jedes OpsItem kann maximal 100 zugehörige Ressourcen-ARNs auflisten.

In der folgenden Tabelle sind die Ressourcentypen aufgeführt, die automatisch Deep-Links zu verwandten Ressourcen erstellen.

Unterstützte Ressourcentypen

Ressourcenname	ARN-Format
AWS Certificate Manager-Zertifikat	`arn:aws:acm: region:account-id :certificate/ certificate-id`
Amazon EC2 Auto Scaling-Gruppe	`arn:aws:autoscaling: region:account-id :autoScalingGroup: groupid:autoScalingGroupName/ groupfriendlyname`
Amazon-CloudFront-Verteilung	`arn:aws:cloudfront:: account-id :*`
AWS CloudFormation-Stack	`arn:aws:cloudformation: region:account-id :stack/stackname /additionalidentifier`
Amazon CloudWatch-Alarm	`arn:aws:cloudwatch: region:account-id :alarm:alarm-name`
AWS CloudTrail-Trail	`arn:aws:cloudtrail: region:account-id :trail/trailname`
AWS CodeBuild-Projekt	`arn:aws:codebuild: region:account-id :resourcetype /resource`
AWS CodePipeline	`arn:aws:codepipeline: region:account-id :resource-specifier`

Ressourcenname	ARN-Format
Amazon DevOps Guru Insight	`arn:aws:devops-guru:region:account-id:insight/proactive or reactive/resource-id`
Amazon-DynamoDB-Tabelle.	`arn:aws:dynamodb:region:account-id:table/tablename`
Amazon Elastic Compute Cloud (Amazon EC2)-Kunden-Gateway	`arn:aws:ec2:region:account-id:customer-gateway/cgw-id`
Amazon EC2 elastic IP	`arn:aws:ec2:region:account-id:eip/eipalloc-id`
Amazon EC2 Dedicated Host	`arn:aws:ec2:region:account-id:dedicated-host/host-id`
Amazon EC2-Instance	`arn:aws:ec2:region:account-id:instance/instance-id`
Amazon EC2 Internet-Gateway	`arn:aws:ec2:region:account-id:internet-gateway/igw-id`
Amazon-EC2-Netzwerk-Zugriffssteuerungsliste (Netzwerk-ACL)	`arn:aws:ec2:region:account-id:network-acl/nacl-id`
Amazon EC2-Netzwerkschnittstelle	`arn:aws:ec2:region:account-id:network-interface/eni-id`
Amazon EC2 Routing-Tabelle	`arn:aws:ec2:region:account-id:route-table/route-table-id`

Ressourcenname	ARN-Format
Amazon EC2-Sicherheitsgruppe	`arn:aws:ec2: region:account-id :security-group/ security-group-id`
Amazon EC2 Subnetz	`arn:aws:ec2: region:account-id :subnet/subnet-id`
Amazon EC2-Volume	`arn:aws:ec2: region:account-id :volume/volume-id`
Amazon EC2 VPC	`arn:aws:ec2: region:account-id :vpc/vpc-id`
Amazon EC2 VPN-Verbindung	`arn:aws:ec2: region:account-id :vpn-connection/ vpn-id`
Amazon EC2 VPN-Gateway	`arn:aws:ec2: region:account-id :vpn-gateway/ vgw-id`
AWS Elastic Beanstalk-Anwendung	`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`

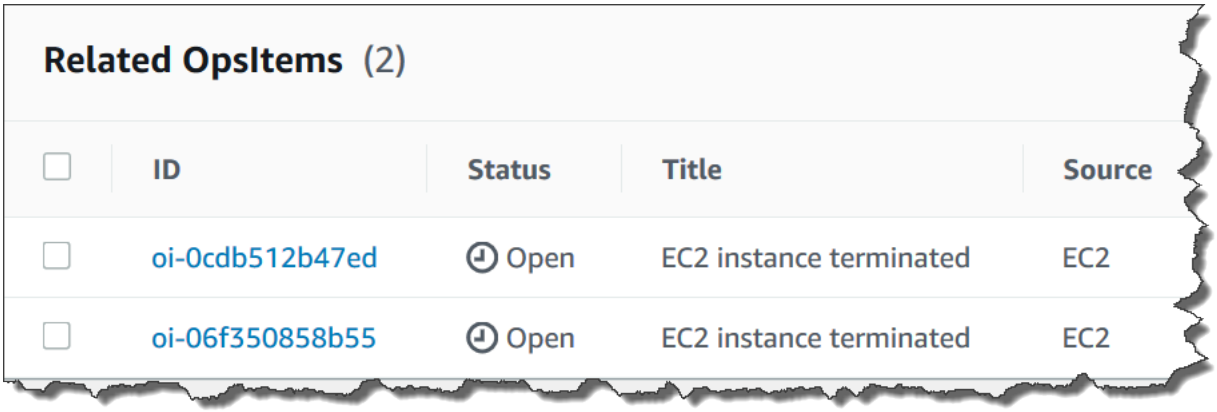
Ressourcenname	ARN-Format
Elastic Load Balancing (Network Load Balancer)	`arn:aws:elasticloadbalancing: region:account-id :loadbalancer/net/ load-balancer-name /load-balancer-id`
AWS Identity and Access Management (IAM)-Gruppe	`arn:aws:iam:: account-id :group/group-name`
IAM-Richtlinie	`arn:aws:iam:: account-id :policy/policy-name`
IAM role (IAM-Rolle)	`arn:aws:iam:: account-id :role/role-name`
IAM-Benutzer	`arn:aws:iam:: account-id :user/user-name`
AWS Lambda Funktion	`arn:aws:lambda: region:account-id :function: function-name`
Amazon Relational Database Service (Amazon RDS)-Cluster	`arn:aws:rds: region:account-id :cluster: db-cluster-name`
Amazon-RDS-Datenbank-Instance	`arn:aws:rds: region:account-id :db:db-instance-name`
Amazon RDS-Abonnement	`arn:aws:rds: region:account-id :es:subscription-name`
Amazon RDS-Sicherheitsgruppe	`arn:aws:rds: region:account-id :secgrp:security-group-name`

Ressourcenname	ARN-Format
Amazon RDS-Clusters Snapshot	`arn:aws:rds: region:account-id :cluster-snapshot: cluster-snapshot-name`
Amazon RDS-Subnetzgruppe	`arn:aws:rds: region:account-id :subgrp:subnet-group-name`
Amazon-Redshift-Cluster	`arn:aws:redshift: region:account-id :cluster: cluster-name`
Amazon-Redshift-Parametergruppe	`arn:aws:redshift: region:account-id :parametergroup: parameter-group-name`
Amazon Redshift-Sicherheitsgruppe	`arn:aws:redshift: region:account-id :securitygroup: security-group-name`
Amazon-Redshift-Clsuter-Snapshots	`arn:aws:redshift: region:account-id :snapshot: cluster-name /snapshot-name`
Amazon-Redshift-Subnetzgruppen	`arn:aws:redshift: region:account-id :subnetgroup: subnet-group-name`
Amazon Simple Storage Service (Amazon S3)-Bucket	`arn:aws:s3::: bucket_name`
AWS Config-Aufzeichnung des von AWS Systems Manager verwalteten Knoten-Bestands	`arn:aws:ssm: region:account-id :managed-instance-inventory / node_id`

Ressourcenname	ARN-Format
Systems Manager State Manager-Zuordnung.	`arn:aws:ssm: region:account-id :association/ association_ID`

Hinzufügen zugehöriger OpsItems zu einem OpsItem

Mithilfe von Zugehörigen OpsItems auf der Seite OpsItems Details können Sie betriebliche Probleme untersuchen und einen Kontext für ein Problem bereitstellen. OpsItems kann auf verschiedene Weise verwandt sein, einschließlich einer übergeordneten Beziehung zwischen OpsItems, einer Grundursache oder einem Duplikat. Sie können ein OpsItem einem anderen zuordnen, um es im Abschnitt Zugehörige OpsItem anzuzeigen. Sie können bis zu 10 IDs für andere OpsItems angeben, die sich auf das aktuelle OpsItem beziehen.

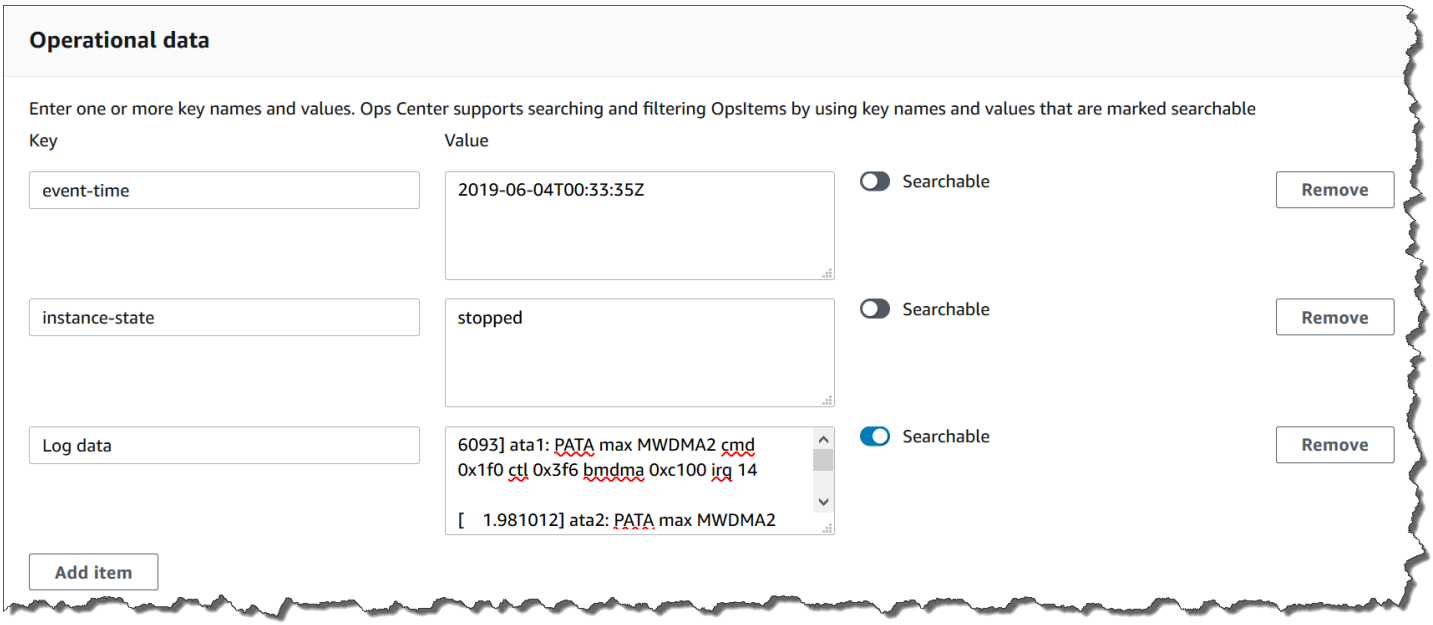


Hinzufügen von verwandten OpsItem

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie eine OpsItem-ID, um die Detailseite zu öffnen.
4. Wählen Sie im Abschnitt Related (Zugehörige) OpsItem die Option Add (Hinzufügen) aus.
5. Geben Sie unter OpsItem ID eine ID ein.
6. Wählen Sie Add (Hinzufügen) aus.

Hinzufügen von Betriebsdaten in ein OpsItem

Betriebsdaten sind benutzerdefinierte Daten, die nützliche Referenzdetails zu einem OpsItem bereitstellen. Sie können mehrere Schlüssel-Wert-Paare von Betriebsdaten eingeben. Sie können beispielsweise Protokolldateien, Fehlerzeichenfolgen, Lizenzschlüssel, Tipps zur Fehlerbehebung oder andere relevante Daten angeben. Die maximale Länge des Schlüssels kann 128 Zeichen und die maximale Größe des Werts 20 KB betragen.



Sie können die Daten für andere Benutzer im Konto durchsuchbar machen oder den Suchzugriff einschränken. Durchsuchbare Daten bedeuten, dass alle Benutzer mit Zugriff auf die OpsItem Übersichts-Seite (wie von der DescribeOpsItems-API-Operation bereitgestellt) die angegebenen Daten anzeigen und durchsuchen können. Operative Daten, die nicht durchsuchbar sind, sind nur für Benutzer sichtbar, die Zugriff auf das OpsItem(wie mit der API-Operation GetOpsItem bereitgestellt) haben.

Einfügen von Betriebsdaten in ein OpsItem

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie eine OpsItem-ID aus, um dessen Detailseite zu öffnen.
4. Erweitern Sie Betriebsdaten.
5. Wenn für das OpsItem keine Betriebsdaten vorhanden sind, wählen Sie Hinzufügen aus. Wenn für das OpsItem bereits Betriebsdaten vorhanden sind, wählen Sie Manage (Verwalten) aus.

Nachdem Sie die Betriebsdaten erstellt haben, können Sie den Schlüssel und den Wert bearbeiten oder zusätzliche Schlüssel-Wert-Paare hinzufügen, indem Sie Manage (Verwalten) auswählen.

6. Geben Sie unter Key (Schlüssel), ein oder mehrere Wörter an, damit Benutzer den Zweck der Daten verstehen.

> **Important**
>
> Betriebsdatenschlüssel können nicht auf diese Weise beginnen: `amazon`, `aws`, `amzn`, `ssm`, `/amazon`, `/aws`, `/amzn`, `/ssm`.

7. Geben Sie unter Value (Wert) die Daten an.
8. Wählen Sie Save (Speichern).

> **Note**
>
> Sie können OpsItems mithilfe des Operators Operational data (Operative Daten) auf der OpsItems-Seite filtern. Wählen Sie im Feld Suche die Option Betriebsdaten aus und geben Sie dann ein Schlüssel-Wert-Paar in JSON ein. Sie müssen das Schlüssel-Wert-Paar im folgenden Format eingeben: `{"key":"key_name","value":"a_value"}`

Erstellen eines Vorfalls für ein OpsItem

Verwenden Sie das folgende Verfahren, um einen Vorfall für ein OpsItem manuell zu erstellen, um es in AWS Systems Manager Incident Manager zu verfolgen und zu verwalten, was eine Funktion von AWS Systems Manager ist. Ein Vorfall ist jede Art von ungeplanter Unterbrechung oder Beeinträchtigung der Qualität von Services. Weitere Informationen zu Incident Manager finden Sie unter the section called “Integrieren von OpsCenter in anderen AWS-Services”.

Manuelles Erstellen eines Vorfalls für ein OpsItem

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.

3. Wenn Incident Manager ein OpsItem für Sie erstellt hat, wählen Sie es aus und fahren Sie mit Schritt 5 fort. Wenn nicht, wählen Sie Create OpsItem (Erstellen) und füllen Sie das Formular aus. Wenn diese Schaltfläche nicht angezeigt wird, wählen Sie die Registerkarte OpsItems und dann Create OpsItem aus.
4. Wenn Sie ein neues OpsItem erstellt haben, öffnen Sie es.
5. Wählen Sie Start Incident (Vorfall starten).
6. Wählen Für für Reaktionsplan den Incident-Manager-Reaktionsplan aus, den Sie diesem Vorfall zuweisen möchten.
7. (Optional) Geben Sie für Title (Titel) einen aussagekräftigen Namen ein, anhand dessen andere Teammitglieder die Art des Vorfalls verstehen können. Wenn Sie keinen neuen Titel eingeben erstellt OpsCenter das OpsItem und den entsprechenden Vorfall im Incident Manager unter Verwendung des Titels im Antwortplan.
8. (Optional) Wählen Sie für Incident impact eine Auswirkungsstufe für diesen Vorfall aus. Wenn Sie keine Auswirkungsstufe eingeben erstellt OpsCenter das OpsItem und den entsprechenden Vorfall im Incident Manager unter Verwendung der Auswirkungsstufe im Antwortplan.
9. Wählen Sie Starten.

Verwalten von OpsItems-Duplikaten

OpsCenter kann mehrere OpsItems-Duplikate für eine einzelne Quelle von mehreren AWS-Services erhalten. OpsCenter verwendet eine Kombination aus integrierter Logik und konfigurierbaren Deduplizierungszeichenfolgen, um die Erstellung von OpsItems-Duplikaten zu vermeiden. AWS Systems Manager wendet die integrierte Deduplizierungslogik an, wenn der API-Vorgang OpsItem erstellen aufgerufen wird.

AWS Systems Manager verwendet die folgende Deduplizierungslogik:

1. Beim Erstellen des OpsItem erstellt und speichert Systems Manager einen Hash-Wert basierend auf der Deduplizierungszeichenfolge und der Ressource, durch die das OpsItem ausgelöst wurde.
2. Wenn eine weitere Anfrage zur Erstellung eines OpsItem gestellt wird, prüft das System die Deduplizierungszeichenfolge der neuen Anfrage.
3. Wenn ein übereinstimmender Hash-Wert für diese Deduplizierungszeichenfolge vorhanden ist, überprüft Systems Manager den Status des vorhandenen OpsItem. Wenn der Status eines vorhandenen OpsItem offen oder in Bearbeitung ist, wird das OpsItem nicht erstellt. Wenn das vorhandene OpsItem gelöst wird, erstellt Systems Manager ein neues OpsItem.

Nachdem Sie ein OpsItem erstellt haben, können Sie die Deduplizierungszeichenfolgen in diesem OpsItem nicht ändern.

Zur Verwaltung von OpsItems-Duplikaten können Sie wie folgt vorgehen:

- Bearbeiten Sie die Deduplizierungszeichenfolge für eine Amazon-EventBridge-Regel, die auf OpsCenter ausgerichtet ist. Weitere Informationen finden Sie unter Bearbeiten einer Deduplizierungszeichenfolge in einer Standard-EventBridge-Regel.
- Geben Sie eine Deduplizierungszeichenfolge an, wenn Sie ein neues OpsItem manuell erstellen. Weitere Informationen finden Sie unter Angeben einer Deduplizierungszeichenfolge mit der AWS CLI.
- Überprüfen und beheben Sie OpsItems-Duplikate mit betrieblichen Einblicken. Sie können Runbooks verwenden, um OpsItems-Duplikate zu lösen.

  Um Sie bei der Auflösung von OpsItems-Duplikaten zu unterstützen und die Anzahl der von einer Quelle erstellten OpsItems zu reduzieren, stellt Systems-Manager-Automation-Runbooks bereit. Weitere Informationen finden Sie unter Lösen von OpsItems-Duplikaten basierend auf Erkenntnissen.

Bearbeiten einer Deduplizierungszeichenfolge in einer Standard-EventBridge-Regel

Führen Sie die folgenden Schritte aus, um eine Deduplizierungszeichenfolge für eine EventBridge-Regel mit Ziel OpsCenter zu erstellen.

So bearbeiten Sie eine Deduplizierungszeichenfolge für eine EventBridge-Regel

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.
3. Wählen Sie eine Regel und anschließend Edit (Bearbeiten) aus.
4. Rufen Sie die Seite Select target(s) (Ziel(e) auswählen) auf.
5. Wählen Sie im Bereich Additional settings (Zusätzliche Einstellungen) die Option Configure input transformer (Eingabetransformator konfigurieren).
6. Suchen Sie im Feld Template (Vorlage) den `"operationalData": { "/aws/dedup"` JSON-Eintrag und die Deduplizierungszeichenfolgen, die Sie bearbeiten möchten.

Der Eintrag der Deduplizierungszeichenfolge in EventBridge-Regeln verwendet das folgenden JSON-Format.

```
"operationalData": { "/aws/dedup": {"type": "SearchableString","value":
 "{\"dedupString\":\"Words the system should use to check for duplicate
 OpsItems\"}"}}
```

Ein Beispiel.

```
"operationalData": { "/aws/dedup": {"type": "SearchableString","value":
 "{\"dedupString\":\"SSMOpsCenter-EBS-volume-performance-issue\"}"}}
```

7. Bearbeiten Sie die Deduplizierungszeichenfolgen und wählen Sie dann Bestätigen aus.
8. Wählen Sie Next (Weiter).
9. Wählen Sie Next (Weiter).
10. Wählen Sie Update rule (Regel aktualisieren) aus.

Angeben einer Deduplizierungszeichenfolge mit der AWS CLI

Sie können eine Deduplizierungszeichenfolge angeben, wenn Sie ein neues OpsItem mit der AWS Systems Manager-Konsole oder der AWS CLI manuell erstellt haben. Weitere Informationen zum Eingeben von Deduplizierungszeichenfolgen, wenn Sie ein OpsItem manuell in der Konsole erstellen, finden Sie unter Manuelles Erstellen der OpsItems. Wenn Sie die AWS CLI verwenden, können Sie die Deduplizierungszeichenfolge für den `OperationalData`-Parameter eingeben. Die Parameter-Syntax verwendet JSON, wie im folgenden Beispiel gezeigt.

```
--operational-data '{"/aws/dedup":{"Value":"{\"dedupString\": \"Words the system should
 use to check for duplicate OpsItems\"}","Type":"SearchableString"}}'
```

Es folgt ein Beispiel für einen Befehl, mit dem die Deduplizierungszeichenfolge `disk full` angegeben wird.

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"
```

Analyse betrieblicher Einblicke zur Reduzierung von OpsItems

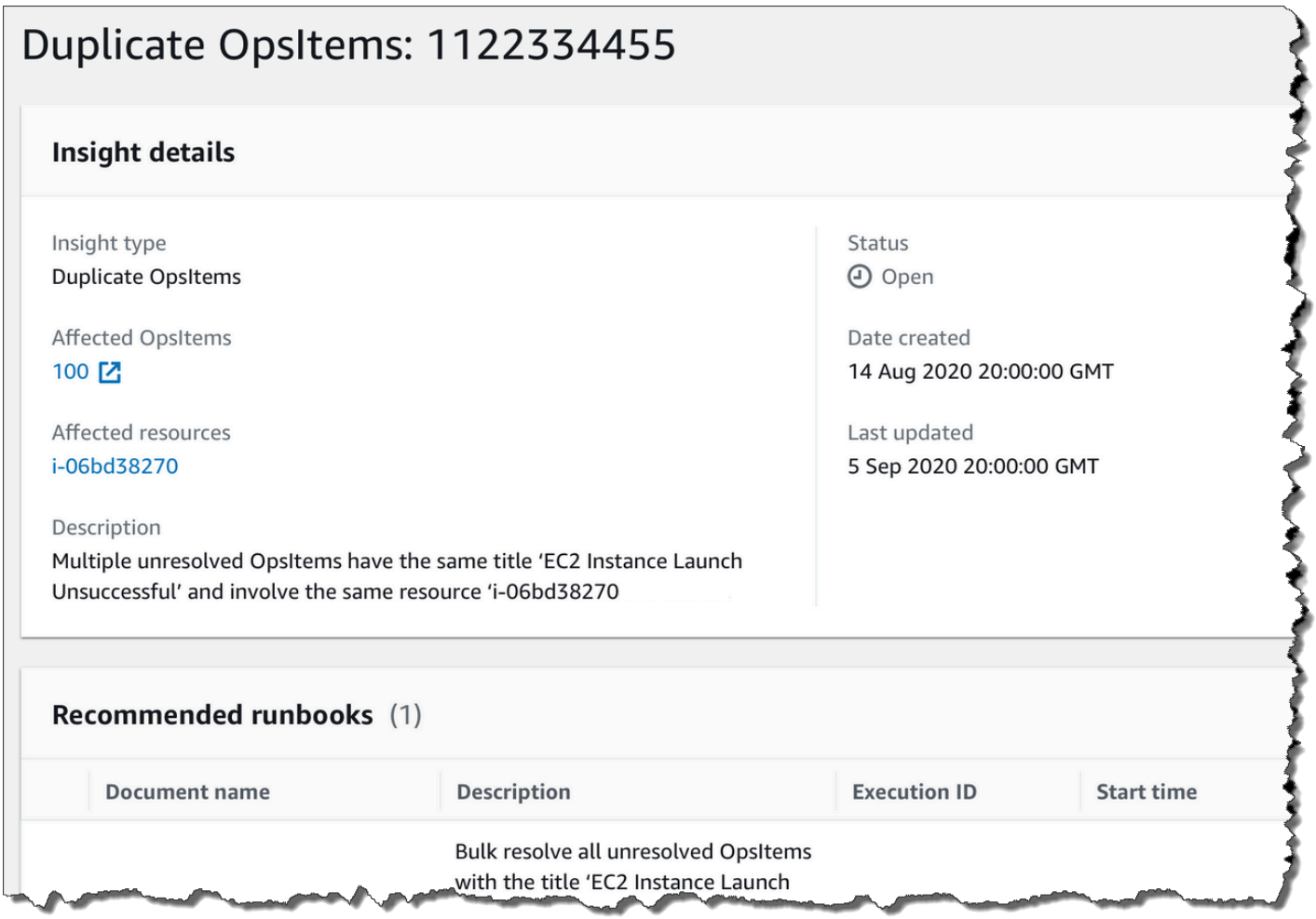
OpsCenter betriebliche Einblicke zeigt Informationen über doppelte OpsItems. OpsCenter analysiert automatisch OpsItems in Ihrem Konto und generiert drei Arten von Einblicken. Sie können diese Informationen im Abschnitt Betriebliche Einblicke auf der OpsCenter-Registerkarte Zusammenfassung einsehen.

- OpsItems-Duplikate – Ein Einblick wird generiert, wenn acht oder mehr OpsItems denselben Titel für dieselbe Ressource haben.
- Die häufigsten Titel – Ein Einblick wird generiert, wenn mehr als 50 OpsItems denselben Titel haben.
- Ressourcen mit den meisten OpsItems – Ein Einblick wird generiert, wenn eine AWS-Ressource mehr als 10 offene OpsItems hat. Diese Einblicke und die dazugehörigen Ressourcen werden in der Tabelle Ressourcen, die die meisten OpsItems erzeugen auf der OpsCenter-Registerkarte Zusammenfassung angezeigt. Die Ressourcen werden in absteigender Reihenfolge ihrer Anzahl an OpsItem aufgeführt.

> **Note**
>
> OpsCenter erstellt Einblicke für Ressourcen, die die meisten OpsItems erzeugen für die folgenden Ressourcentypen:
>
> - Instances von Amazon Elastic Compute Cloud (Amazon EC2)
> - Amazon EC2-Sicherheitsgruppen
> - Amazon EC2 Auto Scaling-Gruppe
> - Datenbank von Amazon Relational Database Service (Amazon RDS)
> - Amazon-RDS-Cluster
> - AWS Lambda Funktion
> - Amazon-DynamoDB-Tabelle.
> - Elastic Load Balancing-Load Balancer
> - Amazon-Redshift-Cluster
> - AWS Certificate Manager-Zertifikat
> - Amazon Elastic Block Store-Volume

OpsCenter erzwingt ein Limit von 15 Einblicken pro Typ. Wenn ein Typ dieses Limit erreicht, zeigt OpsCenter keine weiteren Einblicke für diesen Typ an. Um zusätzliche Einblicke anzuzeigen, müssen Sie alle OpsItems auflösen, die mit einem OpsInsight dieses Typs verknüpft sind. Wenn ein ausstehender Einblick aufgrund der Begrenzung auf 15 Einblicke nicht in der Konsole angezeigt werden kann, wird dieser Einblick sichtbar, nachdem ein anderer Einblick geschlossen wurde.

Wenn Sie eine Erkenntnis auswählen, zeigt OpsCenter Informationen über die betroffenen OpsItems und Ressourcen. Der folgende Screenshot zeigt ein Beispiel mit den Details einer Duplikats OpsItem-Erkenntniss.



Betriebliche Einblicke sind standardmäßig nicht aktiviert. Weitere Informationen zur Arbeit mit betrieblichen Einblicken finden Sie in den folgenden Themen.

Themen

- Aktivieren betrieblicher Einblicke
- Lösen von OpsItems-Duplikaten basierend auf Erkenntnissen
- Deaktivieren betrieblicher Erkenntnisse

Aktivieren betrieblicher Einblicke

Sie können betrieblicher Einblicke auf der Seite OpsCenter in der Systems-Manager-Konsole aktivieren. Wenn Sie betriebliche Einblicke aktivieren, erstellt Systems Manager eine neue serviceverknüpfte AWS Identity and Access Management (IAM)-Rolle mit dem Namen `AWSServiceRoleForAmazonSSM_OpsInsights`. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Systems Manager verknüpft ist. Serviceverknüpfte

Rollen sind vordefiniert und enthalten alle Berechtigungen, die der Service benötigt, um andere AWS-Services in Ihrem Namen aufzurufen. Weitere Informationen zur serviceverknüpften `AWSServiceRoleForAmazonSSM_OpsInsights`-Rolle finden Sie unter Verwenden von Rollen zur Erstellung von OpsItems für betriebliche Einblicke in Systems Manager OpsCenter.

> **Note**
>
> Beachten Sie die folgenden wichtigen Informationen:
>
> - Betrieblicher Einblicke werden Ihrem AWS-Konto in Rechnung gestellt. Weitere Informationen finden Sie unter AWS Systems Manager-Preisgestaltung.
> - OpsCenter aktualisiert regelmäßig Erkenntnisse mithilfe eines Batch-Prozesses. Dies bedeutet, dass die in OpsCenter angezeigte Liste der Einblicke möglicherweise nicht synchron ist.

Gehen Sie folgendermaßen vor, um betriebliche Einblicke in OpsCenter zu aktivieren und anzuzeigen.

So aktivieren Sie betriebliche Einblicke und zeigen sie an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie im Meldungsfeld Betrieblicher Einblick ist verfügbar die Option Aktivieren aus. Wenn Sie diese Meldung nicht sehen, scrollen Sie nach unten zum Abschnitt Betriebliche Einblicke und wählen Sie Aktivieren aus.
4. Nachdem Sie dieses Feature aktiviert haben, scrollen Sie auf der Registerkarte Zusammenfassung nach unten zum Abschnitt Betriebliche Einblicke.
5. Um eine gefilterte Liste von Einblicken anzuzeigen, wählen Sie den Link neben Duplizierten OpsItems, Häufigste Titel oder Ressourcen, die die meisten OpsItems erzeugen. Um alle Erkenntnisse anzuzeigen, wählen Sie View all operational insights (Alle betrieblichen Erkenntnisse anzeigen aus).
6. Wählen Sie eine Erkenntnis-ID, um weitere Informationen anzuzeigen.

Lösen von OpsItems-Duplikaten basierend auf Erkenntnissen

Um Erkenntnisse zu lösen, müssen Sie zuerst alle mit einem Erkenntnis verbunden OpsItems lösen. Sie können das `AWS-BulkResolveOpsItemsForInsight`-Runbook zum Lösen von OpsItems verwenden, die einem Erkenntnis zugeordnet sind.

Systems Manager bietet die folgenden Automation-Runbooks, die Ihnen dabei helfen, OpsItems-Duplikate aufzulösen und die Anzahl der von einer Quelle erstellten OpsItems zu reduzieren:

- Das `AWS-BulkResolveOpsItems`-Runbook löst OpsItems, die mit einem angegebenen Filter übereinstimmen.
- Das `AWS-AddOpsItemDedupStringToEventBridgeRule`-Runbook fügt eine Deduplizierungszeichenfolge für alle OpsItem-Ziele hinzu, die einer bestimmten Amazon EventBridge-Regel zugeordnet sind. Dieses Runbook fügt keine Deduplizierungszeichenfolge hinzu, wenn eine Regel bereits über eine verfügt.
- Die `AWS-DisableEventBridgeRule` deaktiviert eine Regel in EventBridge, wenn die Regel Dutzende oder Hunderte von OpsItems generiert.

Um einen operativen Einblick zu lösen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Scrollen Sie auf der Registerkarte Overview (Übersicht) bis zu Operational insights (Betriebliche Erkenntnisse) runter.
4. Wählen Sie Alle betrieblichen Einblicke anzeigen aus.
5. Wählen Sie eine Erkenntnis-ID, um weitere Informationen anzuzeigen.
6. Wählen Sie ein Runbook, und klicken Sie anschließend auf Ausführen.

Deaktivieren betrieblicher Erkenntnisse

Wenn Sie betriebliche Einblicke deaktivieren, stoppt das System die Erstellung neuer Einblicke und zeigt keine Einblicke in der Konsole an. Alle aktiven Einblicke verbleiben unverändert im System, obwohl sie in der Konsole nicht angezeigt werden. Wenn Sie dieses Feature erneut aktivieren, zeigt das System alle bisher nicht gelösten Einblicke an und beginnt mit der Erstellung neuer Einblicke. Verwenden Sie die folgende Vorgehensweise, um betriebliche Einblicke zu deaktivieren.

So deaktivieren Sie betriebliche Einblicke

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Settings (Einstellungen) aus.
4. Wählen Sie im Abschnitt Operational insights (Betriebliche Erkenntnisse) die Option Edit (Bearbeiten) und schalten Sie dann die Option Disable (deaktivieren) ein.
5. Wählen Sie Save (Speichern).

Anzeigen von OpsCenter-Protokollen und Berichten

AWS CloudTrail protokolliert AWS Systems Manager-OpsCenter-API-Aufrufe an die Konsole, die AWS Command Line Interface (AWS CLI) und das SDK. Sie können die Informationen in der CloudTrail-Konsole oder in einem Amazon Simple Storage Service (Amazon S3)-Bucket anzeigen. Amazon S3 verwendet einen Bucket, um alle CloudTrail-Protokolle für Ihr Konto zu speichern.

Protokolle von OpsCenter-Aktionen zeigen die OpsItem-Aktivitäten „erstellen“, „aktualisieren“, „abrufen“ und „beschreiben“ an. Weitere Informationen zum Anzeigen und Verwenden von CloudTrail-Protokollen von Systems Manager-Aktivitäten finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.

AWS Systems Manager OpsCenter liefert Ihnen die folgenden Informationen über OpsItems:

- OpsItem-Statuszusammenfassung – eine Zusammenfassung der OpsItems nach Status (Offen und in Bearbeitung, Offen oder In Bearbeitung).
- Quellen mit den meisten geöffneten OpsItems – Stellt eine Aufstellung der wichtigsten AWS-Services mit offenen OpsItems bereit.
- OpsItems nach Quelle und Alter – eine Anzahl von OpsItems, gruppiert nach Quelle und Tagen seit der Erstellung.

So zeigen Sie die Zusammenfassung des OpsCenter-Berichts an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.

3. Wählen Sie auf der Seite OpsItems-Übersicht die Option Zusammenfassung aus.
4. Wählen Sie unter OpsItems by source and age (OpsItems nach Quelle und Alter) die Suchleiste zum Filtern von OpsItems nach Source (Quelle). Filtern Sie die Liste nach Status.

Löschen Sie OpsItems

Sie können ein einzelnes OpsItem löschen, indem Sie die API-Operation OpsItem löschen mit dem AWS Command Line Interface- oder dem AWS-SDK aufrufen. Sie können ein OpsItem nicht in der AWS Management Console löschen. Zum Löschen eines OpsItem muss Ihr AWS Identity and Access Management (IAM)-Benutzer, Ihre Gruppe oder Ihre Rolle über eine Administratorberechtigung verfügen oder Ihnen muss die Berechtigung zum Aufrufen der API-Operation „`DeleteOpsItem`“ erteilt worden sein.

Important

Beachten Sie die folgenden wichtigen Informationen zu dieser Operation.

- Das Löschen eines OpsItem lässt sich nicht rückgängig machen. Sie können ein gelöschtes OpsItem nicht wiederherstellen.
- Bei dieser Operation wird ein Konsistenzmodell verwendet, was bedeutet, dass die Operation einige Minuten in Anspruch nehmen kann. Wenn Sie ein OpsItem löschen und sofort aufrufen, z. B. mit OpsItem abrufen, wird das gelöschte OpsItem möglicherweise immer noch in der Antwort angezeigt.
- Dieser Vorgang ist idempotent. Das System löst keine Ausnahme aus, wenn Sie diese Operation wiederholt für dasselbe OpsItem aufrufen. Wenn der erste Aufruf erfolgreich ist, geben alle weiteren Aufrufe die gleiche Antwort wie der erste Aufruf.
- Diese Operation unterstützt keine kontenübergreifenden Aufrufe. Ein delegierter Administrator oder ein Verwaltungskonto kann keine OpsItems in anderen Konten löschen, selbst wenn das OpsCenter für die kontenübergreifende Verwaltung eingerichtet wurde. Weitere Informationen zur kontoübergreifenden Verwaltung finden Sie unter (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.
- Wenn Sie die `OpsItemLimitExceededException` erhalten, können Sie eines oder mehrere OpsItems löschen, um die Gesamtzahl der OpsItems zu verringern und das Kontingent nicht zu überschreiten. Weitere Informationen zur dieser Ausnahme finden Sie unter Beheben von Problemen mit OpsCenter.

Löschen eines OpsItem

Gehen Sie wie folgt vor, um ein OpsItem zu löschen.

So löschen Sie einen OpsItem

1. Installieren und konfigurieren Sie AWS CLI, wenn noch nicht erfolgt. Weitere Informationen finden Sie unter Installieren oder Aktualisierung auf die neueste Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus. Ersetzen Sie *ID* durch die ID des zu löschenden OpsItem.

```
aws ssm delete-ops-item --OpsItemId ID
```

Wenn der Befehl erfolgreich ist, werden keine Daten zurückgegeben.

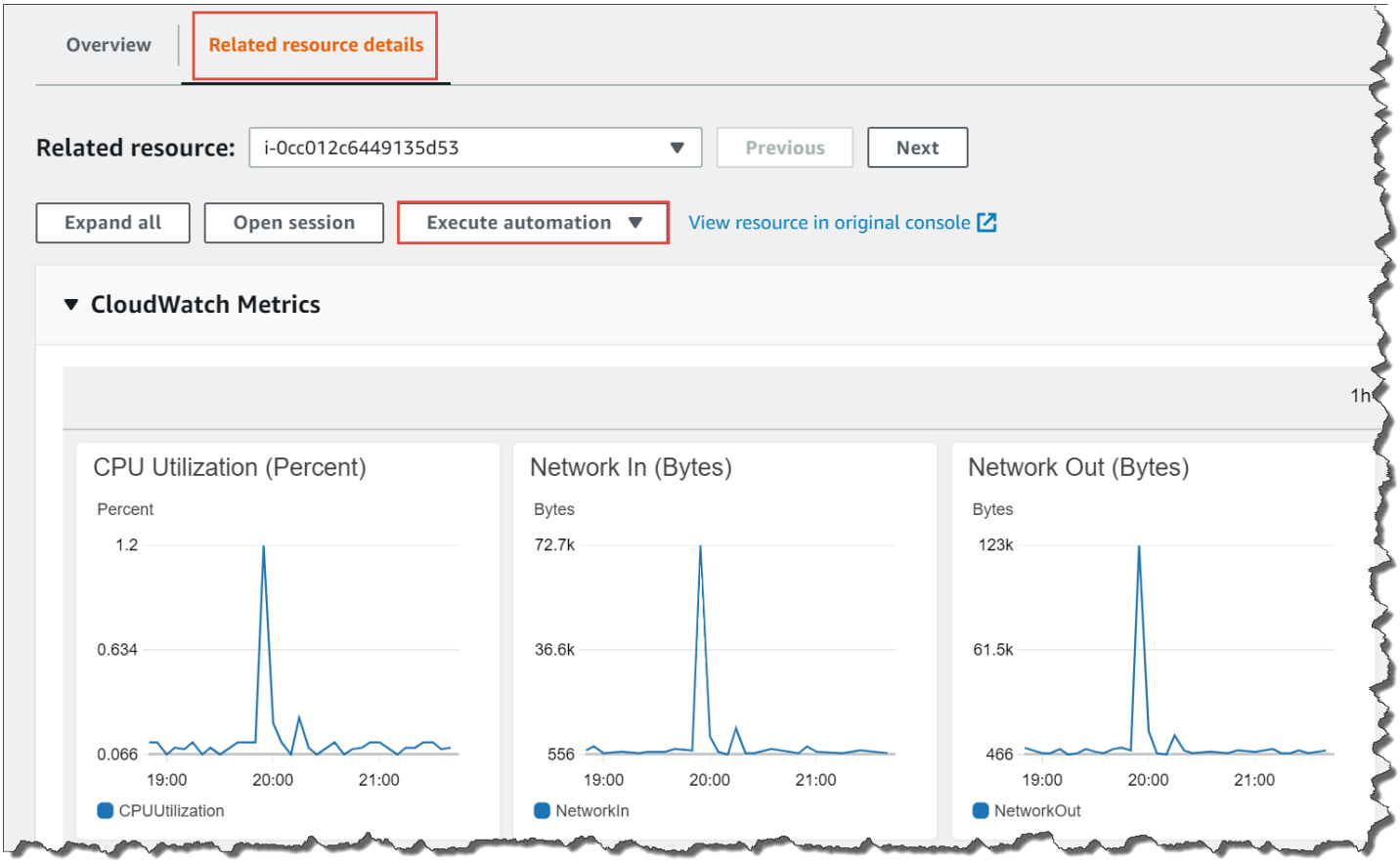
Beheben von OpsItem-Problemen

Mithilfe von AWS Systems Manager Automation-Runbooks können Sie Probleme mit AWS Ressourcen beheben, die in einem identifiziert wurden. OpsItem Die Automatisierung verwendet vordefinierte Runbooks, um häufig auftretende Probleme mit Ressourcen zu beheben. AWS

Jedes OpsItem enthält den Abschnitt Runbooks, der eine Liste von Runbooks enthält, die Sie zur Problembehebung verwenden können. Wenn Sie ein Automation-Runbook aus der Liste auswählen, zeigt OpsCenter automatisch einige der Felder an, die zum Ausführen des Dokuments erforderlich sind. Wenn Sie ein Automation-Runbook ausführen, verknüpft das System das Runbook mit der zugehörigen Ressource von OpsItem. Wenn Amazon ein EventBridge erstelltOpsItem, ordnet es dem ein Runbook zu. OpsItem OpsCenterführt eine 30-Tage-Aufzeichnung der Automatisierungs-Runbooks für einen. OpsItem

Sie können einen Status auswählen, um wichtige Details zum Runbook anzuzeigen, z. B. den Grund, warum eine Automatisierung fehlgeschlagen ist, und welcher Schritt des Automation-Runbooks ausgeführt wurde, als der Fehler auftrat, wie im folgenden Beispiel gezeigt.

Die Seite Related resource details (Details zu verwandten Ressourcen) für ein ausgewähltes OpsItemschließt die Liste Run automation mit ein. Sie können aktuelle oder ressourcenspezifische Automation-Runbooks auswählen und ausführen, um Probleme zu beheben. Diese Seite enthält auch Datenanbieter, darunter CloudWatch Amazon-Metriken und -Alarme, AWS CloudTrail Protokolle und Details von AWS Config.



Sie können Informationen zu einem Automation-Runbook einsehen, indem Sie entweder den Namen in der Konsole oder mithilfe von Referenz zu Systems Manager Automation auswählen.

Korrigieren eines OpsItem mit einem Runbook

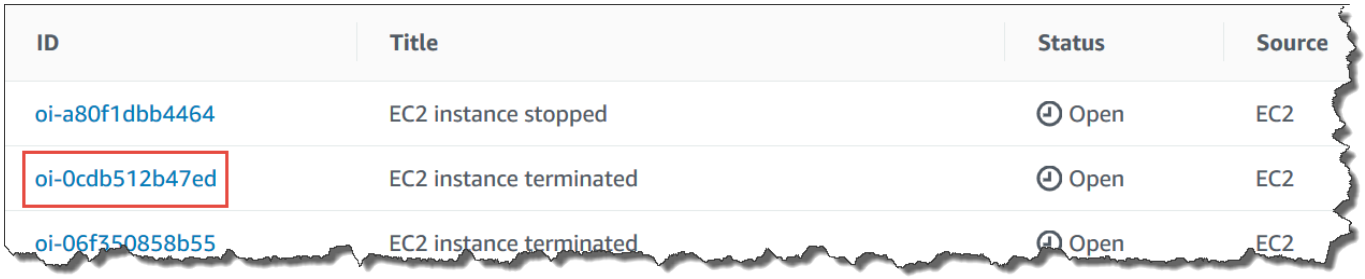
Bevor Sie ein Automation-Runbook verwenden, um ein OpsItem-Problem zu beheben, gehen Sie wie folgt vor:

- Überprüfen Sie, ob Sie über die Berechtigung zur Ausführung von Systems Manager Automation-Runbooks verfügen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
- Erfassen Sie Ressourcen-spezifische ID-Informationen für die Automatisierung, die Sie ausführen möchten. Beispiel: Wenn Sie eine Automatisierung ausführen möchten, die eine EC2-Instance erneut startet, müssen Sie die ID der neu zu startenden EC2-Instance angeben.

Ausführen eines Automation-Runbook, um ein OpsItem-Problem zu beheben

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie die OpsItem-ID, um die Detailseite zu öffnen.



4. Scrollen Sie zum Abschnitt Runbooks.
5. Verwenden Sie die Suchleiste oder die Zahlen oben rechts, um das Automation-Runbook zu finden, das Sie ausführen möchten.
6. Wählen Sie ein Runbook, und klicken Sie anschließend auf Execute (Ausführen).
7. Geben Sie die erforderlichen Informationen für das Runbook ein und klicken Sie anschließend auf Senden.

   Sobald Sie das Runbook gestartet haben, kehrt das System zum vorherigen Bildschirm zurück und zeigt den Status an.
8. Wählen Sie im Abschnitt Automatisierungsausführungen in den letzten 30 Tagen den Link Ausführungs-ID, um die einzelnen Schritte und den Status der Ausführung anzuzeigen.

Beheben eines OpsItem mithilfe eines zugeordneten Runbooks

Nachdem Sie ein Automation-Runbook von einem OpsItem ausgeführt haben, ordnet OpsCenter das Runbook dem OpsItem zu. Ein zugeordnetes Runbook wird in der Runbooks-Liste höher eingestuft als andere Runbooks.

Gehen Sie wie folgt vor, um ein Automation-Runbook auszuführen, das bereits mit einer zugehörigen Ressource in einem OpsItem verknüpft ist. Weitere Informationen zum Hinzufügen von zugehörigen Ressourcen finden Sie unter Verwalten von OpsItems.

Ausühren eines mit einer Ressource verknüpften Runbook zum Beheben eines OpsItem-Problems

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Öffnen Sie die OpsItem.

4. Wählen Sie im Abschnitt Related resources (Zugehörige Ressourcen) die Ressource aus, auf der Sie das Automaion-Runbook ausführen möchten.
5. Wählen Sie Run automation (Automatisierung ausführen) aus und anschließend das zugehörige Automation-Runbook, das Sie ausführen möchten.
6. Geben Sie die erforderlichen Informationen für das Runbook ein und klicken Sie anschließend auf Execute (Ausführen).

   Sobald Sie das Runbook gestartet haben, kehrt das System zum vorherigen Bildschirm zurück und zeigt den Status an.
7. Wählen Sie im Abschnitt Automatisierungsausführungen in den letzten 30 Tagen den Link Ausführungs-ID, um die einzelnen Schritte und den Status der Ausführung anzuzeigen.

Anzeigen von OpsCenter-Zusammenfassungsberichten

AWS Systems Manager OpsCenter enthält eine Zusammenfassungsseite, welche die folgenden Informationen automatisch anzeigt:

- OpsItem-Statusübersicht – Eine Zusammenfassung von OpsItems nach Status, z. B. `Open` und `In progress`.
- Quellen mit den am meisten geöffneten OpsItems – eine Aufstellung der wichtigsten AWS-Services mit offenen OpsItems.
- OpsItems nach Quelle und Alter – eine Anzahl von OpsItems, gruppiert nach Quelle und Tagen seit der Erstellung.

Um OpsCenter-Zusammenfassungsberichte anzuzeigen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter und dann die Registerkarte Zusammenfassung aus.
3. Gehen Sie im Abschnitt OpsItems nach Quelle und Alter wie folgt vor:

   1. (Optional) Wählen Sie im Filterfeld Quelle, wählen Sie `Equal`, `Begin With` oder `Not Equal` aus, und geben Sie dann einen Suchparameter ein.
   2. Wählen Sie in der nebenstehenden Liste einen der folgenden Statuswerte aus:

- `Open`
- `In progress`
- `Resolved`
- `Open and in progress`
- `All`

Beheben von Problemen mit OpsCenter

Hier finden Sie Informationen für die Behebung häufiger Fehler und Probleme mit OpsCenter.

Sie erhalten die Meldung „OpsItemLimitExceededException“.

Wenn Ihr AWS-Konto die maximal zulässige Anzahl von OpsItems erreicht hat, wenn Sie die API-Operation „CreateOpsItem“ aufrufen, erhalten Sie eine `OpsItemLimitExceededException`. OpsCenter gibt die Ausnahme zurück, wenn Ihr Aufruf die maximale Anzahl von OpsItems für eines der folgenden Kontingente überschreiten würde:

- Gesamtzahl der OpsItems pro AWS-Konto pro Region (einschließlich `Open` und `Resolved` OpsItems): 500.000
- Maximale Anzahl der OpsItems pro AWS-Konto pro Monat: 10.000

Diese Kontingente gelten für OpsItems, die aus einer beliebigen Quelle erstellt wurden, mit folgenden Ausnahmen:

- Durch AWS Security Hub-Erkenntnisse erstellte OpsItems
- OpsItems, die automatisch generiert werden, wenn ein Vorfall im Incident Manager geöffnet wird

OpsItems, die aus diesen Quellen erstellt werden, werden nicht auf Ihre OpsItem-Kontingente angerechnet, aber jedes OpsItem wird Ihnen in Rechnung gestellt.

Wenn Sie eine `OpsItemLimitExceededException` erhalten, können Sie OpsItems manuell löschen, bis Sie das Kontingent unterschritten haben. Ansonsten können Sie kein neues OpsItem erstellen. Auch hier gilt, dass das Löschen von OpsItems, die für Security-Hub-Erkenntnisse oder Incident-Manager-Vorfälle erstellt wurden, die Gesamtzahl der durch die Kontingente erzwungenen OpsItems nicht reduziert. Sie müssen OpsItems aus anderen Quellen löschen. Informationen zum Löschen eines OpsItem finden Sie unter Löschen Sie OpsItems.

Sie erhalten eine große Rechnung von AWS für eine große Anzahl von automatisch generierten OpsItems.

Wenn Sie die Integration mit AWS Security Hub konfiguriert haben, erstellt OpsCenter OpsItems für Security-Hub-Erkenntnisse. Abhängig von der Anzahl der Erkenntnisse, die Security Hub generiert, und dem Konto, bei dem Sie bei der Konfiguration der Integration angemeldet waren, kann OpsCenter eine große Anzahl von OpsItems generieren. Dafür fallen Kosten an. Im Folgenden finden Sie genauere Informationen zu den mit Security-Hub-Erkenntnissen generierten OpsItems:

- Wenn Sie bei der Konfiguration von OpsCenter und der Security-Hub-Integration im Security-Hub-Administratorkonto angemeldet sind, erstellt das System OpsItems für Erkenntnisse im Administrator- und allen Mitgliedskonten. Die OpsItems werden alle im Administratorkonto erstellt. Abhängig von einer Vielzahl von Faktoren kann dies zu einer unerwartet hohen Rechnung von AWS führen.

  Wenn Sie bei der Konfiguration der Integration in einem Mitgliedskonto angemeldet sind, erstellt das System nur OpsItems für Erkenntnisse in diesem individuellen Konto. Weitere Informationen über das Security-Hub-Administratorkonto, Mitgliedskonten und ihre Beziehung zum EventBridge-Ereignis-Feed für Erkenntnisse finden Sie unter Arten der Security Hub-Integration mit EventBridge im AWS Security Hub-Benutzerhandbuch.
- Für jede Erkenntnis, das ein OpsItem erstellt, wird Ihnen der reguläre Preis für die Erstellung des OpsItem berechnet. Ihnen wird auch berechnet, wenn Sie das OpsItem bearbeiten oder die entsprechende Erkenntnis im Security Hub aktualisiert wird (was ein OpsItem-Update auslöst).

> **Important**
>
> Wenn Sie der Meinung sind, dass eine große Anzahl von OpsItems irrtümlich erstellt wurde und Ihre AWS-Rechnung ungerechtfertigt ist, wenden Sie sich an AWS Support.

Gehen Sie wie folgt vor, wenn Sie nicht mehr möchten, dass das System OpsItems für Security-Hub-Erkenntnisse erstellt.

Erhalt von OpsItems für Security-Hub-Erkenntnisse stoppen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Settings (Einstellungen) aus.
4. Wählen Sie im Abschnitt Security Hub Erkenntnisse Bearbeiten.
5. Wählen Sie den Schieberegler, um Aktiviert zu Deaktiviert zu ändern. Wenn Sie den Schieberegler nicht umschalten können, wurde Security Hub nicht für Ihr AWS-Konto aktiviert.
6. Wählen Sie Speichern, um Ihre Konfiguration zu speichern. OpsCenter erstellt nicht mehr OpsItems auf der Grundlage von Security-Hub-Erkenntnissen.

> **Important**
>
> Wenn OpsCenter die Einstellung wieder auf Aktiviert setzt und weiterhin OpsItems für Erkenntnisse erstellt, melden Sie sich beim delegierten Administratorkonto von Systems Manager oder dem AWS Organizations-Verwaltungskonto an und wiederholen Sie diesen Vorgang. Sollten Sie keine Berechtigung haben, sich bei einem dieser Konten anzumelden, wenden Sie sich an Ihren Administrator und bitten ihn, diesen Vorgang zu wiederholen, um die Integration für Ihr Konto zu deaktivieren.

Von Systems Manager gehostete CloudWatch Amazon-Dashboards

CloudWatch Amazon-Dashboards sind anpassbare Homepages in der CloudWatch Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können, auch die Ressourcen, die auf verschiedene AWS-Regionen verteilt sind. Sie können CloudWatch Dashboards verwenden, um benutzerdefinierte Ansichten der Kennzahlen und Alarme für Ihre AWS Ressourcen zu erstellen. Mit Dashboards können Sie Folgendes erstellen:

- Eine einzige Ansicht für ausgewählte Metriken und Alarme, um Ihnen die Bewertung des Zustands Ihrer Ressourcen und Anwendungen in einer oder mehreren AWS-Regionen zu erleichtern. Sie können die für jede Metrik in jedem Diagramm verwendete Farbe auswählen, sodass Sie dieselbe Metrik über mehrere Diagramme hinweg verfolgen können.
- Ein operatives Playbook, das Teammitgliedern bei operativen Ereignissen Orientierungshilfen dazu bietet, wie auf bestimmte Vorfälle zu reagieren ist.

- Eine gemeinsame Ansicht wichtiger Maßnahmen für Ressourcen und Anwendungen, die von Teammitgliedern für einen schnelleren Kommunikationsfluss bei operativen Ereignissen gemeinsam genutzt wird.

Sie können Dashboards mithilfe der Konsole, der AWS Command Line Interface (AWS CLI) oder mithilfe der CloudWatch `PutDashboard` API erstellen. Weitere Informationen finden Sie unter Verwenden von CloudWatch Amazon-Dashboards im CloudWatch Amazon-Benutzerhandbuch.

AWS Systems Manager Verwaltung von Anwendungen

Application Management ist eine Suite von Funktionen, mit denen Sie Ihre in AWS ausgeführten Anwendungen verwalten können.

Themen

- AWS Systems Manager Application Manager
- AWS AppConfig
- AWS Systems Manager Parameter Store

AWS Systems Manager Application Manager

Application Manager, eine Funktion von AWS Systems Manager, unterstützt DevOps-Ingenieure bei der Untersuchung und Behebung von Problemen mit ihren AWS Ressourcen im Kontext ihrer Anwendungen und Cluster. Application Manager aggregiert Betriebsinformationen aus mehreren AWS-Services und Systems-Manager-Funktionen auf ein einzelnes AWS Management Console.

In Application Manager ist ein Anwendung eine logische Gruppierung von AWS Ressourcen, die Sie als Einheit betreiben möchten. Diese logische Gruppe kann verschiedene Versionen einer Anwendung, Besitzgrenzen für Operatoren oder Entwicklerumgebungen darstellen, um nur einige zu nennen. Application Manager Unterstützung für Container-Cluster umfasst sowohl Amazon Elastic Kubernetes Service (Amazon EKS) als auch Amazon Elastic Container Service (Amazon ECS) Cluster.

Wenn Sie Get started (Erste Schritte) auf der Startseite von Application Manager auswählen, importiert Application Manager automatisch Metadaten zu Ihren Ressourcen, die in anderen AWS-Services- oder Systems-Manager-Funktionen erstellt wurden. Für Anwendungen importiert Application Manager Metadaten über alle AWS Ressourcen, die in Ressourcengruppen organisiert sind. Jede Ressourcengruppe wird in der Kategorie Benutzerdefinierte Anwendungen als einzigartige Anwendung gelistet. Application Manager importiert automatisch Metadaten zu Ressourcen, die von AWS CloudFormation, AWS Launch Wizard, Amazon ECS und Amazon EKS erstellt wurden. Application Managerzeigt diese Ressourcen dann in vordefinierten Kategorien an.

Für Anwendungen umfasst die Liste Folgendes:

- Benutzerdefinierte Anwendungen

- Launch Wizard
- CloudFormation-Stacks
- AppRegisty-Anwendungen

Für Container-Cluster umfasst die Liste Folgendes:

- Amazon ECS-Cluster
- Amazon EKS-Cluster

Nach Abschluss des Imports können Sie Arbeitsvorgangsinformationen zu Ihren Ressourcen in diesen vordefinierten Kategorien anzeigen. Wenn Sie mehr Kontext zu einer Ressourcensammlung bereitstellen möchten, können Sie eine Anwendung manuell in Application Manager erstellen und Ressourcen oder Ressourcengruppen in diese Anwendung verschieben. So können Sie Betriebsinformationen im Kontext einer Anwendung anzeigen.

Nachdem Sie der Einrichtung und Konfiguration von AWS-Services und Systems-Manager-Funktionen, zeigt Application Manager die folgenden Arten von Informationen zu Ihren Ressourcen an:

- Informationen zum aktuellen Zustand, Status und Zustand von Amazon EC2 Auto Scaling der Amazon Elastic Compute Cloud (Amazon EC2)-Instances in Ihrer Anwendung
- Von Amazon CloudWatch bereitgestellte Alarme
- Compliance-Informationen, die von AWS Config und State Manager (eine Komponente von Systems Manager) bereitgestellt werden
- Kubernetes-Clusterinformationen, die von Amazon EKS bereitgestellt werden
- Protokolldaten, die von AWS CloudTrail und Amazon CloudWatch Logs bereitgestellt werden
- OpsItemsvon Systems Manager bereitgestellt OpsCenter
- Ressourcendetails, die von AWS-Services bereitgestellt werden, die sie hosten.
- Container-Cluster-Informationen, die von Amazon ECS bereitgestellt werden.

Um Probleme mit Komponenten oder Ressourcen zu beheben, stellt Application Manager Ihnen auch Runbooks bereit, die Sie Ihren Anwendungen zuordnen können. Um mit Application Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Application Manager aus.

Was sind die Vorteile der Nutzung von Application Manager?

Application Manager reduziert die Zeit, die DevOps Ingenieure benötigt, um Probleme mit AWSRessourcen zu erkennen und zu untersuchen. Um dies zu tun, zeigt Application Manager viele Arten von Betriebsinformationen im Kontext einer Anwendung in einer Konsole an . Application Manager reduziert auch die Zeit, die für die Behebung von Problemen benötigt wird, indem Runbooks bereitgestellt werden, die allgemeine Fehlerbehebungsaufgaben für AWS Ressourcen ausführen.

Über welche Features verfügt Application Manager?

Application Manager umfasst die folgenden Features:

- automatisches Importieren Ihrer AWSRessourcen

  Während der Ersteinrichtung können Sie wählen, ob Application Manager automatisch importiert und Ressourcen in Ihrem AWS-Konto anzeigt, die auf CloudFormation-Stacks basieren, AWS Resource Groups, Launch Wizard Bereitstellungen, AppRegistry-Anwendungen sowie Amazon ECS- und Amazon EKS-Cluster. Das System zeigt diese Ressourcen in vordefinierten Anwendungs- oder Clusterkategorien an. Danach, immer, wenn neue Ressourcen dieser Art zu Ihrem AWS-Konto hinzugefügt werden, zeigt Application Manager automatisch die neuen Ressourcen in den vordefinierten Anwendungs- und Clusterkategorien an.
- Erstellen oder Bearbeiten von CloudFormation-Stacks und -Vorlagen

  Application Manager unterstützt Sie bei der Bereitstellung und Verwaltung von Ressourcen für Ihre Anwendungen durch die Integration in CloudFormation. Sie können AWS CloudFormation Vorlagen und Stacks in Application Manager erstellen, bearbeiten und löschen. Application Managerenthält auch eine Vorlagenbibliothek, in der Sie Vorlagen klonen, erstellen und speichern können. Application Manager und CloudFormation zeigen dieselben Informationen über den aktuellen Status eines Stacks an. Vorlagen und Vorlagenaktualisierungen werden in Systems Manager gespeichert, bis Sie den Stack bereitstellen. Zu diesem Zeitpunkt werden die Änderungen auch in CloudFormation angezeigt.
- Informationen zu Ihren Instances im Kontext einer Anwendung anzeigen

  Application Manager lässt sich in Amazon Elastic Compute Cloud (Amazon EC2) integrieren, um Informationen zu Ihren Instances im Kontext einer Anwendung anzuzeigen. Application Manager zeigt Instance-Status, Status und den Zustand von Amazon EC2 Auto Scaling für eine ausgewählte Anwendung in einem grafischen Format an. Die Registerkarte Instances enthält auch eine Tabelle mit den folgenden Informationen für jede Instance in Ihrer Anwendung.

  - Instance-Status (Ausstehend, Angehalten, Wird ausgeführt, Beendet)
  - Ping-Status für SSM Agent
  - Status und Name des letzten Systems-Manager-Automation-Runbooks, das auf der Instance verarbeitet wurde
  - Eine Anzahl von Amazon-CloudWatch-Logs-Alarmen pro Status.
    - `ALARM` – Die Metrik oder der Ausdruck liegt außerhalb des festgelegten Schwellenwerts.
    - `OK` – Die Metrik oder der Ausdruck liegt innerhalb des festgelegten Schwellenwerts.
    - `INSUFFICIENT_DATA` – Der Alarm wurde soeben gestartet; die Metrik ist nicht verfügbar oder es sind nicht genügend Daten verfügbar, damit die Metrik den Alarmstatus bestimmen kann.
  - Zustand der Auto-Scaling-Gruppe für die übergeordneten und einzelnen Auto-Scaling-Gruppen
- Anzeigen von Betriebsmetriken und Alarmen für eine Anwendung oder ein Cluster

  Application Manager integriert Amazon CloudWatch, um Betriebsmetriken und Alarme in Echtzeit für eine Anwendung oder ein Cluster bereitzustellen. Sie können einen Drilldown in Ihre Anwendungsstruktur durchführen, um Alarme auf jeder Komponentenebene anzuzeigen oder Alarme für einen einzelnen Cluster anzuzeigen.
- Anzeigen von Protokolldaten für eine Anwendung

  Application Manager integriert Amazon CloudWatch Logs, um Protokolldaten im Kontext Ihrer Anwendung bereitzustellen, ohne Systems Manager verlassen zu müssen.
- Anzeigen und Verwalten von OpsItems für eine Anwendung oder ein Cluster

  Application Managerintegriert AWS Systems Manager OpsCenter, um eine Liste der operativen Arbeitselemente (OpsItems) für Ihre Anwendungen und Cluster bereitzustellen. Die Liste spiegelt automatisch generierte und manuell erstellte OpsItems. Sie können Details über die Ressource anzeigen, die eine OpsItem und den OpsItem Status, Quelle und Schweregrad erstellen.
- Anzeigen von Ressourcen-Compliance-Daten für eine Anwendung oder Cluster

  Application Manager integriert AWS Config, um Compliance- und Verlaufsdetails zu Ihren AWS Ressourcen entsprechend den von Ihnen angegebenen Regeln bereitzustellen. Application Manager integriert ebenfalls in AWS Systems Manager State Manager, um Compliance-Informationen über den Status bereitzustellen, den Sie für Ihre Amazon Elastic Compute Cloud (Amazon EC2)-Instances erhalten möchten.
- Informationen zu Amazon ECS und Amazon EKS Cluster-Infrastruktur anzeigen

Application Manager integriert in Amazon ECS und Amazon EKS, um Informationen über den Zustand Ihrer Cluster-Infrastrukturen und eine Komponentenlaufzeitansicht der Computing-, Netzwerk- und Speicherressourcen in einem Cluster bereitzustellen.

Sie können jedoch keine Betriebsinformationen zu Ihren Amazon EKS-Pods oder -Containern in Application Manager verwalten. Sie können nur Betriebsinformationen zu der Infrastruktur verwalten und anzeigen, die Ihre Amazon EKS-Ressourcen hostet.

- Anzeigen von Ressourcen-Preisdetails für eine Anwendung

  Application Manager ist über das Cost-Widget in AWS Cost Explorer, einem Feature von AWS Billing and Cost Management Cost Management, integriert. Nachdem Sie den Cost Explorer in der Fakturierungs- und Kostenmanagement-Konsole aktiviert haben, zeigt das Cost-Widget in Application Manager Preisdaten für eine bestimmte Anwendung oder Anwendungskomponente ohne Container an. Sie können Filter im Widget verwenden, um Preisdaten nach verschiedenen Zeiträumen, Details und Preisarten in einem Balken- oder Liniendiagramm anzuzeigen.
- Anzeigen detaillierter Ressourceninformationen in einer Konsole

  Wählen Sie einen Ressourcennamen aus, der unter Application Manager gelistet ist und zeigen Sie kontextbezogene Informationen und Betriebsinformationen zu dieser Ressource an, ohne Systems Manager verlassen zu müssen.
- Erhalten Sie automatische Ressourcenaktualisierungen für Anwendungen

  Wenn Sie Änderungen an einer Ressource in einer Dienstkonsole vornehmen und diese Ressource Teil einer Anwendung in Application Manager ist, werden diese Änderungen automatisch von Systems Manager angezeigt. Wenn Sie beispielsweise einen Stack in der AWS CloudFormation Konsole aktualisieren und wenn dieser Stack Teil einerApplication Manager Anwendung ist, werden die Stack-Updates automatisch in Application Manager übertragen.
- Entdecken Sie Launch Wizard-Anwendungen automatisch

  Application Manager ist in AWS Launch Wizard integriert. Wenn Sie den Launch Wizard zum Bereitstellen von Ressourcen für eine Anwendung verwendet haben, kann Application Manager diese automatisch importieren und in einem Abschnitt des Launch Wizard anzeigen.
- Überwachen von Anwendungsressourcen in Application Manager durch Verwendung von CloudWatch Application Insights

  Application Manager integriert Amazon CloudWatch Application Insights Application Insights identifiziert Schlüsselmetriken, Protokolle und Alarme und richtet diese für Ihre

Anwendungsressourcen und Ihren Technologie-Stack ein. Application Insights überwacht kontinuierlich Metriken und Protokolle, um Anomalien und Fehler zu erkennen und zu korrelieren. Wenn das System Fehler oder Anomalien erkennt, generiert Application Insights CloudWatch Events, mit denen Sie Benachrichtigungen einrichten oder Aktionen ausführen können. Sie können Application Insights auf den Tabs Übersicht und Überwachungin Application Manager aktivieren und ansehen. Weitere Informationen zu Application Insights finden Sie unter Was ist Amazon CloudWatch Application Insights im Amazon CloudWatch-Benutzerhandbuch.

- Beheben von Problemen mit Runbooks

  Application Manager enthält vordefinierte Systems Manager-Runbooks zur Behebung häufiger Probleme mit AWS Ressourcen. Sie können ein Runbook für alle anwendbaren Ressourcen in einer Anwendung ausführen, ohne Application Manager verlassen zu müssen.

Entstehen Kosten für die Verwendung von Application Manager?

Application Manager ist ohne Aufpreis erhältlich.

Was sind die Ressourcenkontingente für Application Manager?

Sie können Kontingente für alle Systems Manager Funktionen in den Service Quotas für Systems Manager in der Allgemeinen Allgemeine Amazon Web Services-Referenz einsehen. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region.

Themen

- Erste Schritte mit Systems Manager Application Manager
- Arbeiten mit Application Manager

Erste Schritte mit Systems Manager Application Manager

Verwenden Sie die Informationen in diesem Abschnitt als Unterstützung für die Einrichtung und Konfiguration von Application Manager, eine Funktion vonAWS Systems Manager, um Betriebsinformationen aus verschiedenen AWS-Services und Systems-Manager-Funktionen anzuzeigen. Dieser Abschnitt enthält auch Informationen zum Hinzufügen von Anwendungen und Clustern zu Application Manager.

Themen

- Einrichten von zugehörigen Services

- Konfigurieren von Berechtigungen für Systems Manager Application Manager
- Hinzufügen von Anwendungen und Clustern zu Application Manager

Einrichten von zugehörigen Services

Application Manager, eine Funktion von AWS Systems Manager, zeigt Ressourcen und Informationen aus anderen AWS-Services und Systems-Manager-Funktionen. Um die Menge der Arbeitsvorgangsinformationen zu maximieren, die in Application Manager angezeigt werden, empfehlen wir, diese anderen Services oder Funktionen einzurichten und zu konfigurieren bevor Sie Application Manager verwenden.

Themen

- Einrichten von Aufgaben zum Importieren von Ressourcen
- Einrichten von Aufgaben zum Anzeigen von Vorgangsinformationen zu Ressourcen

Einrichten von Aufgaben zum Importieren von Ressourcen

Mit den folgenden Einrichtungsaufgaben können Sie AWSRessourcen in Application Manager einsehen. Nachdem jede dieser Aufgaben abgeschlossen ist, kann Systems Manager Ressourcen automatisch in Application Manager importieren. Nachdem Ihre Ressourcen importiert wurden, können Sie Anwendungen in Application Manager erstellen und Ihre importierten Ressourcen in sie verschieben. So können Sie Betriebsinformationen im Kontext einer Anwendung anzeigen.

(Optional) Organisieren Sie Ihre AWS-Ressourcen mithilfe von Tags

Sie können Metadaten Ihren AWS-Ressourcen in Form von Tags zuweisen. Jedes Tag ist ein Label, das aus einem benutzerdefinierten Schlüssel und Wert besteht. Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern. Sie können Tags erstellen, um Ressourcen nach Zweck, Eigentümer, Umgebung oder anderen Kriterien zu kategorisieren.

(Optional) Organisieren Sie Ihre AWS-Ressourcen mithilfe von AWS Resource Groups

Sie können mit Ressourcengruppen Ihre AWS-Ressourcen organisieren. Ressourcengruppen vereinfachen die gleichzeitige Verwaltung, Überwachung und Automatisierung von Aufgaben für viele Ressourcen.

Application Manager importiert automatisch alle Ressourcengruppen und listet sie in der Kategorie Benutzerdefinierte Anwendungen.

(Optional) Richten Sie Ihre AWS Ressourcen mithilfe von AWS CloudFormation ein und stellen Sie diese bereitstellen

AWS CloudFormation ermöglicht es Ihnen, die AWS-Infrastrukturen vorhersagbar und wiederholt zu erstellen und bereitzustellen. Es hilft Ihnen bei der Verwendung von AWS-Services wie Amazon EC2, Amazon Elastic Block Store (Amazon EBS), Amazon Simple Notification Service (Amazon SNS), Elastic Load Balancing, und AWS-Auto-Scaling. Mit CloudFormation können Sie zuverlässige, skalierbare und kostengünstige Anwendungen in der Cloud erstellen, ohne sich Gedanken über die Erstellung und Konfiguration des zugrunde liegenden AWS-Infrastruktur machen zu müssen.

Application Manager importiert automatisch alle AWS CloudFormation-Ressourcen und listet sie in der Kategorie AWS CloudFormation Stacks. Sie können CloudFormation-Stacks und -Vorlagen in Application Manager erstellen. Stack- und Template-Änderungen werden automatisch zwischen Application Manager und CloudFormation synchronisiert. Sie können auch Anwendungen in Application Manager erstellen und Stacks in diese verschieben. Auf diese Weise können Sie Betriebsinformationen für Ressourcen in Ihren Stacks im Kontext einer Anwendung anzeigen. Preisinformationen finden Sie unter AWS CloudFormation – Preise.

(Optional) Richten Sie Ihre Anwendungen mithilfe von AWS Launch Wizard ein und stellen Sie diese bereitstellen

Der Launch Wizard führt Sie durch die Prozesse Größenanpassung, Konfigurieren und Bereitstellen von AWS Ressourcen für Anwendungen von Drittanbietern ohne manuelle Identifizierung und Bereitstellung einzelner AWS-Ressourcen.

Application Manager importiert automatisch all Ihre Launch Wizard-Ressourcen und listet sie in der Kategorie Launch Wizard. Weitere Informationen zu AWS Launch Wizard finden Sie unter Erste Schritte mit AWS Launch Wizard für SQL-Server. Launch Wizard ist ohne Aufpreis erhältlich. Sie zahlen nur für die AWS-Ressourcen, die Sie für die Ausführung Ihrer Lösung bereitstellen.

(Optional) Richten Sie Ihre containerisierten Anwendungen mithilfe von Amazon ECS und Amazon EKS ein und stellen diese bereit.

Amazon Elastic Container Service (Amazon ECS) ist ein hoch skalierbarer, schneller Container-Management-Service, der das Ausführen, Beenden und Verwalten von Containern in einem Cluster vereinfacht. Ihre Container sind in einer Aufgabendefinition definiert, die Sie zum Ausführen einzelner Aufgaben oder Aufgaben innerhalb eines Dienstes verwenden.

Amazon EKS ist ein verwalteter Service, der Ihnen hilft, Kubernetes auf AWS auszuführen, ohne Ihre eigene Kubernetes-Steuerebene oder -Knoten zu installieren, zu betreiben und zu warten. Kubernetes ist ein Open-Source-System zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von Anwendungen in Containern.

Application Manager importiert automatisch alle Ihre Amazon ECS- und Amazon EKS-Infrastrukturressourcen und listet sie im Container-Cluster-Tab auf. Sie können jedoch keine Betriebsinformationen zu Ihren Amazon EKS-Pods oder -Containern in Application Manager verwalten. Sie können nur Betriebsinformationen zu der Infrastruktur verwalten und anzeigen, die Ihre Amazon EKS-Ressourcen hostet. Weitere Preisinformationen finden Sie unter Amazon ECS Preis und Amazon EKS Preis.

Einrichten von Aufgaben zum Anzeigen von Vorgangsinformationen zu Ressourcen

Die folgenden Setup-Aufgaben helfen Ihnen beim Anzeigen von Betriebsinformationen über Ihre AWS-Ressourcen in Application Manager.

(Empfohlen) Verifizieren Sie Runbook-Berechtigungen

Sie können Probleme mit AWS-Ressourcen von Application Manager mithilfe von Systems Manager Automation Runbooks beheben. Um diese Funktion zum Beheben zu verwenden, müssen Sie Berechtigungen konfigurieren oder überprüfen. Preisinformationen finden Sie unter AWS Systems Manager – Preise.

(Optional) Aktivieren Sie Cost Explorer

AWS Cost Explorer ist ein Feature von AWS Cost Management, mit dem Sie Ihre Kostendaten für weitere Analysen visualisieren können. Wenn Sie den Cost Explorer aktivieren, können Sie in der Application Manager-Konsole Kosteninformationen, den Kostenverlauf und die Kostenoptimierung für die Ressourcen Ihrer Anwendung einsehen.

(Optional) Einrichten und Konfigurieren von Amazon CloudWatch -Protokollenund -Alarmen

CloudWatch ist ein Überwachungs- und Verwaltungsdienst, der Daten und durchführbare Einblicke für AWS, Hybrid- und Multi-Cloud-Anwendungen und Infrastrukturressourcen liefert. Mit CloudWatch können Sie alle Ihre Leistungs- und Betriebsdaten in Form von Protokollen und Metriken von einer einzigen Plattform aus erfassen und darauf zugreifen. Um CloudWatch Protokolle und Alarme für Ihre Ressourcen in Application Manager anzuzeigen, müssen Sie CloudWatch einrichten und konfigurieren. Preisinformationen finden Sie unter CloudWatch-Preisinformationen.

> **Note**
>
> Die Unterstützung von CloudWatch Logs gilt nur für Anwendungen, nicht für Cluster.

(Optional) Einrichten und Konfigurieren von AWS Config

AWS Config bietet eine detaillierte Übersicht über die Ressourcen, die Ihrem AWS-Konto zugeordnet sind, einschließlich der Informationen darüber, wie sie konfiguriert sind, wie sie zueinander in Beziehung stehen und wie sich die Konfigurationen und ihre Beziehungen im Zeitverlauf geändert haben. Mit AWS Config werten Sie die Konfigurationseinstellungen Ihrer AWS-Ressourcen aus. Dazu erstellen Sie AWS Config-Regeln, die Ihre idealen Konfigurationseinstellungen darstellen. Während AWS Config die Konfigurationsänderungen bei allen Ihren Ressourcen kontinuierlich verfolgt, wird überprüft, ob diese Änderungen gegen eine Bedingung Ihrer Regeln verstoßen. Wenn eine Ressource gegen eine Regel verstößt, kennzeichnet AWS Config die Ressource und die Regel als nicht regelkonform. Application Manager zeigt Compliance-Informationen über AWS Config-Regeln an. Um diese Daten in Application Manager anzuzeigen, müssen Sie AWS Config einstellen und konfigurieren. Preisinformationen finden Sie unter AWS Config – Preise.

(Optional) Erstellen Sie State Manager Zuordnungen

Sie können Systems Manager State Manager verwenden, um eine Konfiguration zu erstellen, die Ihren verwalteten Knoten zugewiesen wird. Die Konfiguration, auch Zuordnung genannt, definiert den Zustand, den Sie auf Ihren Knoten beibehalten möchten. Um Zuordnungs-Compliance-Daten in Application Manager anzuzeigen, müssen Sie eine oder mehrere State Manager-Zuordnungen konfigurieren. State Manager wird ohne Aufpreis angeboten.

(Optional) Einrichten und Konfigurieren von OpsCenter

Sie können operative Arbeitselemente (OpsItems) über Ihre Ressourcen in Application Manager durch Verwendung von OpsCenter anzeigen. Sie können Amazon CloudWatch und Amazon EventBridge so konfigurieren, dass sie automatisch OpsItems auf OpsCenter basierend auf Alarmen und Ereignissen senden. Sie können die Schlüssel auch manuell OpsItems eingeben. Preisinformationen finden Sie unter AWS Systems Manager – Preise.

Konfigurieren von Berechtigungen für Systems Manager Application Manager

Sie können alle Funktionen von Application Manager nutzen, eine Funktion von AWS Systems Manager, wenn Ihre AWS Identity and Access Management (IAM)-Benutzer, -Gruppen oder -Rollen

über Zugriff auf die in diesem Thema aufgeführten API-Operationen verfügen. Die API-Operationen sind in zwei Tabellen unterteilt, um Ihnen zu helfen, die verschiedenen Funktionen zu verstehen, die sie ausführen.

In der folgenden Tabelle sind die API-Vorgänge aufgeführt, die Systems Manager aufruft, wenn Sie eine Ressource in Application Manager wählen, da Sie die Ressourcendetails anzeigen möchten. Zum Beispiel, wenn Application Manager eine Amazon EC2 Auto Scaling Gruppe auflistet und wenn Sie diese Gruppe auswählen, um ihre Details anzuzeigen, ruft Systems Manager die `autoscaling:DescribeAutoScalingGroups`-API-Operationen auf. Wenn Sie keine Auto Scaling Gruppen in Ihrem Konto haben, wird dieser API-Vorgang nicht von Application Manager ausgeführt.

Ausschließlich Ressourcendetails

```
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
```

Ausschließlich Ressourcendetails

```
elasticbeanstalk:ListTagsForResource
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
```

In der folgenden Tabelle sind die API-Vorgänge aufgeführt, die Systems Manager verwendet, um Änderungen an Anwendungen und Ressourcen vorzunehmen, die unter Application Manager gelistet sind oder um Vorgangsinformationen für eine ausgewählte Anwendung oder Ressource anzuzeigen.

Aktionen und Details der Anwendung

```
applicationinsights:CreateApplication
applicationinsights:DescribeApplication
applicationinsights:ListProblems
ce:GetCostAndUsage
ce:GetTags
ce:ListCostAllocationTags
ce:UpdateCostAllocationTagsStatus
cloudformation:CreateStack
```

Aktionen und Details der Anwendung

```
cloudformation:DeleteStack
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
```

Aktionen und Details der Anwendung

```
iam:CreateServiceLinkedRole
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
```

Aktionen und Details der Anwendung

```
ssm:ListTagsForResource
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
```

Konfigurieren von Berechtigungen

Um Application Manager-Berechtigungen für eine IAM-Entität (z. B. einen Benutzer, eine Gruppe oder eine Rolle) zu konfigurieren, erstellen Sie anhand des folgenden Beispiels eine IAM-Richtlinie. Dieses Richtlinienbeispiel enthält alle API-Vorgänge, die von Application Manager verwendet werden.

```
                {
    "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

Sie können die Fähigkeit eines Benutzers einschränken, Änderungen an Anwendungen und Ressourcen in Application Manager vorzunehmen, indem Sie die folgenden API-Operationen aus der IAM-Berechtigungsrichtlinie entfernen, die ihrem Benutzer, ihrer Gruppe oder ihrer Rolle zugeordnet ist. Durch entfernen dieser Aktionen, steht nur der read-only-Modus in Application Manager zur Verfügung. Nachfolgend finden Sie alle APIs, mit denen Benutzer Änderungen an der Anwendung oder an anderen verwandten Ressourcen vornehmen können.

```
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
```

Informationen zum Erstellen und Bearbeiten von IAM-Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch. Informationen zum Zuweisen dieser Richtlinie zu einer IAM-Entität (z. B. einem Benutzer, einer Gruppe oder einer Rolle) finden Sie unter Hinzufügen und Entfernen von IAM-Identitätsberechtigungen.

Hinzufügen von Anwendungen und Clustern zu Application Manager

Application Manager ist eine Komponente von AWS Systems Manager. In Application Manager ist eine Anwendung eine logische Gruppierung von AWS-Ressourcen, die Sie als Einheit betreiben möchten. Diese logische Gruppe kann verschiedene Versionen einer Anwendung, Besitzgrenzen für Operatoren oder Entwicklerumgebungen darstellen, um nur einige zu nennen.

Wenn Sie Get started (Erste Schritte) auf der Startseite von Application Manager auswählen, importiert Application Manager automatisch Metadaten zu Ihren Ressourcen, die in anderen

AWS-Services- oder Systems-Manager-Funktionen erstellt wurden. Für Anwendungen importiert Application Manager Metadaten über alle AWS Ressourcen, die in Ressourcengruppen organisiert sind. Jede Ressourcengruppe wird in der Kategorie Custom applications (Benutzerdefinierte Anwendungen) als einzigartige Anwendung gelistet. Application Manager importiert automatisch Metadaten zu Ressourcen, die von AWS CloudFormation, AWS Launch Wizard, Amazon Elastic Container Service (Amazon ECS) und Amazon Elastic Kubernetes Service (Amazon EKS). Application Manager zeigt diese Ressourcen dann in vordefinierten Kategorien an.

Für Anwendungen umfasst die Liste Folgendes:

- Benutzerdefinierte Anwendungen
- Launch Wizard
- CloudFormation-Stacks
- AppRegisty-Anwendungen

Für Container-Cluster umfasst die Liste Folgendes:

- Amazon ECS-Cluster
- Amazon EKS-Cluster

Nach Abschluss des Imports können Sie Vorgangsinformationen für eine Anwendung oder eine bestimmte Ressource in diesen vordefinierten Kategorien anzeigen. Wenn Sie mehr Kontext zu einer Ressourcensammlung bereitstellen möchten, können Sie eine Anwendung manuell in Application Manager erstellen. Anschließend können Sie Ressourcen oder Ressourcengruppen zu dieser Anwendung hinzufügen. Nachdem Sie eine Anwendung in Application Manager erstellt haben, können Sie Betriebsinformationen zu Ihrer Ressource im Kontext einer Anwendung anzeigen.

Erstellen einer Anwendung in Application Manager

Gehen Sie wie folgt vor, um eine Anwendung in Application Manager zu erstellen und um Ressourcen zu dieser Anwendung hinzuzufügen.

So erstellen Sie eine Anwendung in Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.

3. Wählen Sie die Registerkarte Anwendungen und dann Anwendung erstellen aus.
4. Geben Sie für den Anwendungsnamen einen Namen ein, um den Zweck der Ressourcen zu verstehen, die dieser Anwendung hinzugefügt werden.
5. Geben Sie unter Anwendungsbeschreibung Informationen zur Anwendung ein.
6. In der Sektion Auswählen von Anwendungskomponenten verwenden Sie die bereitgestellten Optionen, um Ressourcen für diese Anwendung auszuwählen. Sie können einer Anwendung eine Kombination aus getaggten Ressourcen, Ressourcengruppen und Stacks hinzufügen. Sie müssen ein Minimum von zwei Komponenten und ein Maximum von 15 wählen. Wenn Sie Ressourcen mithilfe von Tags auswählen, werden alle Ressourcen, die diesen Tags zugewiesen sind, auf der Registerkarte Ressourcen gelistet, nachdem Sie die neue Anwendung hinzugefügt haben. Dies gilt auch für Ressourcen, die in einer Ressourcengruppe oder in einem Stack enthalten sind.

   Wenn die Ressourcen, die Sie der Anwendung hinzufügen möchten, nicht angezeigt werden, stellen Sie sicher, dass die Ressourcen ordnungsgemäß markiert wurden und zu einer AWS Resource Groups-Gruppe oder zum einem AWS CloudFormation-Stack hinzugefügt wurden.
7. Spezifizieren Sie Tags für diese Anwendung für Anwendungs-Tags - optional.
8. Wählen Sie Erstellen aus.

Application Manager erstellt die Anwendung und öffnet sie. Der Komponenten-Baum listet die neue Anwendung als Komponente der obersten Ebene und die Ressourcen, Gruppen oder Stacks auf, die Sie als Unterkomponenten ausgewählt haben. Wenn Sie das nächste Mal Application Manager öffnen, finden Sie die neue Anwendung in der Kategorie Benutzerdefinierte Anwendungen.

Arbeiten mit Application Manager

Application Manager ist eine Komponente von AWS Systems Manager. Dieser Abschnitt enthält Themen, die Ihnen bei der Verwendung vonApplication Manager-Anwendungen und -Cluster helfen und helfen, Betriebsinformationen über Ihre AWS-Ressourcen einzusehen.

Inhalt

- Arbeiten mit -Anwendungen
- Arbeiten mit AWS CloudFormation-Vorlagen und Stacks in Application Manager
- Arbeiten mit Clustern in Application Manager

Arbeiten mit -Anwendungen

Application Manager ist eine Komponente von AWS Systems Manager. Dieser Abschnitt enthält Themen, die Ihnen bei der Verwendung vonApplication Manager-Anwendungen helfen und helfen, Betriebsinformationen über Ihre AWS-Ressourcen einzusehen.

Inhalt

- Anzeigen von Übersichtsinformationen einer Anwendung
- Arbeiten mit Ihren Anwendungs-Instances
- Markieren von Anwendungsressourcen
- Anzeigen von Compliance-Informationen
- Anzeigen von Überwachungsinformationen
- Anzeigen von OpsItems für eine Anwendung
- Anzeigen von Protokollgruppen und Protokolldaten
- Arbeiten mit Runbooks in Application Manager
- Arbeiten mit Tags in Application Manager

Anzeigen von Übersichtsinformationen einer Anwendung

In Application Manager, eine Komponente von AWS Systems Manager, zeigt der Reiter Übersicht eine Zusammenfassung der Amazon CloudWatch Alarme, betriebliche Arbeitselemente (OpsItems), CloudWatch Application Insights und Runbook-Verlauf. Klicken Sie auf Alle anzeigen (View all) für jede Karte, um die entsprechende Registerkarte zu öffnen, auf der Sie alle Anwendungseinblicke, Alarme, OpsItems, oder den Runbook-Verlauf einsehen können.

Informationen zu Application Insights

CloudWatch Application Insights identifiziert Schlüsselmetriken, Protokolle und Alarme und richtet diese für Ihre Anwendungsressourcen und Ihren Technologie-Stack ein. Application Insights überwacht kontinuierlich Metriken und Protokolle, um Anomalien und Fehler zu erkennen und zu korrelieren. Wenn das System Fehler oder Anomalien erkennt, generiert Application Insights CloudWatch Events, mit denen Sie Benachrichtigungen einrichten oder Aktionen ausführen können. Wenn Sie die Schaltfläche Konfiguration bearbeiten auf der Registerkarte Überwachung wählen, öffnet das System die CloudWatch Application Insights-Konsole. Weitere Informationen zu Application Insights finden Sie unter Was ist Amazon CloudWatch Application Insights im Amazon CloudWatch-Benutzerhandbuch.

Über Kosten-Explorer

Application Manager ist über das Cost-Widget und die Registerkarte Cost mit AWS Cost Explorer, einem Feature von AWS Cost Management, integriert. Nachdem Sie den Cost Explorer in der Kostenmanagement-Konsole aktiviert haben, zeigt das Cost-Widget und die Registerkarte Cost in Application Manager die Kostendaten für eine bestimmte Nicht-Container-Anwendung oder Anwendungskomponente an. Sie können Filter im Widget oder der Registerkarte verwenden, um Preisdaten nach verschiedenen Zeiträumen, Details und Preisarten in einem Balken- oder Liniendiagramm anzuzeigen.

Sie können dieses Feature aktivieren, indem Sie die Schaltfläche Go to AWS Cost Management console wählen. Standardmäßig werden die Daten auf die letzten drei Monate gefiltert. Wenn Sie für eine Nicht-Container-Anwendung die Schaltfläche View all (Alle anzeigen) wählen, öffnet Application Manager die Registerkarte Resources (Ressourcen). Für Container-Anwendungen öffnet die Schaltfläche View all (Alle anzeigen) die AWS Cost Explorer-Konsole.

Aktionen, die Sie auf dieser Seite ausführen können

Auf der Registerkarte Overview (Übersicht) auf dieser Seite können Sie Informationen zu den folgenden Widgets aktivieren und abrufen. Wenn ein Widget aktiviert ist, wählen Sie dessen View all (Alle anzeigen) aus, um relevante Anwendungsdetails für diesen Bereich anzuzeigen.

- Wählen Sie im Abschnitt Insights and Alarms (Erkenntnisse und Alarme) die Zahl für einen Schweregrad aus, um die Registerkarte Monitoring (Überwachung) zu öffnen, auf der Sie weitere Details zu Alarmen des ausgewählten Schweregrads anzeigen können.
- Wählen Sie im Abschnitt Cost (Kosten) die Option View all (Alle anzeigen) aus, um die Registerkarte Resources (Ressourcen) zu öffnen, auf der Sie Kostendaten für eine bestimmte Anwendung oder Anwendungskomponente anzeigen können.
- Wählen Sie im Abschnitt Compliance die Option View all (Alle anzeigen) aus, um die Registerkarte Compliance zu öffnen, auf der Sie Compliance-Informationen von AWS Config und State Manager-Zuordnungen anzeigen können.

> **Note**
>
> Um Patch-Compliance-Details anzuzeigen, wählen Sie direkt die Registerkarte Compliance aus. Anschließend können Sie Patch-Compliance-Details für die verwalteten Knoten anzeigen, die von der ausgewählten Anwendung verwendet werden.

- Wählen Sie in der Sektion Runbooks ein Runbook aus, um es auf der Seite Dokumente des Systems Managers, auf der Sie weitere Details zum Dokument anzeigen können, zu öffnen.
- Wählen Sie in der Sektion OpsItems einen Schweregrad aus, um die Registerkarte OpsItems zu öffnen, auf der Sie alle OpsItems des gewählten Schweregrads einsehen können.
- Wählen Sie eine Alle anzeigen-Schaltfläche, um die entsprechende Registerkarte zu öffnen. Sie können alle Alarm-, OpsItems oder Runbook-Verlaufeinträge für die Anwendung anzeigen.

So öffnen Sie die Registerkarte Overview (Übersicht)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.

Arbeiten mit Ihren Anwendungs-Instances

Application Manager lässt sich in Amazon Elastic Compute Cloud (Amazon EC2) integrieren, um Informationen zu Ihren Instances im Kontext einer Anwendung anzuzeigen. Application Manager zeigt Instance-Status, Status und den Zustand von Amazon EC2 Auto Scaling für eine ausgewählte Anwendung in einem grafischen Format an. Die Registerkarte Instances enthält auch eine Tabelle mit den folgenden Informationen für jede Instance in Ihrer Anwendung:

- Instance-Status (Ausstehend, Angehalten, Wird ausgeführt, Beendet)
- Ping-Status für SSM Agent
- Status und Name des letzten Systems-Manager-Automation-Runbooks, das auf der Instance verarbeitet wurde
- Eine Anzahl von Amazon- CloudWatch Logs-Alarmen pro Status.
  - `ALARM` – Die Metrik oder der Ausdruck liegt außerhalb des festgelegten Schwellenwerts.
  - `OK` – Die Metrik oder der Ausdruck liegt innerhalb des festgelegten Schwellenwerts.
  - `INSUFFICIENT_DATA` – Der Alarm wurde soeben gestartet; die Metrik ist nicht verfügbar oder es sind nicht genügend Daten verfügbar, damit die Metrik den Alarmstatus bestimmen kann.

- Zustand der Auto-Scaling-Gruppe für die übergeordneten und einzelnen Auto-Scaling-Gruppen

Wenn Sie eine Instance in der Tabelle All instances (Alle Instances) auswählen, zeigt Application Manager Informationen zu dieser Instance auf vier Registerkarten an:

- Details – Alle Instance-Details von Amazon EC2, einschließlich Amazon Machine Image (AMI), DNS-Informationen, IP-Adressinformationen und mehr.
- Health (Zustand) – Der aktuelle Status, wie er von EC2-System- und Instance-Statusprüfungen bereitgestellt wird.
- Execution history (Ausführungshistorie) – Ausführungsprotokolle für Systems-Manager-Automation-Runbooks und API-Aufrufe, die von der Instance verarbeitet werden.
- CloudWatch Alarme – Der Name, der Status und mehr für Alarme, die von der Instance CloudWatch ausgelöst werden.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Instances starten, anhalten und beenden.
- Wenden Sie ein Chef Rezept an.
- Fügen Sie Instances einer Auto-Scaling-Gruppe hinzu oder trennen Sie Instances von einer Auto-Scaling-Gruppe.
- Aktivieren Sie automatische Updates für SSM Agent.

So öffnen Sie die Registerkarte Instances

1. Öffnen Sie die - AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Custom applications (Benutzerdefinierte Anwendungen).
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Instances aus.

So zeigen Sie die Details Ihrer Anwendungs-Instances an

1. Öffnen Sie die - AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Custom applications (Benutzerdefinierte Anwendungen).
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Instances aus.
6. Wählen Sie die Schaltfläche neben der Instance aus, deren Details Sie anzeigen möchten.
7. Überprüfen Sie die Instance-Details unten auf der Seite.

So aktualisieren Sie SSM Agent automatisch

1. Öffnen Sie die - AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Custom applications (Benutzerdefinierte Anwendungen).
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Instances aus.
6. Wählen Sie in der Dropdownliste Agentenaktionen die Option SSM Agent-Update konfigurieren aus.
7. Wählen Sie Alle Instances aus, um automatische SSM Agent-Updates für alle verwalteten Instances zu konfigurieren. Wählen Sie alternativ Instance, um automatische SSM Agent-Updates für eine einzelne Instance in Ihrer Anwendung zu konfigurieren.
8. Wählen Sie den Schalter Automatische Updates aktivieren aus.
9. Wählen Sie in der Dropdownliste Zeitplan angeben den Zeitplan aus, den Sie für SSM Agent-Updates verwenden möchten.

10. Wählen Sie Konfigurieren.

Markieren von Anwendungsressourcen

In Application Manager, eine Komponente von AWS Systems Manager, zeigt die Registerkarte Ressourcen die AWS-Ressourcen in Ihrer Anwendung an. Wenn Sie eine Komponente der obersten Ebene auswählen, werden auf dieser Seite alle Ressourcen für diese Komponente und alle Unterkomponenten angezeigt. Wenn Sie eine Unterkomponente auswählen, werden auf dieser Seite nur die Ressourcen angezeigt, die dieser Unterkomponente zugewiesen sind.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie einen Ressourcennamen aus, um Informationen darüber anzuzeigen, einschließlich Details, die von der Konsole bereitgestellt wurden, auf der die Ressource erstellt wurde, Tags, Amazon CloudWatch Alarme, AWS Config-Details und AWS CloudTrail-Protokollinformationen.
- Wählen Sie die Optionsschaltfläche neben einem Ressourcennamen. Wählen Sie dann die Schaltfläche Zeitplan der Ressource, um die AWS Config-Konsole zu öffnen, in der Sie Compliance-Informationen zu einer ausgewählten Ressource anzeigen können.
- Wenn Sie AWS Cost Explorer aktiviert haben, zeigt der Abschnitt Cost Explorer Preisdaten für eine bestimmte Nicht-Container-Anwendung oder -Anwendungs-Komponente an. Sie können diese Funktion aktivieren, indem Sie die Schaltfläche Go to AWS Cost Management console wählen. Verwenden Sie die Filter in diesem Abschnitt, um Preisinformationen zu Ihrer Anwendung anzuzeigen.

So öffnen Sie die Registerkarte Resources (Ressourcen)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.

5. Wählen Sie die Registerkarte Resources (Ressourcen) aus.

Anzeigen von Compliance-Informationen

In Application Manager, eine Komponente von AWS Systems Manager, zeigt die Seite Configurations (Konfigurationen) Informationen zur Compliance von Regeln für AWS Config-Ressourcen und -Konfigurationen an. Auf dieser Seite werden auch AWS Systems Manager State Manager Compliance-Informationen angezeigt. Sie können eine Ressource, eine Regel oder eine Zuordnung auswählen, um die entsprechende Konsole für weitere Informationen zu öffnen. Auf dieser Seite werden die Compliance-Informationen der letzten 90 Tage angezeigt.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie einen Ressourcennamen, um die AWS Config-Konsole zu öffnen, in der Sie Compliance-Informationen zu einer ausgewählten Ressource anzeigen können.
- Wählen Sie die Optionsschaltfläche neben einem Ressourcennamen. Wählen Sie dann die Schaltfläche Zeitplan der Ressource, um die AWS Config-Konsole zu öffnen, in der Sie Compliance-Informationen zu einer ausgewählten Ressource anzeigen können.
- In der Sektion Compliance-Regeln können Sie zudem Folgendes durchführen:
  - Wählen Sie einen Namen aus, um die AWS Config-Konsole zu wählen, auf der Sie Informationen über diese Regel anzeigen können.
  - Wählen Sie Hinzufügen von Regeln, um die AWS Config-Konsole zu öffnen, in der Sie eine Regel erstellen können.
  - Wählen Sie die Optionsschaltfläche neben einem Regelnamen, wählen Sie Aktionen und wählen Sie dann Verwalten der Behebung, um die Behebungsaktion für eine Regel zu ändern.
  - Wählen Sie die Optionsschaltfläche neben einem Regelnamen, wählen Sie Aktionen und wählen Sie dann Erneut bewerten, um AWS Config eine Compliance-Überprüfung für die gewählte Regel auszuführen.
- In der Sektion Association compliance können Sie zudem Folgendes durchführen:
  - Wählen Sie einen Zuordnungsnamen aus, um die Seite Associations zu öffnen, wo Sie Informationen über diese Assoziation einsehen können.
  - Wählen Sie Erstellen einer Zuordnung, um Systems Manager zu öffnenState Manager, in welchem Sie eine Zuordnung erstellen können.

- Wählen Sie die Optionsschaltfläche neben einem Assoziationsnamen und wählen Sie Zuordnung anwenden, um alle in der Zuordnung angegebenen Aktionen sofort zu starten.

So öffnen Sie die Compliance-Registerkarte

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Compliance-Registerkarte.

Anzeigen von Überwachungsinformationen

InApplication Manager, einer Komponente von AWS Systems Manager, werden auf der Registerkarte Überwachung Amazon CloudWatch Application Insights und Alarmdetails für Ressourcen in einer Anwendung angezeigt.

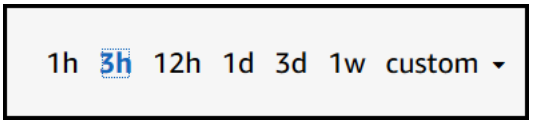
Informationen zu Application Insights

CloudWatch Application Insights identifiziert und richtet wichtige Kennzahlen, Protokolle und Alarme für Ihre Anwendungsressourcen und Ihren Technologie-Stack ein. Application Insights überwacht kontinuierlich Metriken und Protokolle, um Anomalien und Fehler zu erkennen und zu korrelieren. Wenn das System Fehler oder Anomalien erkennt, generiert Application Insights CloudWatch Ereignisse, anhand derer Sie Benachrichtigungen einrichten oder Maßnahmen ergreifen können. Wenn Sie auf der Registerkarte Überwachung auf die Schaltfläche Konfiguration bearbeiten klicken, öffnet das System die CloudWatch Application Insights-Konsole. Weitere Informationen zu Application Insights finden Sie unter Was ist Amazon CloudWatch Application Insights im CloudWatch Amazon-Benutzerhandbuch.

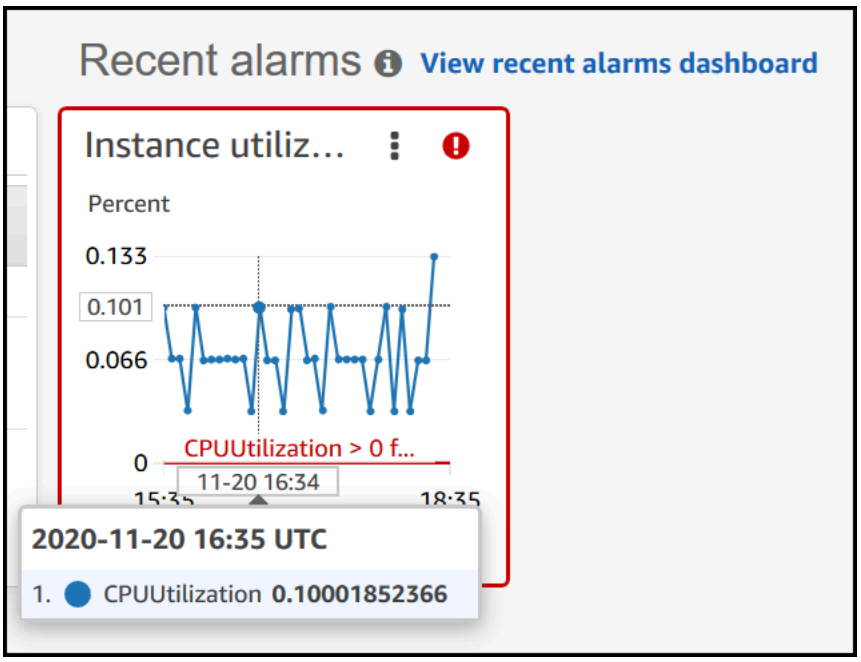
Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

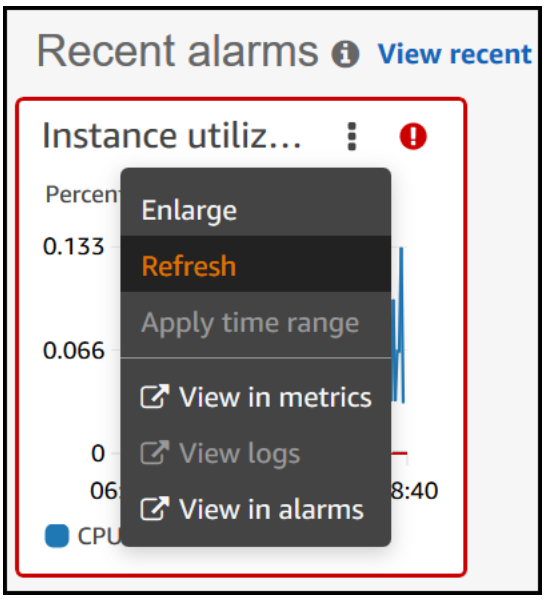
- Wählen Sie im Bereich Alarme nach Service einen AWS Servicenamen aus, CloudWatch um den ausgewählten Service und Alarm zu öffnen.
- Passen Sie den Zeitraum für Daten an, die in Widgets in der Sektion Aktuelle Alarme angezeigt werden, indem Sie einen der vordefinierten Zeitperiodenwerte auswählen. Sie können benutzerdefiniert wählen, um Ihren eigenen Zeitraum zu definieren.



- Bewegen Sie den Cursor über ein Widget in der Sektion Aktuelle Alarme, um ein Datenpop-up für eine bestimmte Zeit anzuzeigen.

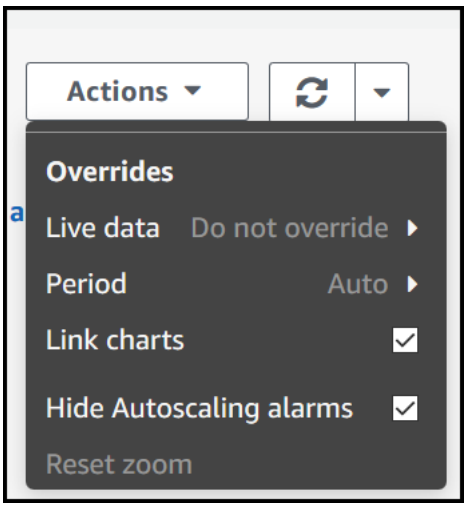


- Wählen Sie das Optionsmenü in einem Widget, um Anzeigeoptionen anzuzeigen. Klicken Sie auf Vergrößern, um ein Widget zu erweitern. Klicken Sie auf Aktualisieren, um die Daten in einem Widget zu aktualisieren. Klicken und ziehen Sie den Cursor in einer Widget-Datenanzeige, um einen bestimmten Bereich auszuwählen. Sie können dann Zeitrahmen auswählen wählen.



- Wählen Sie das Menü Aktionen, um Optionen zum Überschreiben von Alarmdaten anzuzeigen. Folgende Optionen sind verfügbar:
  - Wählen Sie, ob Ihr Widget Live-Daten anzeigt. Live-Daten sind Daten, die innerhalb der letzten Minute veröffentlicht und noch nicht vollständig aggregiert wurden. Wenn Live-Daten deaktiviert sind, werden nur Datenpunkte mit einem Aggregationszeitraum von mindestens einer Minute in der Vergangenheit angezeigt. Bei Verwendung von 5-Minuten-Zeiträumen wird der Datenpunkt für 12:35 von 12:35 zu 12:40 aggregiert und um 12:41 angezeigt.

    Wenn Live-Daten aktiviert sind, wird der neueste Datenpunkt angezeigt, sobald Daten im entsprechenden Aggregationsintervall veröffentlicht werden. Bei jeder Aktualisierung der Anzeige, ändert sich der aktuellste Datenpunkt möglicherweise, wenn neue Daten innerhalb dieses Aggregationszeitraums veröffentlicht werden.
  - Geben Sie einen Zeitraum für Live-Daten an.
  - Verknüpfen Sie die Diagramme in der Sektion Aktuelle Alarme, sodass, wenn Sie ein Diagramm vergrößern oder verkleinern, das andere Diagramm gleichzeitig vergrößert oder verkleinert wird. Sie können die Verknüpfung mit Diagrammen aufheben, um den Zoom auf ein Diagramm zu beschränken.
  - Auto Scaling-Alarme ausblenden.



So öffnen Sie die Registerkarte Monitoring (Überwachung)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Überwachung.

Anzeigen von OpsItems für eine Anwendung

In Application Manager, eine Komponente vonAWS Systems Manager, zeigt die Registerkarte OpsItems operationelle Arbeitselemente (OpsItems) für Ressourcen in der ausgewählten Anwendung an. Sie können Systems Manager OpsCenter so konfigurieren, dass automatisch OpsItems aus Amazon CloudWatch Alarmen und Amazon EventBridge-Ereignisse erstellt werden. Sie können auch manuell OpsItems erstellen.

Aktionen, die Sie auf dieser Registerkarte ausführen können

Hier sind folgende Aktionen möglich:

- Filtern Sie die Liste von OpsItems durch Verwendung des Suchfelds. Sie können nach OpsItem-Name, ID, Quell-ID oder Schweregrad filtern. Sie können die Liste auch basierend auf dem Status filtern. OpsItems unterstützt die folgenden Status: „Offen“, „In Bearbeitung“, „Öffnen und In Bearbeitung“, „Abgeschlossen“ oder „Alle“.
- Ändern des Status einer OpsItem, indem Sie die Optionsschaltfläche daneben auswählen und dann eine Option im Menü Status einstellen wählen.
- Öffnen Sie Systems Manager OpsCenter, um eine OpsItem zu erstellen, indem Sie Create OpsItem wählen.

Um die Registerkarte OpsItems zu öffnen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte OpsItems aus.

Anzeigen von Protokollgruppen und Protokolldaten

In Application Manager, eine Komponente vonAWS Systems Manager, zeigt die Registerkarte Protokolle eine Liste der Protokollgruppen aus Amazon CloudWatch Logs an.

Aktionen, die Sie auf dieser Registerkarte ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie einen Protokollgruppennamen aus, um ihn in CloudWatch Logs zu öffnen. Sie können dann einen Protokolldatenstrom auswählen, um Protokolle für eine Ressource im Kontext einer Anwendung anzuzeigen.
- Klicken Sie auf Erstellen von Protokollgruppen, um eine Protokollgruppe in CloudWatch Logs zu erstellen.

So öffnen Sie die Registerkarte Logs (Protokolle)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Protokolle aus.

Arbeiten mit Runbooks in Application Manager

Sie können Probleme mit AWS-Ressourcen von Application Manager, eine Funktion von AWS Systems Manager, mithilfe von Automation-Runbooks beheben. Ein Automation-Runbook definiert die Aktionen, die Systems Manager auf Ihren verwalteten Instances und AWS-Ressourcen durchführt, wenn eine Automatisierung läuft). Automation ist eine Funktion von AWS Systems Manager. Ein Runbook enthält einen oder mehrere Schritte, die in sequenzieller Reihenfolge ausgeführt werden. Jeder Schritt basiert auf einer einzigen Aktion. Die Ausgabe von einem Schritt kann als Eingabe in einem späteren Schritt verwendet werden.

Wenn Sie Start runbook (Starten von Runbook) von einer Application Manager-Anwendung oder -Cluster wählen, zeigt das System eine gefilterte Liste von verfügbaren Runbooks basierend auf dem Typ der Ressourcen in Ihrer Anwendung oder Ihrem Cluster an. Wenn Sie das Runbook auswählen, das Sie starten möchten, öffnet Systems Manager die Seite Ausführen des Automatisierungsdokuments.

Application Manager umfasst die folgenden Verbesserungen für die Arbeit mit Runbooks.

- Wenn Sie den Namen einer Ressource in Application Manager wählen und dann Runbook ausführen wählen, zeigt das System eine gefilterte Liste von Runbooks für diesen Ressourcentyp an.
- Sie können eine Automatisierung für alle Ressourcen desselben Typs initiieren, indem Sie ein Runbook in der Liste auswählen und dann Für Ressourcen desselben Typs ausführen wählen.

Bevor Sie beginnen

Bevor Sie ein Runbook von Application Managerstarten, gehen Sie wie folgt vor:

- Stellen Sie sicher, dass Sie über die richtigen Berechtigungen zum Starten von Runbooks verfügen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
- Lesen Sie die Dokumentation zur Automatisierungsprozedur zum Starten von Runbooks. Weitere Informationen finden Sie unter Ausführen von Automatisierungen.

So starten Sie ein Runbook aus Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie Runbook starten. Application Manager öffnet das Popup-Fenster Automatisierungs-Widget. Informationen zu den Optionen im Automatisierungs-Widget finden Sie unter Ausführen von Automatisierungen.

Arbeiten mit Tags in Application Manager

Sie können Tags schnell in Anwendungen und AWS-Ressourcen in Application Manager löschen oder hinzufügen. Weitere Informationen zu Tags erhalten Sie unter Markieren von Systems Manager-Ressourcen.

Gehen Sie folgendermaßen vor, um ein Tag zu einer Anwendung hinzuzufügen oder aus einer Anwendung und alle AWS-Ressourcen in dieser Anwendung.

Um ein Tag in einer Anwendung und allen Ressourcen in der Anwendung hinzuzufügen oder zu löschen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.

3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. In der Sektion Anwendungsinformation wählen Sie die Zahl unter Anwendungstags. Wenn der Anwendung keine Tags zugewiesen sind, ist die Zahl Null.
6. Um einen Tag hinzuzufügen, wählen Sie Add new tag (Neuen Tag hinzufügen). Geben Sie einen Schlüssel und einen optionalen Wert ein. Zum Entfernen eines Tags wählen Sie Remove (Entfernen).
7. Wählen Sie Save (Speichern).

Gehen Sie wie folgt vor, um ein Tag einer bestimmten Ressource in Application Manager hinzuzufügen oder zu entfernen.

So fügen Sie ein Tag zu einer Ressource hinzu oder löschen es aus

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Resources (Ressourcen) aus.
6. Wählen Sie einen Ressourcennamen.
7. Klicken Sie im Abschnitt Tags auf Edit (Bearbeiten).
8. Um einen Tag hinzuzufügen, wählen Sie Add new tag (Neuen Tag hinzufügen). Geben Sie einen Schlüssel und einen optionalen Wert ein. Zum Entfernen eines Tags wählen Sie Remove (Entfernen).
9. Wählen Sie Save (Speichern).

Arbeiten mit AWS CloudFormation-Vorlagen und Stacks in Application Manager

Application Manager, eine Funktion von AWS Systems Manager, unterstützt Sie bei der Bereitstellung und Verwaltung von Ressourcen für Ihre Anwendungen durch die Integration in AWS CloudFormation. Sie können AWS CloudFormation-Vorlagen und -Stacks in Application Manager erstellen, bearbeiten und löschen. Bei einem Stack handelt es sich um eine Sammlung von AWS-Ressourcen, die Sie als einzelne Einheit verwalten können. Dies bedeutet, dass Sie eine Sammlung von AWS-Ressourcen mithilfe von CloudFormation Stacks erstellen, aktualisieren oder löschen können. Eine Vorlage ist eine formatierte Textdatei in JSON oder YAML, die die Ressourcen angibt, die Sie in Ihren Stacks bereitstellen möchten.

Application Manager enthält auch eine Vorlagenbibliothek, in der Sie Vorlagen klonen, erstellen und speichern können. Application Manager und CloudFormation zeigen dieselben Informationen über den aktuellen Status eines Stacks an. Vorlagen und Vorlagenaktualisierungen werden in Systems Manager gespeichert, bis Sie den Stapel bereitstellen. Zu diesem Zeitpunkt werden die Änderungen auch in CloudFormation angezeigt.

Nachdem Sie einen Stapel in Application Manager erstellt haben, zeigt die Seite CloudFormation-Stacks hilfreiche Informationen dazu an. Dazu gehört die Vorlage, die zum Erstellen verwendet wurde, eine Anzahl von OpsItems für Ressourcen in Ihrem Stack, der Stack-Status undDrift-Status.

Über Kosten-Explorer

Application Manager ist über das Cost-Widget in AWS Cost Explorer, ein Feature von AWS Cost Management, integriert. Nachdem Sie den Cost Explorer in der Kostenmanagement-Konsole aktiviert haben, zeigt das Cost-Widget in Application Manager Preisdaten für eine bestimmte Anwendung oder Anwendungskomponente ohne Container an. Sie können Filter im Widget verwenden, um Preisdaten nach verschiedenen Zeiträumen, Details und Preisarten in einem Balken- oder Liniendiagramm anzuzeigen.

Sie können dieses Feature aktivieren, indem Sie die Schaltfläche Go to AWS Cost Management console wählen. Standardmäßig werden die Daten auf die letzten drei Monate gefiltert. Wenn Sie für eine Nicht-Container-Anwendung die Schaltfläche View all (Alle anzeigen) wählen, öffnet Application Manager die Registerkarte Resources (Ressourcen). Für Container-Anwendungen öffnet die Schaltfläche View all (Alle anzeigen) die AWS Cost Explorer-Konsole.

> **Note**
>
> Cost Explorer verwendet Tags, um Ihre Anwendungskosten zu verfolgen. Wenn Ihre AWS CloudFormation-Stack-basierte Anwendung nicht mit dem `AppManagerCFNStackKey-`

Tag-Schlüssel konfiguriert ist, kann Cost Explorer keine genauen Kostendaten in Application Manager anzeigen. Wenn der `AppManagerCFNStackKey`-Tag-Schlüssel nicht erkannt wird, werden Sie in der Konsole aufgefordert, das Tag zu Ihrem CloudFormation-Stack hinzuzufügen, um die Kostenverfolgung zu aktivieren. Durch das Hinzufügen wird der Tag-Schlüssel dem Amazon-Ressourcennamen (ARN) Ihres Stacks zugeordnet und das Cost-Widget kann genaue Kostendaten anzeigen.

Important

Das Hinzufügen des `AppManagerCFNStackKey`-Tags löst ein Stack-Update aus. Alle manuellen Konfigurationen, die nach der ursprünglichen Bereitstellung des Stacks vorgenommen wurden, werden nach dem Hinzufügen des Benutzer-Tags nicht mehr berücksichtigt. Weitere Informationen über das Aktualisierungsverhalten von Ressourcen finden Sie unter Aktualisierungsverhalten von Stack-Ressourcen im AWS CloudFormation-Benutzerhandbuch

Bevor Sie beginnen

Verwenden Sie die folgenden Links, um mehr über CloudFormation Konzepte zu erfahren, bevor Sie CloudFormation-Vorlagen und -Stacks mithilfe von Application Manager erstellen bearbeiten oder löschen.

- Was ist AWS CloudFormation?
- Bewährte Methoden für AWS CloudFormation
- Lernen der Grundlagen von Vorlagen
- Arbeiten mit AWS CloudFormation-Stacks
- Arbeiten mit AWS CloudFormation-Vorlagen
- Mustervorlagen

Themen

- Arbeiten mit CloudFormation-Vorlagen
- Arbeiten mit CloudFormation-Stacks

Arbeiten mit CloudFormation-Vorlagen

Application Manager, eine Funktion von AWS Systems Manager, enthält eine Vorlagenbibliothek und andere Tools, mit denen Sie AWS CloudFormation-Vorlagen verwalten können. Dieser Abschnitt enthält folgende Informationen.

Themen

- Arbeiten mit der Vorlagenbibliothek
- Erstellung von Vorlagen
- Bearbeiten einer Vorlage

Arbeiten mit der Vorlagenbibliothek

Die Application Manager-Vorlagenbibliothek bietet Tools, mit denen Sie Vorlagen anzeigen, erstellen, bearbeiten, löschen und klonen können. Sie können Stacks auch direkt aus der Vorlagenbibliothek bereitstellen. Die Vorlagen werden als Systems Manager (SSM) -Dokumente vom Typ `CloudFormation` gespeichert. Wenn Sie Vorlagen als SSM-Dokumente speichern, können Sie Versionskontrollen verwenden, um mit verschiedenen Versionen einer Vorlage zu arbeiten. Sie können auch Berechtigungen festlegen und Vorlagen teilen. Nachdem Sie einen Stack erfolgreich bereitgestellt haben, sind der Stack und die Vorlage in Application Manager und CloudFormation verfügbar.

Bevor Sie beginnen

Es wird empfohlen, die folgenden Themen zu lesen, um mehr über SSM-Dokumente zu erfahren, bevor Sie mit dem Arbeiten mit CloudFormation-Vorlagen in Application Manager beginnen.

- AWS Systems Manager-Documents
- Freigeben von SSM-Dokumenten
- Bewährte Methoden für freigegebene SSM-Dokumente

So zeigen Sie die Vorlagenbibliothek in Application Manager an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. In der Sektion Anwendungen wählen Sie die OptionCloudFormation-Stacks.

4. Wählen Sie Template-Bibliothek.

Erstellung von Vorlagen

Im folgenden Verfahren wird beschrieben, wie Sie eine CloudFormation-Vorlage in Application Manager erstellen. Wenn Sie eine Vorlage erstellen, geben Sie die Stackdetails der Vorlage entweder in JSON oder YAML ein. Wenn Sie noch keine Erfahrungen mit JSON oder YAML haben, können Sie AWS CloudFormationDesigner, ein Tool zum visuellen Erstellen und Ändern von Vorlagen, verwenden. Weitere Informationen finden Sie unter Was ist AWS CloudFormation-Designer? im AWS CloudFormation-Benutzerhandbuch. Weitere Informationen zur Struktur und Syntax einer Vorlage finden Sie unter Vorlagenanatomie.

Sie können eine Vorlage auch aus mehreren Vorlagenausschnitten erstellen. Vorlagenausschnitte sind Beispiele, die zeigen, wie Vorlagen für eine bestimmte Ressource geschrieben werden. Sie können z. B. Ausschnitte für Amazon Elastic Compute Cloud (Amazon EC2) -Instances, Amazon Simple Storage Service (Amazon S3) -Domänen, AWS CloudFormation-Mappings und mehr einsehen. Ausschnitte werden nach Ressourcen gruppiert. Sie finden AWS CloudFormation-Vorlagenausschnitte für allgemeine Zwecke in der Sektion Allgemeine Vorlagenausschnitte im AWS CloudFormation-Benutzerhandbuchaus.

Erstellen einer CloudFormation Vorlage in Application Manager (Konsole)

Führen Sie die folgenden Schritte aus, um eine CloudFormation-Vorlage in Application Manager mithilfe von AWS Management Console auszuführen.

Erstellen einer CloudFormation Vorlage in Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. In der Sektion Anwendungen wählen Sie die OptionCloudFormation-Stacks.
4. Klicken Sie auf Template-Bibliothek und wählen Sie dann entweder Vorlage erstellen oder wählen Sie eine vorhandene Vorlage aus und wählen Sie Aktionen, Klonen.
5. Geben Sie für Name einen Namen für die Vorlage ein, mit dem Sie die erstellten Ressourcen oder den Zweck des Stacks identifizieren können.
6. (Optional) Geben Sie für Versionsname einen Namen oder eine Nummer ein, um die Vorlagenversion zu identifizieren.

7. (Optional) Geben Sie unter Description (Beschreibung) Informationen zu dieser Vorlage ein.
8. In der Sektion Code-Editor wählen Sie entweder YAML oder JSONund geben den Vorlagencode ein oder kopieren ihn und fügen ihn ein.
9. (Optional) Wenden Sie im Abschnitt Tags ein oder mehrere Tag-Schlüssel-Name/Wert-Paare auf die Vorlage an.

   Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Weitere Informationen über das Taggen von System Manager-Ressourcen finden Sie unter Markieren von Systems Manager-Ressourcen.
10. (Optional) Geben Sie in der Sektion Berechtigungen eine AWS-Konto-ID ein und wählen Hinzufügen eines Kontos. Diese Aktion stellt die Leseberechtigung für die Vorlage bereit. Der Kontoinhaber kann die Vorlage bereitstellen und klonen, kann sie jedoch nicht bearbeiten oder löschen.
11. Wählen Sie Erstellen aus. Die Vorlage wird im Systems Manager (SSM) Document service gespeichert.

Erstellen einer CloudFormation Vorlage in Application Manager (Konsole)

Nachdem Sie den Inhalt Ihrer CloudFormation Vorlage in JSON oder YAML erstellt haben, können Sie den AWS Command Line Interface (AWS CLI) oder AWS Tools for PowerShell verwenden, um die Vorlage als SSM-Dokument zu speichern. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

Bevor Sie beginnen

Installieren und konfigurieren Sie die AWS CLI oder AWS Tools for PowerShell, falls noch nicht erfolgt. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for 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"
```

Bei erfolgreicher Ausführung gibt der Befehl eine Antwort zurück, die in etwa wie folgt aussieht:

```
{
    "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"
            }
```

```
        ]
    }
```

Bearbeiten einer Vorlage

Führen Sie die folgenden Schritte aus, um eine CloudFormation-Vorlage in Application Manager zu bearbeiten. Vorlagenänderungen sind in CloudFormation verfügbar, nachdem Sie ein Stack bereitgestellt haben, das die aktualisierte Vorlage verwendet.

Bearbeiten einer CloudFormation Vorlage in Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. In der Sektion Anwendungen wählen Sie die OptionCloudFormation-Stacks.
4. Wählen Sie Template-Bibliothek.
5. Wählen Sie eine Vorlage aus und wählen Sie dann Actions (Akitionen), Edit (Bearbeiten). Sie können den Namen einer Vorlage nicht ändern, aber Sie können alle anderen Details ändern.
6. Wählen Sie Save (Speichern). Die Vorlage wird im Systems Manager-Dokumentdienst gespeichert.

Arbeiten mit CloudFormation-Stacks

Application Manager, eine Funktion von AWS Systems Manager, unterstützt Sie bei der Bereitstellung und Verwaltung von Ressourcen für Ihre Anwendungen durch die Integration in AWS CloudFormation. Sie können CloudFormation-Vorlagen und -Stacks in Application Manager erstellen, bearbeiten und löschen. Bei einem Stack handelt es sich um eine Sammlung von AWS-Ressourcen, die Sie als einzelne Einheit verwalten können. Dies bedeutet, dass Sie eine Sammlung von AWS-Ressourcen mithilfe von CloudFormation Stacks erstellen, aktualisieren oder löschen können. Eine Vorlage ist eine formatierte Textdatei in JSON oder YAML, die die Ressourcen angibt, die Sie in Ihren Stacks bereitstellen möchten. Dieser Abschnitt enthält folgende Informationen.

Themen

- Erstellen eines Stacks
- Aktualisieren eines Stacks

Erstellen eines Stacks

In den folgenden Verfahren wird beschrieben, wie Sie ein CloudFormation-Stack mithilfe von Application Manager erstellen. Ein Stack basiert auf einer Vorlage. Wenn Sie einen Stack erstellen, können Sie entweder eine vorhandene Vorlage auswählen oder eine neue erstellen. Nachdem Sie den Stack erstellt haben, versucht das System sofort, die im Stack identifizierten Ressourcen zu erstellen. Nachdem das System die Ressourcen erfolgreich bereitgestellt hat, können die Vorlage und der Stack in Application Manager und CloudFormation eingesehen und bearbeitet werden.

> **Note**
>
> Es fallen keine Gebühren an beim Erstellen eines Stacks in Application Manager an, aber Sie zahlen für AWS-Ressourcen, die im Stack erstellt wurden.

Erstellen eines CloudFormation -Stacks mithilfe von Application Manager (Konsole)

Gehen Sie wie folgt vor, um einen neuen Stack mithilfe von Application Manager in der AWS Management Console.

Erstellen eines CloudFormation-Stacks

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. In der Sektion Anwendungen wählen Sie die OptionCloudFormation-Stacks.
4. In der Sektion Vorbereiten einer -Vorlage wählen Sie eine Option aus. Wenn Sie Vorhandene Vorlage verwenden wählen, können Sie zudem die Registerkarten in der Sektion Auswahl einer Vorlage verwenden, um die gewünschte Vorlage zu suchen. Wenn Sie eine der anderen Optionen auswählen, schließen Sie den Assistenten ab, um eine Vorlage vorzubereiten.
5. Überprüfen Sie auf der Seite Vorlagendetails angeben die Details der Vorlage, um sicherzustellen, dass der Prozess die gewünschten Ressourcen erstellt.

   - (Optional) Wenden Sie im Abschnitt Tags ein oder mehrere Tag-Schlüssel-Name/Wert-Paare auf die Vorlage an.
   - Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder

Umgebung. Weitere Informationen über das Taggen von System Manager-Ressourcen finden Sie unter Markieren von Systems Manager-Ressourcen.

- Wählen Sie Next (Weiter).

6. Geben Sie auf der Seite Stack-Details bearbeiten für Stack-Name einen Namen ein, der Ihnen hilft, die vom Stack erstellten Ressourcen oder seinen Zweck zu identifizieren.

   - Die Sektion Parameter enthält alle optionalen und erforderlichen Parameter, die in der Vorlage angegeben sind. Geben Sie in jedes Feld einen oder mehrere Parameter ein.
   - (Optional) Wenden Sie im Bereich Tags ein oder mehrere Tag-Schlüsselname/-wertpaare auf den Stack an.
   - (Optional) Klicken Sie im Bereich Berechtigungen eine AWS Identity and Access Management(IAM) -Rolle oder einen IAM Amazon-Ressourcennamen (ARN) ein. Das System verwendet die angegebene Dienstrolle, um alle in Ihrem Stack angegebenen Ressourcen zu erstellen. Wenn Sie keine IAM-Rolle angeben, verwendet AWS CloudFormation eine temporäre Sitzung, die das System anhand Ihrer Benutzeranmeldeinformationen erstellt. Weitere Informationen über diese IAM-Rolle finden Sie unter AWS CloudFormation-Servicerolle im AWS CloudFormation-Benutzerhandbuch.
   - Wählen Sie Next (Weiter).

7. Überprüfen Sie auf der Seite Überprüfung und Bereitstellung alle Details des Stacks. Wählen Sie eine Bearbeiten-Schaltfläche auf dieser Seite, um Änderungen vorzunehmen.
8. Wählen Sie Stack bereitstellen.

Application Manager zeigt die Seite CloudFormation-Stacks und den Status der Stackerstellung und -bereitstellung. Wenn CloudFormation den Stack nicht erstellen und bereitstellen kann, lesen Sie die folgenden Themen im AWS CloudFormation-Benutzerhandbuch.

- Stack-Statuscodes
- Fehlerbehebung für AWS CloudFormation

Nachdem Ihre Stack-Ressourcen bereitgestellt und ausgeführt wurden, können Benutzer Ressourcen direkt bearbeiten, indem sie den zugrunde liegenden Service verwenden, der die Ressource erstellt hat. Beispielsweise kann ein Benutzer mit der Amazon Elastic Compute Cloud (Amazon EC2) Konsole eine Server-Instance aktualisieren, die als Teil eines CloudFormation-Stacks erstellt wurde. Einige Änderungen können versehentlich oder absichtlich vorgenommen werden, um auf zeitkritische Betriebsereignisse zu reagieren. Unabhängig davon können Änderungen, die außerhalb von

CloudFormation vorgenommen werden, Stack-Aktualisierungen oder Löschvorgänge erschweren. Mit der Abweichungserkennung oder dem Abweichungsstatus können Sie Stack-Ressourcen identifizieren, an denen Konfigurationsänderungen außerhalb des CloudFormation-Managements vorgenommen wurden. Informationen zum Abweichungsstatus finden Sie unter Erkennen von nicht verwalteten Konfigurationsänderungen an Stacks und Ressourcen.

Erstellen eines CloudFormation -Stacks mithilfe von Application Manager (Befehlszeile)

Verwenden Sie die folgenden AWS Command Line Interface(AWS CLI) Verfahren, um einen Stack mithilfe einer CloudFormation Vorlage bereitzustellen, die als SSM-Dokument in Systems Manager gespeichert ist. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Weitere Informationen zu anderen AWS CLI-Verfahren zum Erstellen von Stacks finden Sie unter Erstellen eines Stacks im AWS CloudFormation-Benutzerhandbuch.

Bevor Sie beginnen

Installieren und konfigurieren Sie die AWS CLI oder AWS Tools for PowerShell, falls noch nicht erfolgt. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for 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" `
```

Aktualisieren eines Stacks

Sie können Updates auf einem CloudFormation-Stack bereitstellen, indem Sie den Stack direkt in Application Manager bearbeiten. Mit einer direkten Aktualisierung legen Sie Aktualisierungen für eine Vorlage oder Eingabeparameter fest. Nachdem Sie die Änderungen gespeichert und bereitgestellt haben, aktualisiert CloudFormation die AWS-Ressourcen entsprechend den von Ihnen angegebenen Änderungen.

Vor der Aktualisierung können Sie eine Vorschau der Änderungen anzeigen, die CloudFormation an Ihrem Stack mithilfe von Änderungssets vornehmen wird. Weitere Informationen finden Sie unter Aktualisieren von Stacks mithilfe von Änderungssets im AWS CloudFormation-Benutzerhandbuch.

So aktualisieren Sie einen CloudFormation-Stack in Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. In der Sektion Anwendungen wählen Sie die OptionCloudFormation-Stacks.
4. Wählen Sie einen Stack in der Liste aus und wählen Sie Aktionen, Stack aktualisieren.
5. Wählen Sie auf der Seite Vorlagenquelle angeben eine der folgenden Optionen aus und wählen Sie dann Next (Weiter).
   - Wählen Sie Aktuell im Stack bereitgestellten Vorlagencode verwenden, um eine Vorlage anzuzeigen. Wählen Sie in der Liste Versions (Versionen) eine Vorlagenversion aus und wählen Sie dann Next (Weiter) aus.
   - Wählen Sie Wechseln zu einer anderen Vorlage, um eine neue Vorlage für den Stack auszuwählen oder zu erstellen.
6. Wenn Sie die Änderungen an der Vorlage vorgenommen haben, wählen Sie Weiter aus.
7. Auf der Seite Stackdetails bearbeiten können Sie Parameter, Tags und Berechtigungen bearbeiten. Sie können den Namen eines Stacks nicht ändern. Nehmen Sie die gewünschten Änderungen vor und wählen Sie dann Weiter.
8. Überprüfen Sie auf der Seite Überprüfung und Bereitstellung alle Details des Stacks und wählen Sie dann Stack bereitstellen.

Arbeiten mit Clustern in Application Manager

Dieser Abschnitt enthält Themen, die Ihnen bei der Arbeit mit Container-Clustern von Amazon Elastic Container Service (Amazon ECS) und Amazon Elastic Kubernetes Service (Amazon EKS) in Application Manager helfen, eine Komponente vonAWS Systems Manager.

Inhalt

- Arbeiten mit Amazon ECS in Application Manager
- Arbeiten mit Amazon EKS in Application Manager
- Arbeiten mit Runbooks für Cluster

Arbeiten mit Amazon ECS in Application Manager

Mit Application Manager, eine Funktion von AWS Systems Manager, können Sie Ihre Amazon Elastic Container Service (Amazon ECS)-Cluster-Infrastruktur anzeigen und verwalten. Application Manager wendet ein Tag auf Ihren Amazon ECS-Cluster an, indem der Amazon-Ressourcenname (ARN) des Clusters als Tag-Wert verwendet wird. Application Manager bietet eine Komponentenlaufzeitansicht der Rechen-, Netzwerk- und Speicherressourcen in einem Cluster.

> **Note**
>
> Sie können keine Betriebsinformationen zu Ihren Containern in Application Manager verwalten oder anzeigen. Sie können nur Betriebsinformationen über die Infrastruktur verwalten und anzeigen, die Ihre Amazon-ECS-Ressourcen hostet.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie Verwalten von Clustern, um den Cluster in Amazon ECS zu öffnen.
- Wählen Sie Alle anzeigen, um eine Liste der Ressourcen in Ihrem Cluster anzuzeigen.
- Wählen Sie Anzeigen in CloudWatch, um Ressourcenalarme in Amazon anzuzeigen CloudWatch.
- Wählen Sie Manage nodes (Verwalten von Knoten) oder Manage Fargate profiles, (Fargate-Profile verwalten) um diese Ressourcen in Amazon ECS anzuzeigen.
- Wählen Sie eine Ressourcen-ID aus, um detaillierte Informationen darüber in der Konsole anzuzeigen, in der sie erstellt wurde.

- Anzeigen einer Liste der OpsItems im Zusammenhang mit Ihren Clustern.
- Zeigen Sie einen Verlauf von Runbooks an, die auf Ihren Clustern ausgeführt wurden.

So öffnen Sie den ECS-Cluster

1. Öffnen Sie die - AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie in der Sektion Container-Cluster ECS-Cluster.
4. Wählen Sie einen Cluster in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.

Arbeiten mit Amazon EKS in Application Manager

Application Manager, eine Funktion von AWS Systems Manager, ist in Amazon Elastic Kubernetes Service (Amazon EKS) integriert, um Informationen über den Zustand Ihrer Amazon-EKS-Cluster-Infrastruktur bereitzustellen. Application Manager wendet ein Tag auf Ihren Amazon-EKS-Cluster an, indem der Amazon-Ressourcenname (ARN) des Clusters als Tag-Wert verwendet wird. Application Manager bietet eine Komponentenlaufzeitansicht der Rechen-, Netzwerk- und Speicherressourcen in einem Cluster.

> **Note**
>
> Sie können keine Betriebsinformationen zu Ihren Amazon EKS-Pods oder -Containern in Application Manager verwalten. Sie können nur Betriebsinformationen zu der Infrastruktur verwalten und anzeigen, die Ihre Amazon EKS-Ressourcen hostet.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie Verwalten von Clustern, um den Cluster in Amazon EKS zu öffnen.
- Wählen Sie Alle anzeigen, um eine Liste der Ressourcen in Ihrem Cluster anzuzeigen.
- Wählen Sie Anzeigen in CloudWatch, um Ressourcenalarme in Amazon anzuzeigen CloudWatch.
- Wählen Sie Manage nodes (Verwalten von Knoten) oder Manage Fargate profiles (Fargate-Profile verwalten), um diese Ressourcen in Amazon EKS anzuzeigen.

- Wählen Sie eine Ressourcen-ID aus, um detaillierte Informationen darüber in der Konsole anzuzeigen, in der sie erstellt wurde.
- Anzeigen einer Liste der OpsItems im Zusammenhang mit Ihren Clustern.
- Zeigen Sie einen Verlauf von Runbooks an, die auf Ihren Clustern ausgeführt wurden.

So öffnen Sie eine EKS-Cluster-Anwendung

1. Öffnen Sie die - AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. In der Sektion Container Cluster wählen Sie EKS-Cluster.
4. Wählen Sie einen Cluster in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.

Arbeiten mit Runbooks für Cluster

Sie können Probleme mit AWS-Ressourcen von Application Manager, eine Funktion von AWS Systems Manager, mithilfe von Systems Manager Automation Runbooks beheben. Wenn Sie Runbook starten von einem Application Manager-Cluster wählen, zeigt das System eine gefilterte Liste von Runbooks basierend auf dem Typ der Ressourcen in Ihrem Cluster an. Wenn Sie das Runbook auswählen, das Sie starten möchten, öffnet Systems Manager die Seite Ausführen des Automatisierungsdokuments.

Bevor Sie beginnen

Bevor Sie ein Runbook von Application Managerstarten, gehen Sie wie folgt vor:

- Stellen Sie sicher, dass Sie über die richtigen Berechtigungen zum Starten von Runbooks verfügen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
- Lesen Sie die Dokumentation zur Automatisierungsprozedur zum Starten von Runbooks. Weitere Informationen finden Sie unter Ausführen von Automatisierungen.
- Wenn Sie Runbooks auf mehreren Ressourcen gleichzeitig starten möchten, lesen Sie die Dokumentation zur Verwendung von Zielen und Tarifkontrollen. Weitere Informationen finden Sie unter Ausführen von Automatisierungen im großen Maßstab.

So starten Sie ein Runbook für Cluster von Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. In der Sektion Container-Cluster wählen Sie einen Containertyp aus.
4. Wählen Sie den Cluster in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie auf der Registerkarte Runbooks die Option Start Runbook (Runbook starten) aus. Application Manager öffnet die Seite Execute automation document (Automatisierungsdokument ausführen) auf einer neuen Registerkarte. Weitere Informationen zu den Optionen auf der Seite Ausführen des Automationsdokuments finden Sie unter Ausführen von Automatisierungen.

AWS AppConfig

AWS AppConfig Feature-Flags und dynamische Konfigurationen helfen Softwareentwicklern dabei, das Anwendungsverhalten in Produktionsumgebungen ohne vollständige Codebereitstellungen schnell und sicher anzupassen. AWS AppConfig beschleunigt die Häufigkeit von Softwareveröffentlichungen, verbessert die Ausfallsicherheit von Anwendungen und hilft Ihnen, neu auftretende Probleme schneller zu lösen. Mithilfe von Feature-Flags können Sie schrittweise neue Funktionen für Benutzer bereitstellen und die Auswirkungen dieser Änderungen messen, bevor Sie die neuen Funktionen vollständig für alle Benutzer bereitstellen. Mithilfe von Betriebsflags und dynamischen Konfigurationen können Sie Sperrlisten und Zulassungslisten aktualisieren, Grenzwerte einschränken, den Umfang der Protokollierung einschränken und andere betriebliche Optimierungen vornehmen, um schnell auf Probleme in Produktionsumgebungen zu reagieren.

Weitere Informationen finden Sie unter Was ist? AWS AppConfig im AWS AppConfig Benutzerhandbuch.

AWS Systems Manager Parameter Store

Parameter Store, eine Funktion von AWS Systems Manager, bietet sicheren, hierarchischen Speicher für die Verwaltung von Konfigurationsdaten und Geheimnissen. Sie können Daten wie Passwörter, Datenbankzeichenfolgen, Amazon Machine Image (AMI) IDs und Lizenzcodes als Parameterwerte speichern. Sie können Werte als Klartext oder als verschlüsselte Daten speichern. Sie können Systems Manager-Parameter in Skripts, Befehlen, SSM-Dokumenten und Konfigurations- und Automatisierungs-Workflows referenzieren, indem Sie den eindeutigen Namen verwenden, den Sie

beim Erstellen des Parameters angegeben haben. Um mit Parameter Store zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Parameter Store aus.

Parameter Store ist auch in Secrets Manager integriert. Sie können Secrets-Manager-Geheimnisse abrufen, wenn Sie andere AWS-Services verwenden, die bereits Referenzen zu Parameter Store-Parametern unterstützen. Weitere Informationen finden Sie unter Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter.

> **Note**
>
> Um Lebenszyklen für die Passwortrotation zu implementieren, verwenden Sie. AWS Secrets Manager Sie können Datenbankanmeldeinformationen, API-Schlüssel und andere geheime Informationen mit Secrets Manager während ihres gesamten Lebenszyklus mühelos rotieren, verwalten und abfragen. Weitere Informationen finden Sie unter Was ist? AWS Secrets Manager im AWS Secrets Manager Benutzerhandbuch.

Welche Vorteile bietet Parameter Store meiner Organisation?

Parameter Store bietet die folgenden Vorteile:

- Verwenden Sie einen sicheren, skalierbaren, gehosteten Verschlüsselungsmanagementservice ohne zu verwaltende Server.
- Verbessern Sie Ihre Sicherheit, indem Sie Ihre Daten von Ihrem Code trennen.
- Speichern Sie Konfigurationsdaten und verschlüsselte Zeichenfolgen in Hierarchien und verfolgen Sie Versionen nach.
- Steuern und prüfen Sie Zugriff genau.
- Speichern Sie Parameter zuverlässig, da Parameter Store in mehreren Availability Zones in einer AWS-Region gehostet wird.

An wen richtet sich Parameter Store?

- Jeder AWS Kunde, der eine zentrale Möglichkeit zur Verwaltung von Konfigurationsdaten haben möchte.
- Softwareentwickler, die verschiedene Logins und Referenzströme speichern möchten.

- Administratoren, die Benachrichtigungen erhalten möchten, wenn ihre Secrets und Passwörter geändert werden oder nicht.

Über welche Features verfügt Parameter Store?

- Änderungsbenachrichtigung

  Sie können Änderungsbenachrichtigungen konfigurieren und automatisierte Aktionen für beide Parameter und Parameterrichtlinien auslösen. Weitere Informationen finden Sie unter Einrichten von Benachrichtigungen oder Auslöseraktionen basierend auf Parameter Store-Ereignissen.

- Organisieren von Parametern

  Sie können Ihre Parameter individuell markieren, um anhand der Tags, die Sie ihnen zugewiesen haben, einen oder mehrere Parameter zu identifizieren. Sie können Parameter z. B. nach bestimmten Umgebungen oder Abteilungen taggen. Weitere Informationen finden Sie unter Markieren von Systems Manager-Parametern.

- Beschriftungsversionen

  Sie können einen Alias für Versionen Ihres Parameters zuordnen, indem Sie Beschriftungen erstellen. Dank Beschriftungen können Sie sich den Zweck einer Parameterversion merken, wenn mehrere Versionen vorhanden sind.

- Datenvalidierung

  Sie können Parameter erstellen, die auf eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance verweisen, und Parameter Store überprüft diese Parameter, um sicherzustellen, dass es auf den erwarteten Ressourcentyp verweist, dass die Ressource vorhanden ist und dass der Kunde die Berechtigung hat, die Ressource zu verwenden. Sie können beispielsweise einen Parameter mit Amazon Machine Image (AMI) ID als einen Wert mit `aws:ec2:image`-Datentyp erstellen und Parameter Store führt einen asynchronen Validierungsvorgang aus, um sicherzustellen, dass der Parameterwert die Formatierungsanforderungen für eine AMI ID erfüllt und dass das angegebene AMI in Ihrem AWS-Konto verfügbar ist.

- Referenz-Secrets

  Parameter Storeist integriert, AWS Secrets Manager sodass Sie Secrets Manager abrufen können, wenn Sie andere verwenden AWS-Services , die bereits Verweise auf Parameter Store Parameter unterstützen.

- Parameter mit anderen Konten teilen

Sie können die Konfigurationsdaten optional in einer einzigen Datei zentralisieren AWS-Konto und Parameter mit anderen Konten teilen, die darauf zugreifen müssen.

- Von anderen aus zugänglich AWS-Services

  Sie können Parameter Store-Parameter mit anderen Systems-Manager-Funktionen und AWS-Services zum Abrufen von Geheimnissen und Konfigurationsdaten aus einem zentralen Speicher verwenden. Parameter funktionieren mit Systems Manager Manager-Funktionen wie Run Command Automatisierung und State Manager Funktionen von AWS Systems Manager. Sie können auch in einer Reihe anderer Parameter auf Parameter verweisen AWS-Services, z. B. in den folgenden:

  - 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
- Integrieren Sie mit anderen AWS-Services

  Konfigurieren Sie die Integration mit den folgenden Optionen AWS-Services für Verschlüsselung, Benachrichtigung, Überwachung und Prüfung:

  - AWS Key Management Service (AWS KMS)
  - Amazon-Simple-Notification-Service (Amazon-SNS)
  - Amazon CloudWatch: Weitere Informationen finden Sie unterKonfigurieren von EventBridge Regeln für Parameter und Parameterrichtlinien.
  - Amazon EventBridge: Weitere Informationen finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen undReferenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager.
  - AWS CloudTrail: Weitere Informationen finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.

Was ist ein Parameter?

Ein Parameter Store-Parameter ist eine beliebige Datenmenge, die in Parameter Store gespeichert wird, z. B. ein Textblock, eine Namensliste, ein Passwort, eine AMI-ID, ein Lizenzschlüssel usw. Sie können diese Daten zentral und sicher in Ihren Skripten, Befehlen und SSM-Dokumenten referenzieren.

Wenn Sie auf einen Parameter verweisen, geben Sie den Parameternamen unter Verwendung der folgenden Konvention an:

{{ssm:*parameter-name*}}

> **Note**
>
> Parameter können nicht referenziert oder in den Werten anderer Parameter verschachtelt werden. Sie können {{}} oder {{ssm:*parameter-name*}} nicht in einen Parameterwert aufnehmen.

Parameter Store bietet Unterstützung für drei Arten von Parametern: `String`, `StringList` und `SecureString`.

Mit einer Ausnahme geben Sie beim Erstellen oder Aktualisieren eines Parameters den Parameterwert als Klartext ein und Parameter Store führt keine Validierung für den eingegebenen Text aus. Bei `String`-Parametern können Sie jedoch den Datentyp als `aws:ec2:image` angeben und Parameter Store prüft , ob der eingegebene Wert das richtige Format für ein Amazon EC2 AMI aufweist. Beispiel: `ami-12345abcdeEXAMPLE`.

Parametertyp: String

Standardmäßig bestehen `String`-Parameter aus einem beliebigen Textblock, den Sie eingeben. Beispielsweise:

- `abc123`
- `Example Corp`
- `<img src="images/bannerImage1.png"/>`

Typ des Parameters: StringList

`StringList`-Parameter enthalten eine durch Komma getrennte Liste von Werten wie in den folgenden Beispielen gezeigt.

```
Monday,Wednesday,Friday
```

```
CSV,TSV,CLF,ELF,JSON
```

Parametertyp: SecureString

Ein `SecureString`-Parameter kann aus beliebigen vertraulichen Daten bestehen, die auf sichere Weise gespeichert und referenziert werden müssen. Wenn Sie Daten haben, die Benutzer nicht ändern oder als Klartext referenzieren sollen (z. B. Passwörter oder Lizenzschlüssel), erstellen Sie diese Parameter mit dem `SecureString`-Datentyp.

> **Important**
>
> Speichern Sie keine vertraulichen Daten in einem `String`- oder `StringList`-Parameter. Verwenden Sie für alle vertraulichen Daten, die verschlüsselt bleiben müssen, nur den `SecureString`-Parametertyp.
> Weitere Informationen finden Sie unter Erstellen eines SecureString-Parameters (AWS CLI).

Wir empfehlen die Verwendung von `SecureString`-Parametern in den folgenden Szenarien:

- Sie möchten Daten/Parameter überall verwenden, AWS-Services ohne die Werte als Klartext in Befehlen, Funktionen, Agentenprotokollen oder Protokollen verfügbar zu machen. CloudTrail
- Sie möchten steuern, welche Personen auf vertrauliche Daten zugreifen können.
- Sie möchten in der Lage sein, zu überprüfen, wann auf sensible Daten zugegriffen wird (). CloudTrail
- Sie möchten Ihre sensiblen Daten verschlüsseln und Sie möchten Ihre eigenen Verschlüsselungsschlüssel für die Zugriffsverwaltung verwenden.

> **Important**
>
> Nur der Wert eines `SecureString`-Parameters wird verschlüsselt. Der Name des Parameters, die Beschreibung und andere Eigenschaften sind nicht verschlüsselt.

Sie können den `SecureString` Parametertyp für Textdaten verwenden, die Sie verschlüsseln möchten, z. B. Kennwörter, Anwendungsgeheimnisse, vertrauliche Konfigurationsdaten oder andere Arten von Daten, die Sie schützen möchten. `SecureString`Daten werden mit einem Schlüssel ver- und entschlüsselt. AWS KMS Sie können entweder einen Standard-KMS-Schlüssel verwenden, der von bereitgestellt wird, AWS oder Sie können Ihren eigenen AWS KMS key erstellen und verwenden. (Verwenden Sie Ihre eigenen AWS KMS key , wenn Sie den Benutzerzugriff auf `SecureString`-Parameter einschränken möchten. Weitere Informationen finden Sie unter IAM-Berechtigungen für die Verwendung von AWS Standardschlüsseln und vom Kunden verwalteten Schlüsseln.)

Sie können `SecureString` Parameter auch zusammen mit anderen verwenden AWS-Services. Im folgenden Beispiel ruft die Lambda-Funktion mithilfe der GetParametersAPI einen `SecureString` Parameter ab.

```
from __future__ import print_function

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
```

AWS KMS Verschlüsselung und Preisgestaltung

Wenn Sie bei der Erstellung Ihres `SecureString` Parameters den Parametertyp wählen, verschlüsselt Systems AWS KMS Manager den Parameterwert.

> **Important**
>
> Parameter Store unterstützt nur KMS-Schlüssel zur symmetrischen Verschlüsselung. Sie können keinen KMS-Schlüssel zur asymmetrischen Verschlüsselung verwenden, um Ihre Parameter zu verschlüsseln. Wie Sie feststellen, ob ein KMS-Schlüssel symmetrisch oder

asymmetrisch ist, erfahren Sie unter Erkennen symmetrischer und asymmetrischer Schlüssel im AWS Key Management Service -Entwicklerhandbuch.

Für die Erstellung eines `SecureString` Parameters fallen keine Gebühren Parameter Store an, es fallen jedoch Gebühren für die Verwendung der AWS KMS Verschlüsselung an. Weitere Informationen finden Sie unter AWS Key Management Service -Preise.

Weitere Informationen zu Von AWS verwaltete Schlüssel und vom Kunden verwalteten Schlüsseln finden Sie unter AWS Key Management Service Konzepte im AWS Key Management Service Entwicklerhandbuch. Weitere Informationen zu AWS KMS Verschlüsselung Parameter Store und Verschlüsselung finden Sie unter AWS Systems ManagerParameter StoreAnwendungsmöglichkeiten AWS KMS.

Note

Verwenden Sie die AWS KMS `DescribeKey` Operation Von AWS verwalteter Schlüssel, um eine anzuzeigen. Dieses AWS Command Line Interface (AWS CLI) Beispiel dient `DescribeKey` zum Anzeigen eines Von AWS verwalteter Schlüssel.

```
aws kms describe-key --key-id alias/aws/ssm
```

Weitere Informationen

- Erstellen eines SecureString-Parameters und Verknüpfen eines Knotens mit einer Domain (PowerShell)
- Wird verwendetParameter Store, um sicher auf Geheimnisse zuzugreifen und Config zu konfigurieren in CodeDeploy
- Interessante Artikel zu Amazon EC2 Systems Manager Parameter Store

Einrichten von Parameter Store

Um Parameter in Parameter Store, einer Funktion von AWS Systems Manager, einrichten zu können, müssen Sie zunächst AWS Identity and Access Management (IAM)-Richtlinien konfigurieren, die den Benutzern in Ihrem Konto die Berechtigung zur Ausführung der von Ihnen festgelegten Aktionen erteilen. In diesem Abschnitt finden Sie Informationen darüber, wie Sie diese

Richtlinien mithilfe der IAM-Konsole manuell konfigurieren und sie Benutzern und Benutzergruppen zuweisen. Darüber hinaus können Sie Richtlinien erstellen und zuordnen, um zu steuern, welche Parameteraktionen auf einem verwalteten Knoten ausgeführt werden dürfen. Außerdem enthält dieser Abschnitt Informationen dazu, wie Sie Amazon EventBridge-Regeln erstellen, anhand derer Sie Benachrichtigungen über Änderungen an Systems Manager-Parametern erhalten. Anhand der EventBridge-Regeln können Sie zudem andere Aktionen in AWS aufrufen, die auf Änderungen in Parameter Store basieren.

Inhalt

- Einschränken des Zugriffs auf Systems Manager-Parameter mithilfe von IAM-Richtlinien
- Verwalten von Parameterstufen
- Durchsatz erhöhen oder zurücksetzen Parameter Store
- Einrichten von Benachrichtigungen oder Auslöseraktionen basierend auf Parameter Store-Ereignissen

Einschränken des Zugriffs auf Systems Manager-Parameter mithilfe von IAM-Richtlinien

Sie schränken den Zugriff auf AWS Systems Manager Parameter mithilfe von AWS Identity and Access Management (IAM) ein. Genauer gesagt können Sie IAM-Richtlinien erstellen, die den Zugriff auf die folgenden API-Operationen beschränken:

- DeleteParameter
- DeleteParameters
- DescribeParameters
- GetParameter
- GetParameters
- GetParameterHistory
- GetParametersByPath
- PutParameter

Wenn Sie IAM-Richtlinien verwenden, um den Zugriff auf Systems Manager-Parameter einzuschränken, sollten Sie restriktive IAM-Richtlinien erstellen und verwenden. Die folgende Richtlinie ermöglicht z. B. den Aufruf der API-Operationen `DescribeParameters` und

`GetParameters` für einen eingeschränkten Satz von Ressourcen. Das bedeutet, dass der Benutzer Informationen zu allen Parametern, die mit `prod-*` beginnen, abrufen und diese verwenden kann.

```
{
    "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

Wenn ein Benutzer Zugriff auf einen Pfad hat, kann er auf alle Ebenen dieses Pfads zugreifen. Wenn ein Benutzer beispielsweise die Berechtigung für den Zugriff auf den Pfad `/a` besitzt, dann kann er auch auf `/a/b` zugreifen. Selbst wenn einem Benutzer in IAM der Zugriff auf den Parameter `/a/b` ausdrücklich verweigert wurde, kann er dennoch die `GetParametersByPath`-API-Operation rekursiv für `/a` aufrufen und `/a/b` anzeigen.

Vertrauenswürdigen Administratoren können Sie mithilfe einer Richtlinie ähnlich dem folgenden Beispiel Zugriff auf alle API-Operationen für Systems Manager-Parameter gewähren. Mit dieser Richtlinie erhält der Benutzer vollen Zugriff auf alle Produktionsparameter, die mit `dbserver-prod-*` beginnen.

```
{
    "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-2:123456789012:parameter/dbserver-prod-*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:DescribeParameters",
            "Resource": "*"
        }
    ]
}
```

Berechtigungen verweigern

Jede API ist einzigartig und verfügt über unterschiedliche Operationen und Berechtigungen, die Sie einzeln zulassen oder verweigern können. Eine explizite Zugriffsverweigerung überschreibt jede Zugriffserlaubnis in einer Richtlinie.

> **Note**
>
> Der Standardschlüssel AWS Key Management Service (AWS KMS) verfügt über `Decrypt` Berechtigungen für alle IAM-Prinzipale innerhalb von. AWS-Konto Wenn Sie unterschiedliche Zugriffsebenen für `SecureString`-Parameter in Ihrem Konto haben möchten, raten wir Ihnen davon ab, den Standardschlüssel zu verwenden.

Wenn Sie möchten, dass alle API-Operationen, die Parameterwerte abrufen, das gleiche Verhalten haben, dann können Sie ein Muster wie `GetParameter*` in einer Richtlinie verwenden. Im folgenden Beispiel wird gezeigt, wie Sie `GetParameter`, `GetParameters`, `GetParameterHistory` und `GetParametersByPath` für alle Parameter, die mit `prod-*` beginnen, verweigern.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-east-2:123456789012:parameter/prod-*"
        }
    ]
}
```

Das folgende Beispiel zeigt, wie einige Befehle verweigert werden, während der Benutzer andere Befehle für alle Parameter ausführen kann, die mit `prod-*` beginnen.

```
{
    "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-2:123456789012:parameter/prod-*"
        }
    ]
}
```

> **Note**
>
> Der Parameterverlauf umfasst alle Parameterversionen, einschließlich der aktuellen. Wenn einem Benutzer daher die Berechtigung für `GetParameter`, `GetParameters` und `GetParameterByPath` nicht gewährt wird, er aber die Berechtigung für `GetParameterHistory` erhält, kann er den aktuellen Parameter, einschließlich `SecureString`, unter Verwnedung von `GetParameterHistory` sehen.

Erlauben nur bestimmter Parameter für die Ausführung auf Knoten

Sie können den Zugriff so steuern, dass verwaltete Knoten nur von Ihnen angegebene Parameter ausführen können.

Wenn Sie bei der Erstellung Ihres `SecureString` Parameters den Parametertyp wählen, verschlüsselt Systems AWS KMS Manager den Parameterwert. AWS KMS verschlüsselt den Wert entweder mithilfe eines Von AWS verwalteter Schlüssel oder eines vom Kunden verwalteten Schlüssels. Weitere Informationen zu AWS KMS und AWS KMS key finden Sie im AWS Key Management Service Entwicklerhandbuch.

Sie können das anzeigen, Von AWS verwalteter Schlüssel indem Sie den folgenden Befehl von der aus ausführen AWS CLI.

```
aws kms describe-key --key-id alias/aws/ssm
```

Im folgenden Beispiel dürfen Knoten einen Parameterwert nur für Parameter abrufen, die mit `prod-` beginnen. Wenn der Parameter ein `SecureString`-Parameter ist, entschlüsselt der Knoten die Zeichenfolge mit AWS KMS.

> **Note**
>
> Instance-Richtlinien wie im folgenden Beispiel werden der Instance-Rolle in IAM zugeordnet. Weitere Informationen zur Konfiguration des Zugriffs auf Systems Manager-Funktionen einschließlich einer Anleitung für die Zuweisung von Richtlinien für Benutzer und Instances finden Sie unter Systems Manager mit EC2-Instances verwenden.

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-2:123456789012:parameter/prod-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-2:123456789012:key/4914ec06-e888-4ea5-
a371-5b88eEXAMPLE"
            ]
        }
    ]
}
```

IAM-Berechtigungen für die Verwendung von AWS Standardschlüsseln und vom Kunden verwalteten Schlüsseln

Parameter Store`SecureString`Parameter werden mithilfe von Schlüsseln ver- und entschlüsselt AWS KMS . Sie können wählen, ob Sie Ihre `SecureString` Parameter entweder mit einem AWS KMS key oder mit dem Standard-KMS-Schlüssel von verschlüsseln möchten. AWS

Wenn Sie einen kundenverwalteten Schlüssel verwenden, muss die IAM-Richtlinie, die einem Benutzer Zugriff auf einen Parameter oder Parameterpfad erteilt, explizite `kms:Encrypt`-Berechtigungen für den Schlüssel bereitstellen. Die folgende Richtlinie ermöglicht es einem Benutzer beispielsweise, `SecureString` Parameter zu erstellen, zu aktualisieren und anzuzeigen, die mit `prod-` dem angegebenen AWS-Region und AWS-Konto beginnen.

```
{
    "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"
  1
            ],
            "Resource": [
                "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-12345EXAMPLE"
            ]
        }
    ]
}
```

[1]Um verschlüsselte erweiterte Parameter mit dem angegebenen kundenverwalteten Schlüssel zu erstellen, ist die Berechtigung `kms:GenerateDataKey` erforderlich.

Im Gegensatz hierzu haben alle Benutzer innerhalb des Kundenkontos Zugriff auf den standardmäßigen AWS -verwalteten Schlüssel. Wenn Sie diesen Standardschlüssel zum Verschlüsseln von `SecureString`-Parametern verwenden und nicht möchten, dass Benutzer mit `SecureString`-Parametern arbeiten, müssen ihre IAM-Richtlinien den Zugriff auf den Standardschlüssel ausdrücklich ablehnen, wie im folgenden Richtlinienbeispiel gezeigt.

> **Note**
>
> Sie finden den Amazon-Ressourcennamen (ARN) des Standardschlüssels in der AWS KMS -Konsole auf der Seite AWS -verwaltete Schlüssel. Der Standardschlüssel ist der Schlüssel, der mit `aws/ssm` in der Spalte Alias (Alias) identifiziert wird.

```
{
```

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

Wenn Sie in Bezug auf die `SecureString`-Parameter in Ihrem Konto eine granulare Zugriffskontrolle benötigen, sollten Sie einen kundenverwalteten Schlüssel verwenden, um den Zugriff auf diese Parameter zu schützen und einzuschränken. Wir empfehlen außerdem, die Verwendung AWS CloudTrail zur Überwachung von `SecureString` Parameteraktivitäten zu verwenden.

Weitere Informationen finden Sie unter den folgenden Themen:

- Auswertungslogik für Richtlinien im IAM-Benutzerhandbuch
- Verwenden von Schlüsselrichtlinien in AWS KMS im AWS Key Management Service - Benutzerhandbuch
- Ereignisse mit dem CloudTrail Ereignisverlauf im AWS CloudTrail Benutzerhandbuch anzeigen

Verwalten von Parameterstufen

Parameter Store, eine Fähigkeit von AWS Systems Manager, umfasst Standardparameter und erweiterte Parameter. Parameter werden einzeln konfiguriert, sodass sie entweder die Standardparameterstufe (Standardstufe) oder die erweiterte Parameterstufe verwenden.

Sie können einen Standardparameter jederzeit in einen erweiterten Parameter ändern. Sie können jedoch einen erweiterten Parameter nicht auf einen Standardparameter zurücksetzen. Das Zurücksetzen eines erweiterten Parameters auf einen Standardparameter würde zu Datenverlust führen, weil das System die Größe des Parameters von 8 KB auf 4 KB kürzt. Durch das Zurücksetzen

würden auch etwaige dem Parameter angefügte Richtlinien entfernt. Erweiterte Parameter verwenden eine andere Form der Verschlüsselung als Standardparameter. Weitere Informationen finden Sie unter Arbeiten von AWS Systems ManagerParameter Store mit AWS KMS im AWS Key Management Service -Entwicklerhandbuch.

Wenn Sie einen erweiterten Parameter nicht mehr benötigen oder verhindern wollen, dass weitere Gebühren dafür anfallen, löschen Sie ihn und erstellen Sie ihn als Standardparameter neu.

Die folgende Tabelle beschreibt die Unterschiede zwischen den Stufen.

	Standard	Advanced
Gesamtzahl der zulässigen Parameter (pro AWS-Konto und AWS-Region)	10.000	100 000
Maximale Größe eines Parameterwerts.	4 KB	8 KB
Parameterrichtlinien verfügbar	Nein	Ja Weitere Informationen finden Sie unter Zuweisen von Parameterrichtlinien.
Kosten	Keine zusätzlichen Gebühren	Gebührenpflichtig Weitere Informationen finden Sie unter AWS Systems Manager Preise für Parameter Store.

Themen

- Angeben einer Standardparameterstufe
- Ändern eines Standardparameters in einen fortgeschrittenen Parameter

Angeben einer Standardparameterstufe

In Anforderungen zum Erstellen oder Aktualisieren eines Parameters (d. h. der Operation PutParameter) können Sie die Parameterstufe angeben, die in der Anforderung verwendet werden soll. Im Folgenden finden Sie ein Beispiel unter Verwendung der 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"
```

Wenn Sie eine Stufe in der Anforderung angeben, erstellt oder aktualisiert Parameter Store den Parameter entsprechend Ihrer Anforderung. Wenn Sie jedoch nicht explizit eine Stufe in einer Anforderung angeben, bestimmt die Parameter Store-Standardstufeneinstellung, in welcher Stufe der Parameter erstellt wird.

Die Standardstufe, wenn Sie beginnen Parameter Store zu verwenden, ist die Standardparameterstufe. Wenn Sie die erweiterte Parameterstufe verwenden, können Sie einen der folgenden als Standardwert angeben:

- Erweitert: Mit dieser Option wertet Parameter Store alle Anforderungen als erweiterte Parameter aus.
- Intelligent-Tiering: Mit dieser Option wertet Parameter Store jede Anforderung aus, um zu ermitteln, ob es sich um einen Standard- oder erweiterten Parameter handelt.

  Wenn die Anforderung keine Optionen enthält, die einen erweiterten Parameter erfordern, wird der Parameter in der Standardparameterstufe erstellt. Wenn eine oder mehrere Optionen, die einen

erweiterten Parameter erfordern, in der Anforderung enthalten sind, erstellt Parameter Store einen Parameter in der erweiterten Parameterstufe.

Vorteile von Intelligent-Tiering

Nachstehend sind Gründe, warum Sie Intelligent-Tiering als Standardstufe auswählen können.

Kostenkontrolle – Intelligent-Tiering hilft Ihnen, Ihre parameterbezogenen Kosten zu kontrollieren, indem immer Standardparameter erstellt werden, außer ein erweiterter Parameter ist absolut notwendig.

Automatisches Upgrade auf die erweiterte Parameterstufe – Wenn Sie eine Änderung an Ihrem Code vornehmen, die ein Upgrade eines Standardparameters auf einen erweiterten Parameter erfordert, übernimmt Intelligent-Tiering die Konvertierung für Sie. Sie müssen Ihren Code nicht ändern, um das Upgrade abzuwickeln.

Hier finden Sie einige Beispiele für automatische Upgrades:

- Ihre AWS CloudFormation Vorlagen stellen zahlreiche Parameter bereit, wenn sie ausgeführt werden. Wenn Sie durch diesen Prozess das Kontingent von 10.000 Parametern in der Stufe mit den Standardparametern erreichen, führt Intelligent-Tiering automatisch ein Upgrade auf die Stufe mit erweiterten Parametern durch, sodass Ihre Prozesse nicht unterbrochen werden. AWS CloudFormation
- Sie speichern einen Zertifikatwert in einem Parameter, drehen den Zertifikatswert regelmäßig und der Inhalt liegt unter dem Limit von 4 KB des Standard-Parameter-Kontingents. Wenn ein Ersatzzertifikatwert 4 KB überschreitet, aktualisiert Intelligent-Tiering den Parameter automatisch auf die erweiterte Parameterstufe.
- Sie möchten einer Parameterrichtlinie zahlreiche vorhandene Standardparameter zuordnen, die die erweiterte Parameterstufe erfordert. Anstatt die Option `--tier Advanced` in allen Aufrufen inkludieren zu müssen, um die Parameter zu aktualisieren, aktualisiert Intelligent-Tiering die Parameter automatisch auf die erweiterte Parameterstufe. Mit der Option „Intelligent-Tiering“ werden Parameter immer dann von „Standard“ auf „erweitert“ aktualisiert, wenn Kriterien für die erweiterte Parameterstufe eingeführt werden.

Optionen, für die ein erweiterter Parameter erforderlich ist, umfassen die folgenden:

- Die Inhaltsgröße des Parameters beträgt mehr als 4 KB.
- Der Parameter verwendet eine Parameterrichtlinie.

- Derzeit sind in Ihrem System bereits mehr als 10.000 Parameter vorhanden. AWS-Konto AWS-Region

Optionen für die Standardstufe

Die Stufenoptionen, die Sie als Standard festlegen können, umfassen die folgenden.

- Standard – Der Die Standardparameterstufe, wenn Sie beginnen Parameter Store zu verwenden. Mithilfe der Ebene mit den Standardparametern können Sie 10.000 Parameter für jeden Parameter AWS-Region in einem erstellen. AWS-Konto Die Inhaltsgröße jedes Parameters darf maximal 4 KB betragen. Standardparameter unterstützen keine Parameterrichtlinien. Für die Nutzung der Standardparameterstufe fallen keine zusätzlichen Gebühren an. Die Auswahl von Standard als Standardstufe bedeutet, dass Parameter Store immer versucht, einen Standardparameter für Anforderungen zu erstellen, die keine Stufe angeben.
- Erweitert — Verwenden Sie die Stufe mit erweiterten Parametern, um maximal 100.000 Parameter für jeden Parameter in einem zu erstellen. AWS-Region AWS-Konto Die Inhaltsgröße jedes Parameters darf maximal 8 KB betragen. Erweiterte Parameter unterstützen Parameterrichtlinien. Für die Nutzung der erweiterten Parameterstufe fallen Gebühren an. Weitere Informationen finden Sie unter AWS Systems Manager Preise für. Parameter Store Die Auswahl von Advanced (Erweitert) als Standardstufe bedeutet, dass Parameter Store immer versucht, einen erweiterten Parameter für Anforderungen zu erstellen, die keine Stufe angeben.

  > **Note**
  >
  > Wenn Sie die erweiterte Parameterstufe auswählen, müssen Sie AWS explizit autorisieren, Ihrem Konto für alle erweiterten Parameter, die Sie erstellen, Gebühren zu verrechnen.

- Intelligent-Tiering – Mit der Option Intelligent-Tiering bestimmt Parameter Store, ob die Standardparameterstufe oder die erweiterte Parameterstufe basierend auf dem Inhalt der Anforderung verwendet werden soll. Wenn Sie beispielsweise einen Befehl ausführen, um einen Parameter mit einem Inhalt unter 4 KB zu erstellen, und der aktuelle Wert AWS-Region in Ihrem AWS-Konto weniger als 10.000 Parameter enthält und Sie keine Parameterrichtlinie angeben, wird ein Standardparameter erstellt. Wenn Sie einen Befehl ausführen, um einen Parameter mit mehr als 4 KB Inhalt zu erstellen, haben Sie bereits mehr als 10.000 Parameter AWS-Region in Ihrem System AWS-Konto, oder wenn Sie eine Parameterrichtlinie angeben, wird ein erweiterter Parameter erstellt.

> **Note**
>
> Wenn Sie Intelligent-Tiering wählen, autorisieren Sie ausdrücklich, Ihr Konto mit allen von Ihnen erstellten erweiterten Parametern AWS zu belasten.

Sie können die Parameter Store-Standardstufeneinstellung jederzeit ändern.

Konfigurieren von Berechtigungen zum Angeben einer Parameter Store-Standardstufe

Stellen Sie sicher, dass Sie in AWS Identity and Access Management (IAM) berechtigt sind, die Standardparameterstufe zu ändern, Parameter Store indem Sie einen der folgenden Schritte ausführen:

- Stellen Sie sicher, dass Sie die `AdministratorAccess`-Richtlinie an Ihre IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) anfügen.
- Stellen Sie sicher, dass Sie über die Berechtigung zum Ändern der Standardstufeneinstellung verfügen, indem Sie die folgenden API-Operationen verwenden:
  - GetServiceSetting
  - UpdateServiceSetting
  - ResetServiceSetting

Gewähren Sie der IAM-Entität die folgenden Berechtigungen, damit ein Benutzer die Standard-Kontingent-Einstellung für Parameter in einer bestimmten AWS-Region in einem AWS-Konto anzeigen und ändern kann.

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

```
            "Action": [
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:region:account-id:servicesetting/ssm/parameter-
store/default-parameter-tier"
        }
    ]
}
```

Administratoren können schreibgeschützte Berechtigungen festlegen, indem sie die folgenden Berechtigungen zuweisen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "*"
        }
    ]
}
```

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter Erstellen einer Rolle für einen externen Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch.

- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter Erstellen einer Rolle für einen IAM-Benutzer im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Angeben oder Ändern der Parameter Store-Standardstufe (Konsole)

Das folgende Verfahren zeigt, wie Sie die Systems Manager Manager-Konsole verwenden, um die Standardparameterebene für das aktuelle AWS-Konto und anzugeben oder zu ändern AWS-Region.

> **Tip**
>
> Wenn Sie noch keinen Parameter erstellt haben, können Sie das AWS Command Line Interface (AWS CLI) oder verwenden, AWS Tools for Windows PowerShell um die Standardparameterebene zu ändern. Weitere Informationen finden Sie unter Angeben oder Ändern der Parameter Store-Standardstufe (AWS CLI) und Angeben oder Ändern der Parameter Store Standardstufe (PowerShell).

So legen Sie die Parameter Store-Standardstufe fest oder ändern sie

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie die Registerkarte Settings.
4. Klicken Sie auf Ändern der Standardstufe.
5. Wählen Sie eine der folgenden Optionen aus.
   - Standard
   - Advanced
   - Intelligent-Tiering

Weitere Informationen zu diesen Optionen finden Sie unter Angeben einer Standardparameterstufe.

6. Überprüfen Sie die Nachricht und klicken Sie dann auf Confirm (Bestätigen).

Wenn Sie die Standardstufeneinstellung später ändern möchten, wiederholen Sie diesen Vorgang und geben Sie eine andere Option für die Standardstufe an.

Angeben oder Ändern der Parameter Store-Standardstufe (AWS CLI)

Das folgende Verfahren zeigt, wie Sie mit dem AWS CLI die Standardeinstellung der Parameterschicht für den aktuellen Wert AWS-Konto und ändern können AWS-Region.

So geben Sie die Parameter Store-Standardstufe mithilfe der AWS CLI an oder ändern sie

1. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus, um die Standardeinstellung für die Parameterebene für eine bestimmte AWS-Region Zeile zu ändern AWS-Konto.

   ```
   aws ssm update-service-setting --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/parameter-store/default-parameter-tier --setting-value tier-option
   ```

   Region steht für den Bezeichner für eine Region AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

   Tier-Option (Stufenoption)-Werte sind `Standard`, `Advanced` und `Intelligent-Tiering`. Weitere Informationen zu diesen Optionen finden Sie unter Angeben einer Standardparameterstufe.

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Standardeinstellungen für den Dienst auf Parameterebene für Parameter Store die aktuelle Version AWS-Konto und anzuzeigen AWS-Region.

   ```
   aws ssm get-service-setting --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/parameter-store/default-parameter-tier
   ```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "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"
    }
}
```

Wenn Sie die Standardstufeneinstellung erneut ändern möchten, wiederholen Sie diesen Vorgang und geben Sie eine andere `SettingValue`-Option an.

Angeben oder Ändern der Parameter Store Standardstufe (PowerShell)

Das folgende Verfahren zeigt, wie Sie die Tools für Windows verwenden, PowerShell um die Standardeinstellung für die Parameterstufe für ein bestimmtes Konto AWS-Region in einem Amazon Web Services Services-Konto zu ändern.

Um die Parameter Store Standardstufe anzugeben oder zu ändern, verwenden Sie PowerShell

1. Ändern Sie die Parameter Store Standardstufe in der aktuellen AWS-Konto und AWS-Region mithilfe von AWS Tools for PowerShell (Tools für PowerShell).

   ```
   Update-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/
   ssm/parameter-store/default-parameter-tier" -SettingValue "tier-option" -
   Region region
   ```

   `Region` steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

Tier-Option (Stufenoption)-Werte sind `Standard`, `Advanced` und `Intelligent-Tiering`. Weitere Informationen zu diesen Optionen finden Sie unter Angeben einer Standardparameterstufe.

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Standardeinstellungen für den Dienst auf Parameterebene für Parameter Store die aktuelle Version AWS-Konto und anzuzeigen AWS-Region.

```
Get-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/
parameter-store/default-parameter-tier" -Region region
```

region steht für den Bezeichner einer Region AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
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
```

Wenn Sie die Standardstufeneinstellung erneut ändern möchten, wiederholen Sie diesen Vorgang und geben Sie eine andere `SettingValue`-Option an.

Ändern eines Standardparameters in einen fortgeschrittenen Parameter

Gehen Sie wie folgt vor, um einen vorhandenen Standardparameter in einen erweiterten Parameter zu ändern. Weitere Informationen zum Erstellen eines neuen erweiterten Parameters finden Sie unter Erstellen von Systems Manager-Parametern.

So ändern Sie einen Standardparameter in einen erweiterten Parameter

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie einen Parameter aus und klicken Sie dann auf Edit (Bearbeiten).
4. Geben Sie unter Description (Beschreibung) Informationen zu diesem Parameter ein.
5. Wählen Sie Advanced (Erweitert) aus.
6. Geben Sie unter Value (Wert) den Wert dieses Parameters ein. Erweiterte Parameter haben ein maximales Wertlimit von 8 KB.
7. Wählen Sie Save Changes.

Durchsatz erhöhen oder zurücksetzen Parameter Store

Durch Erhöhung des Parameter Store Durchsatzes steigt die maximale Anzahl von Transaktionen pro Sekunde (TPS)Parameter Store, die bei einer Kapazität von AWS Systems Manager verarbeitet werden können. Ein erhöhter Durchsatz ermöglicht Ihnen, Parameter Store mit höheren Volumina zur Unterstützung von Anwendungen und Arbeitslasten zu betreiben, die gleichzeitigen Zugriff auf mehrere Parameter benötigen. Sie können das Kontingent auf der Registerkarte Einstellungen bis zum maximalen Durchsatz erhöhen.

Weitere Informationen zum maximalen Durchsatz, zum Standard und zu den Höchstgrenzen finden Sie unter AWS Systems Manager Endpunkte und Kontingente.

Wenn Sie das Durchsatzkontingent erhöhen, wird Ihnen eine Gebühr berechnet. AWS-Konto Weitere Informationen finden Sie unter AWS Systems Manager -Preisgestaltung.

> **Note**
>
> Die Parameter Store Durchsatzeinstellung gilt für alle Transaktionen, die von allen IAM-Benutzern in der aktuellen AWS-Konto Version und erstellt wurden. AWS-Region Die Durchsatzeinstellung gilt für Standard- und erweiterte Parameter.

Themen

- Konfiguration von Berechtigungen zur Änderung des Durchsatzes Parameter Store

- Durchsatz erhöhen oder zurücksetzen (Konsole)
- Durchsatz erhöhen oder zurücksetzen ()AWS CLI
- Durchsatz erhöhen oder zurücksetzen () PowerShell

Konfiguration von Berechtigungen zur Änderung des Durchsatzes Parameter Store

Stellen Sie sicher, dass Sie in IAM berechtigt sind, den Parameter Store Durchsatz zu ändern, indem Sie einen der folgenden Schritte ausführen:

- Stellen Sie sicher, dass die `AdministratorAccess`-Richtlinie Ihrer IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) angefügt ist.
- Stellen Sie sicher, dass Sie über die Berechtigung zum Ändern des Servicedurchsatzes verfügen, indem Sie die folgenden API-Operationen verwenden:
  - GetServiceSetting
  - UpdateServiceSetting
  - ResetServiceSetting

Gewähren Sie der IAM-Entität die folgenden Berechtigungen, damit ein Benutzer die Durchsatzeinstellung für Parameter in einer bestimmten AWS-Region in einem AWS-Konto anzeigen und ändern kann.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:region:account-id:servicesetting/ssm/parameter-
store/high-throughput-enabled"
```

```
        }
    ]
}
```

Administratoren können schreibgeschützte Berechtigungen festlegen, indem sie die folgenden Berechtigungen zuweisen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "*"
        }
    ]
}
```

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter Erstellen einer Rolle für einen externen Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter Erstellen einer Rolle für einen IAM-Benutzer im IAM-Benutzerhandbuch.

- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Durchsatz erhöhen oder zurücksetzen (Konsole)

Das folgende Verfahren zeigt, wie Sie mithilfe von Systems Manager die Anzahl der Transaktionen pro Sekunde, die Parameter Store verarbeiten kann, für das aktuelle AWS-Konto und die AWS-Region erhöhen können. Außerdem wird gezeigt, wie Sie zu den Standardeinstellungen zurückkehren können, wenn Sie keinen erhöhten Durchsatz mehr benötigen oder keine Gebühren mehr anfallen möchten.

> **Tip**
>
> Wenn Sie noch keinen Parameter erstellt haben, können Sie das AWS Command Line Interface (AWS CLI) oder verwenden, um den Durchsatz AWS Tools for Windows PowerShell zu erhöhen. Weitere Informationen finden Sie unter Durchsatz erhöhen oder zurücksetzen ()AWS CLI und Durchsatz erhöhen oder zurücksetzen () PowerShell.

Um den Parameter Store Durchsatz zu erhöhen oder zurückzusetzen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie die Registerkarte Settings.
4. Um den Durchsatz zu erhöhen, wählen Sie „Limit festlegen“.

   –oder–

   Um zum Standardlimit zurückzukehren, wählen Sie Limit zurücksetzen.
5. Wenn Sie das Limit erhöhen, gehen Sie wie folgt vor:

   - Aktivieren Sie das Kontrollkästchen Ich akzeptiere, dass durch das Ändern dieser Einstellung Gebühren auf meinem AWS-Konto Konto anfallen.
   - Wählen Sie Set limit (Limit festlegen) aus.

–oder–

Wenn Sie das Limit auf die Standardeinstellung zurücksetzen, gehen Sie wie folgt vor:

- Aktivieren Sie das Kontrollkästchen Ich akzeptiere, dass das Zurücksetzen auf das Standard-Durchsatzlimit Parameter Store dazu führt, dass weniger Transaktionen pro Sekunde verarbeitet werden.
- Wählen Sie Limit zurücksetzen.

Durchsatz erhöhen oder zurücksetzen ()AWS CLI

Das folgende Verfahren zeigt, wie Sie mithilfe von AWS CLI die Anzahl der Transaktionen pro Sekunde erhöhen Parameter Store können, die für den aktuellen AWS-Konto und verarbeitet werden können. AWS-Region Sie können auch zum Standardlimit zurückkehren.

Um den Parameter Store Durchsatz zu erhöhen, verwenden Sie AWS CLI

1. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus, um die Anzahl der Transaktionen pro Sekunde zu erhöhen, die im aktuellen AWS-Konto und verarbeitet werden Parameter Store können AWS-Region.

   ```
   aws ssm update-service-setting --setting-id arn:aws:ssm:region:account-
   id:servicesetting/ssm/parameter-store/high-throughput-enabled --setting-value true
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Durchsatz-Serviceeinstellungen für Parameter Store im aktuellen AWS-Konto und anzuzeigen AWS-Region.

   ```
   aws ssm get-service-setting --setting-id arn:aws:ssm:region:account-
   id:servicesetting/ssm/parameter-store/high-throughput-enabled
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden:

   ```
   {
       "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"
    }
}
```

Wenn Sie den erhöhten Durchsatz nicht mehr benötigen oder Kosten vermeiden wollen, können Sie die Standardeinstellungen wiederherstellen. Um Ihre Einstellungen wiederherzustellen, führen Sie den folgenden Befehl aus.

```
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"
    }
}
```

Durchsatz erhöhen oder zurücksetzen () PowerShell

Das folgende Verfahren zeigt, wie Sie mithilfe der Tools für Windows PowerShell die Anzahl der Transaktionen pro Sekunde erhöhen, die für den aktuellen AWS-Konto und verarbeitet Parameter Store werden können. AWS-Region Sie können auch zum Standardlimit zurückkehren.

Um den Parameter Store Durchsatz zu erhöhen, verwenden Sie PowerShell

1. Erhöhen Sie den Parameter Store Durchsatz im aktuellen AWS-Konto und AWS-Region mithilfe von AWS Tools for PowerShell (Tools für PowerShell).

```
Update-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/
ssm/parameter-store/high-throughput-enabled" -SettingValue "true" -Region region
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Durchsatz-Serviceeinstellungen für Parameter Store die aktuelle Version AWS-Konto und anzuzeigen AWS-Region.

```
Get-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/
parameter-store/high-throughput-enabled" -Region region
```

Die von den Systemen zurückgegebenen Informationen ähneln den Folgenden:

```
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
```

Wenn Sie den erhöhten Durchsatz nicht mehr benötigen oder Kosten vermeiden wollen, können Sie die Standardeinstellungen wiederherstellen. Um Ihre Einstellungen wiederherzustellen, führen Sie den folgenden Befehl aus.

```
Reset-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/
parameter-store/high-throughput-enabled" -Region region
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden:

```
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
```

Einrichten von Benachrichtigungen oder Auslöseraktionen basierend auf Parameter Store-Ereignissen

In den Themen in diesem Abschnitt wird erläutert, wie Sie Amazon EventBridge und Amazon Simple Notification Service (Amazon SNS) verwenden, um Sie über Änderungen an -AWS Systems ManagerParametern zu informieren. Sie können eine EventBridge Regel erstellen, die Sie benachrichtigt, wenn ein Parameter oder eine Parameterbezeichnungsversion erstellt, aktualisiert oder gelöscht wird. Ereignisse werden auf bestmögliche Weise ausgegeben. Sie können über Änderungen oder den Status der Parameterrichtlinien benachrichtigt werden, wenn beispielsweise ein Parameter abgelaufen ist, bald abläuft oder sich seit einem bestimmten Zeitraum nicht geändert hat.

> **Note**
>
> Parameterrichtlinien sind nur verfügbar für Parameter, die das Kontingent für erweiterte Parameter verwenden. Gebührenpflichtig. Weitere Informationen finden Sie unter Zuweisen von Parameterrichtlinien und Verwalten von Parameterstufen.

Das Thema in diesem Abschnitt erläutert zudem, wie Sie andere Aktionen auf einem Ziel anhand bestimmter Parameter-Ereignisse auslösen. So können Sie z. B. eine AWS Lambda-Funktion ausführen, die einen Parameter automatisch neu erstellt, wenn dieser abgelaufen ist oder gelöscht wird. Außerdem können Sie eine Benachrichtigung einrichten, die eine Lambda-Funktion aufruft, wenn dass Passwort für Ihre Datenbank aktualisiert wird. Die Lambda-Funktion kann das Zurücksetzen oder erneute Herstellen einer Verbindung mit dem neuen Passwort erzwingen. unterstützt EventBridge auch das Ausführen von Run Command Befehlen und Automation-Ausführungen sowie Aktionen in vielen anderen AWS-Services. Run Command und Automation sind beide Funktionen von AWS Systems Manager. Weitere Informationen finden Sie im Amazon-EventBridge Benutzerhandbuch.

Bevor Sie beginnen

Erstellen Sie alle Ressourcen, die Sie zum Festlegen der Zielaktion für die Regel benötigen, die Sie erstellen möchten. Möchten Sie beispielsweise eine Regel zum Senden von Benachrichtigungen erstellen, müssen Sie zunächst ein Amazon SNS-Thema anlegen. Weitere Informationen finden Sie unter Erste Schritte mit Amazon SNS im Benutzerhandbuch für Amazon Simple Notification Service.

Konfigurieren von EventBridge Regeln für Parameter und Parameterrichtlinien

Dieses Thema erklärt Folgendes:

- So erstellen Sie eine - EventBridge Regel, die ein Ziel basierend auf Ereignissen aufruft, die mit einem oder mehreren Parametern in Ihrem geschehenAWS-Konto.
- So erstellen Sie EventBridge Regeln, die Ziele basierend auf Ereignissen aufrufen, die mit einer oder mehreren Parameterrichtlinien in Ihrem geschehenAWS-Konto. Wenn Sie einen erweiterten Parameter erstellen, geben Sie an, wann ein Parameter abläuft, wann eine Benachrichtigung gesendet wird, dass ein Parameter abläuft, und wie lange gewartet werden soll, bevor eine Benachrichtigung darüber gesendet wird, dass ein Parameter sich nicht verändert hat. Sie richten die Benachrichtigungen für diese Ereignisse anhand der folgenden Schritte ein. Weitere Informationen finden Sie unter Zuweisen von Parameterrichtlinien und Verwalten von Parameterstufen.

So konfigurieren Sie eine - EventBridge Regel für einen Systems Manager-Parameter oder eine Parameterrichtlinie

1. Öffnen Sie die Amazon- EventBridge Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules (Regeln) und anschließend Create rule (Regel erstellen) aus.

   –oder–

   Wenn sich die EventBridge Startseite zuerst öffnet, wählen Sie Regel erstellen aus.
3. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
4. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel bei übereinstimmenden Ereignissen, die von Ihrem eigenen AWS-Konto stammen, ausgelöst wird, wählen Sie Standard. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es stets an den standardmäßigen Event Bus Ihres Kontos weitergeleitet.
5. Lassen Sie für Rule type (Regeltyp) die Standardoption Rule with an event pattern (Regel mit einem Ereignismuster) ausgewählt.
6. Wählen Sie Weiter aus.
7. Lassen Sie für Ereignisquelle die AWS Standardereignisse oder EventBridge Partnerereignisse ausgewählt. Sie können den Abschnitt Beispielereignis überspringen.
8. Gehen Sie bei Event pattern (Ereignismuster) wie folgt vor:

- Wählen Sie Custom patterns (JSON editor) (Benutzerdefinierte Muster (JSON-Editor)) aus.
- Fügen Sie für Event pattern (Ereignismuster) einen der folgenden Inhalte in das Feld ein, je nachdem, ob Sie eine Regel für einen Parameter oder eine Parameter-Richtlinie erstellen:

  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"
        ]
    }
}
```

- Ändern Sie den Inhalt für die Parameter und die Operationen, auf die Sie reagieren möchten, wie in den folgenden Beispielen gezeigt.

  Parameter

  In diesem Beispiel wird eine Aktion ausgeführt, wenn einer der Parameter namens `/Oncall` und `/Project/Teamlead` aktualisiert wird:

```
{
    "source": [
        "aws.ssm"
    ],
    "detail-type": [
        "Parameter Store Change"
    ],
    "detail": {
        "name": [
            "/Oncall",
            "/Project/Teamlead"
        ],
        "operation": [
            "Update"
        ]
    }
}
```

  Parameter policy

  In diesem Beispiel wird immer dann eine Aktion ausgeführt, wenn der Parameter mit dem Namen `/OncallDuties` abläuft und gelöscht wird:

```
{
    "source": [
        "aws.ssm"
```

```
    ],
    "detail-type": [
        "Parameter Store Policy Action"
    ],
    "detail": {
        "parameter-name": [
            "/OncallDuties"
        ],
        "policy-type": [
            "Expiration"
        ]
    }
}
```

9. Wählen Sie Weiter aus.
10. Für Target 1 (Ziel 1) wählen Sie einen Zieltyp und eine unterstützte Ressource aus. Wenn Sie beispielsweise SNS-Thema auswählen, treffen Sie eine Auswahl für Topic (Thema). Wenn Sie wählenCodePipeline, geben Sie einen Pipeline-ARN für Pipeline-ARN ein. Geben Sie bei Bedarf zusätzliche Konfigurationswerte an.

> **Tip**
>
> Wählen Sie Add another target (Weiteres Ziel hinzufügen), wenn Sie zusätzliche Ziele für die Regel benötigen.

11. Wählen Sie Weiter aus.
12. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Amazon EventBridge-Tags im Amazon- EventBridge Benutzerhandbuch.
13. Wählen Sie Weiter aus.
14. Wählen Sie Regel erstellen aus.

Weitere Informationen

- Verwenden von Parameterbezeichnungen für die einfache Aktualisierung der Konfiguration über mehrere Umgebungen hinweg
- Tutorial: Verwenden EventBridge von zum Weiterleiten von Ereignissen an AWS Systems ManagerRun Command im Amazon EventBridge -Benutzerhandbuch
- Tutorial: Festlegen von AWS Systems Manager Automation als EventBridge Ziel im Amazon EventBridge -Benutzerhandbuch

Arbeiten mit Parameter Store

In diesem Abschnitt wird beschrieben, wie Sie Parameter organisieren, erstellen und markieren und verschiedene Versionen von Parametern erstellen. Sie können die AWS Systems Manager Konsole, die Amazon Elastic Compute Cloud (Amazon EC2)-Konsole oder die AWS Command Line Interface (AWS CLI) verwenden, um Parameter zu erstellen und mit ihnen zu arbeiten. Weitere Informationen zu Parametern finden Sie unter Was ist ein Parameter?

Themen

- Erstellen von Systems Manager-Parametern
- Nach Systems Manager-Parametern suchen
- Zuweisen von Parameterrichtlinien
- Arbeiten mit Parameterhierarchien
- Arbeiten mit Parameterbezeichnungen
- Arbeiten mit Parameterversionen
- Mit gemeinsam genutzten Parametern arbeiten
- Arbeiten mit Parametern unter Verwendung von Run Command-Befehlen
- Unterstützung für native Parameter für Amazon Machine Image-IDs
- Löschen von Systems-Manager-Parametern

Erstellen von Systems Manager-Parametern

Verwenden Sie die Informationen in den folgenden Themen, um Ihnen beim Erstellen von Systems Manager-Parametern über die AWS Systems Manager-Konsole, der AWS Command Line Interface (AWS CLI) oder den AWS Tools for Windows PowerShell (Tools for Windows PowerShell) zu helfen.

Dieser Abschnitt zeigt, wie Sie Parameter mit Parameter Store in einer Testumgebung erstellen, speichern und ausführen können. Es zeigt auch, wie Sie Parameter Store mit anderen Systems-Manager-Funktionen und AWS-Services verwenden. Weitere Informationen finden Sie unter Was ist ein Parameter?

Anforderungen und Einschränkungen für Parameternamen

Die Informationen in diesem Thema sind hilfreich bei der Angabe gültiger Werte für Parameternamen, wenn Sie einen Parameter erstellen.

Diese Informationen ergänzen die Details im Thema PutParameter in der AWS Systems Manager API-Referenz, das auch Informationen zu den Werten AllowedPattern, Description, KeyId, Overwrite, Type und Value bereitstellt.

Die Anforderungen und Einschränkungen für Parameternamen umfassen Folgendes:

- Berücksichtigung der Groß-/Kleinschreibung: Bei Parameternamen werden Groß- und Kleinschreibung berücksichtigt.
- Leerstellen: Parameternamen dürfen keine Leerzeichen enthalten.
- Gültige Zeichen: Parameternamen können nur die folgenden Symbole und Buchstaben enthalten: `a-zA-Z0-9_.-`

  Darüber hinaus wird der Schrägstrich (/) verwendet, um Hierarchien in Parameternamen zu beschreiben. Zum Beispiel: `/Dev/Production/East/Project-ABC/MyParameter`
- Gültiges AMI-Format: Wenn Sie `aws:ec2:image` als Datentyp für einen `String`-Parameter wählen, muss die eingegebene ID für das AMI-ID-Format `ami-12345abcdeEXAMPLE` validiert werden.
- Vollständig qualifiziert: Wenn Sie einen Parameter in einer Hierarchie anlegen oder darauf verweisen, müssen Sie einen vorangehenden Schrägstrich (/) einfügen. Wenn Sie auf einen Parameter verweisen, der Teil einer Hierarchie ist, müssen Sie den gesamten Hierarchiepfad einschließlich des ersten Schrägstrichs (/) angeben.
  - Vollständig qualifizierte Parameternamen: `MyParameter1`, `/MyParameter2`, `/Dev/Production/East/Project-ABC/MyParameter`
  - Nicht vollständig qualifizierter Parametername: `MyParameter3/L1`
- Länge: Die maximale Länge für einen Parameternamen, den Sie erstellen, beträgt 1011 Zeichen. Dazu gehören die Zeichen im ARN, die vor dem von Ihnen angegebenen Namen stehen, z. B. `arn:aws:ssm:us-east-2:111122223333:parameter/`.
- Präfixe: Einem Parameternamen darf kein „`aws`“- oder „`ssm`„“-Präfix vorangestellt werden (ohne Berücksichtigung der Groß-/Kleinschreibung). Beispiel: Versuche, Parameter mit den folgenden Namen zu erstellen, schlagen mit einer Ausnahme fehl:
  - `awsTestParameter`
  - `SSM-testparameter`
  - `/aws/testparam1`

> **Note**
>
> Wenn Sie einen Parameter in einem SSM-Dokument, -Befehl oder -Skript angeben, schließen Sie `ssm` als Teil der Syntax ein. Zum Beispiel {{ssm:*parameter-name*}} und {{ ssm:*parameter-name* }}, z. B. `{{ssm:MyParameter}}` und `{{ ssm:MyParameter }}`.

- Eindeutigkeit: Ein Parametername muss innerhalb einer AWS-Regioneindeutig sein. Systems Manager behandelt beispielsweise die folgenden Parameter als separate Parameter, wenn sie sich in derselben Region befinden:
  - `/Test/TestParam1`
  - `/TestParam1`

  Die folgenden Beispiele sind ebenfalls eindeutig:
  - `/Test/TestParam1/Logpath1`
  - `/Test/TestParam1`

  Die folgenden Beispiele sind, sofern sie sich in derselben Region befinden, jedoch nicht eindeutig:
  - `/TestParam1`
  - `TestParam1`
- Hierarchietiefe: Wenn Sie eine Parameterhierarchie angeben, darf die Hierarchie maximal fünfzehn Ebenen tief sein. Sie können auf jeder Ebene der Hierarchie einen Parameter definieren. Die beiden folgenden Beispiele zeigen strukturell gültige Parameter:
  - `/Level-1/L2/L3/L4/L5/L6/L7/L8/L9/L10/L11/L12/L13/L14/parameter-name`
  - `parameter-name`

  Der Versuch, den folgenden Parameter zu erstellen, löst eine `HierarchyLevelLimitExceededException`-Ausnahme aus:
  - `/Level-1/L2/L3/L4/L5/L6/L7/L8/L9/L10/L11/L12/L13/L14/L15/L16/parameter-name`

> **Important**
>
> Wenn ein Benutzer Zugriff auf einen Pfad hat, kann er auf alle Ebenen dieses Pfads zugreifen. Wenn ein Benutzer beispielsweise die Berechtigung für den Zugriff auf den Pfad

`/a` besitzt, dann kann er auch auf `/a/b` zugreifen. Selbst wenn einem Benutzer in AWS Identity and Access Management (IAM) der Zugriff auf den Parameter `/a/b` ausdrücklich verweigert wurde, kann er dennoch die API-Operation GetParametersByPath rekursiv für `/a` aufrufen und `/a/b` anzeigen.

Themen

- Erstellen eines Systems Manager-Parameters (Konsole)
- Erstellen eines Systems Manager-Parameters (AWS CLI)
- Erstellen eines Systems Manager-Parameters (Tools for Windows PowerShell)

Erstellen eines Systems Manager-Parameters (Konsole)

Sie können die AWS Systems Manager Konsole verwenden, um `SecureString` Parametertypen zu erstellen und auszuführen`String`. `StringList` Warten Sie nach dem Löschen eines Parameters mindestens 30 Sekunden, um einen Parameter mit dem gleichen Namen zu erstellen.

> **Note**
>
> Parameter sind nur dort verfügbar AWS-Region , wo sie erstellt wurden.

Das folgende Verfahren führt Sie durch die Schritte zum Erstellen eines Parameters mithilfe der Parameter Store-Konsole. Sie können `String`-, `StringList`- und `SecureString`-Parametertypen über die Konsole erstellen.

So erstellen Sie einen Parameter

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie Create parameter (Parameter erstellen) aus.
4. Geben Sie in das Feld Name (Name) eine Hierarchie und einen Namen ein. Geben Sie z. B. ei **`/Test/helloWorld`**.

   Weitere Informationen zu Parameterhierarchien finden Sie unter Arbeiten mit Parameterhierarchien.

5. Geben Sie im Feld Description (Beschreibung) eine Beschreibung ein, anhand der dieser Parameters als Test-Parameter erkannt werden kann.
6. Wählen Sie für Parameter tier (Parameterstufe) entweder Standard oder Advanced (Erweitert) aus. Weitere Informationen zu erweiterten Parameter finden Sie unter Verwalten von Parameterstufen.
7. Wählen Sie als Typ die Option String, StringList, or aus SecureString.

   - Wenn Sie String (Zeichenfolge) wählen, wird das Feld Data type (Datentyp) angezeigt. Wenn Sie einen Parameter für die Ressourcen-ID für ein Amazon Machine Image (AMI) erstellen, wählen Sie `aws:ec2:image` aus. Behalten Sie andernfalls die Standardeinstellung `text` bei.
   - Wenn Sie möchten SecureString, wird das Feld KMS-Schlüssel-ID angezeigt. Wenn Sie keine AWS Key Management Service AWS KMS key ID, keinen AWS KMS key Amazon-Ressourcennamen (ARN), einen Aliasnamen oder einen Alias-ARN angeben`alias/aws/ssm`, verwendet das System den Von AWS verwalteter Schlüssel für Systems Manager. Wenn Sie diesen Schlüssel nicht verwenden möchten, können Sie einen kundenverwalteten Schlüssel verwenden. Weitere Informationen über Von AWS verwaltete Schlüssel und kundenverwaltete Schlüssel finden Sie unter AWS Key Management Service -Konzepte im AWS Key Management Service -Entwicklerhandbuch. Weitere Informationen Parameter Store zur AWS KMS Verschlüsselung finden Sie unter AWS Systems ManagerParameter StoreAnwendungsmöglichkeiten AWS KMS.

     Important

     Parameter Store unterstützt nur KMS-Schlüssel zur symmetrischen Verschlüsselung. Sie können keinen KMS-Schlüssel zur asymmetrischen Verschlüsselung verwenden, um Ihre Parameter zu verschlüsseln. Wie Sie feststellen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, erfahren Sie unter Erkennen symmetrischer und asymmetrischer Schlüssel im AWS Key Management Service -Entwicklerhandbuch.

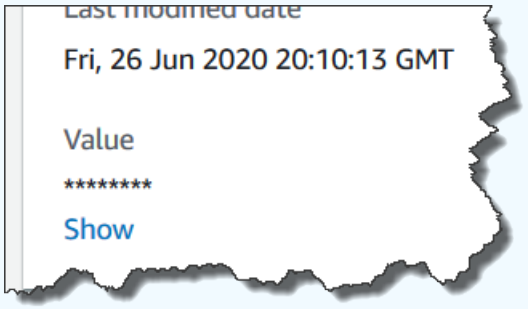
   - Wenn Sie unter Verwendung des Parameters `key-id` mit einem kundenverwalteten Schlüssel-Aliasnamen oder -Alias-ARN in der Konsole einen `SecureString`-Parameter erstellen, müssen Sie vor dem Alias das Präfix `alias/` angeben. Nachfolgend ein ARN-Beispiel.

     ```
     arn:aws:kms:us-east-2:123456789012:alias/abcd1234-ab12-cd34-ef56-abcdeEXAMPLE
     ```

     Im Folgenden finden Sie ein Beispiel für einen Aliasnamen.

```
alias/MyAliasName
```

8. Geben Sie im Feld Value (Wert) einen Wert ein. Geben Sie beispielsweise **`This is my first parameter`** oder **`ami-0dbf5ea29aEXAMPLE`** ein.

> **Note**
>
> Parameter können nicht referenziert oder in den Werten anderer Parameter verschachtelt werden. Sie können `{{}}` oder `{{ssm:parameter-name}}` nicht in einen Parameterwert aufnehmen.
> Wenn Sie sich dafür entscheiden SecureString, wird der Wert des Parameters standardmäßig maskiert („******“), wenn Sie ihn später auf der Registerkarte Parameterübersicht anzeigen. Klicken Sie auf Anzeigen, um den Parameterwert anzuzeigen.
>
> 
>

9. (Optional) Wenden Sie im Bereich Tags ein oder mehrere Tag-Schlüssel-Wert-Paare auf den Parameter an.

   Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise einen Systems Manager-Parameter markieren, um den Typ der Ressource, für die er gilt, die Umgebung oder den Typ der Konfigurationsdaten, auf die vom Parameter verwiesen wird, zu identifizieren. In diesem Fall können Sie die folgenden Schlüssel-Wert-Paare angeben:

   - `Key=Resource,Value=S3bucket`
   - `Key=OS,Value=Windows`
   - `Key=ParameterType,Value=LicenseKey`

10. Wählen Sie Create parameter (Parameter erstellen) aus.

11. Wählen Sie in der Liste der Parameter den Namen des Parameters aus, den Sie gerade erstellt haben. Überprüfen Sie die Details auf der Registerkarte Overview. Wenn Sie einen `SecureString`-Parameter erstellt haben, wählen Sie Show aus, um die unverschlüsselten Werte anzuzeigen.

> **Note**
>
> Sie können einen erweiterten Parameter nicht in einen Standardparameter ändern. Wenn Sie einen erweiterten Parameter nicht mehr benötigen oder verhindern wollen, dass weitere Gebühren dafür anfallen, löschen Sie ihn und erstellen Sie ihn als Standardparameter neu.

Erstellen eines Systems Manager-Parameters (AWS CLI)

Sie können mit der AWS Command Line Interface (AWS CLI) `String`-, `StringList`- und `SecureString`-Parametertypen erstellen. Warten Sie nach dem Löschen eines Parameters mindestens 30 Sekunden, um einen Parameter mit dem gleichen Namen zu erstellen.

Parameter können nicht referenziert oder in den Werten anderer Parameter verschachtelt werden. Sie können `{{}}` oder `{{ssm:parameter-name}}` nicht in einen Parameterwert aufnehmen.

> **Note**
>
> Parameter sind nur in den AWS-Region verfügbar, in denen sie erstellt wurden.

Themen

- Erstellen eines String-Parameters (AWS CLI)
- Erstellen eines StringList-Parameters (AWS CLI)
- Erstellen eines SecureString-Parameters (AWS CLI)
- Erstellen eines mehrzeiligen Parameters (AWS CLI)

Erstellen eines **`String`**-Parameters (AWS CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um einen Parameter vom `String`-Typ zu erstellen. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

   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"
   ```

   –oder–

   Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen, der eine Amazon Machine Image (AMI)-ID als Parameterwert enthält.

   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"
```

Die Option `--name` unterstützt Hierarchien. Weitere Informationen zu Hierarchien finden Sie unter Arbeiten mit Parameterhierarchien.

Die Option `--data-type` muss nur angegeben werden, wenn Sie einen Parameter erstellen, der eine AMI-ID enthält. Es wird überprüft, dass der eingegebene Parameterwert eine ordnungsgemäß formatierte Amazon Elastic Compute Cloud (Amazon EC2) AMI-ID ist. Für alle anderen Parameter lautet der Standarddatentyp `text` und Sie können optional einen Wert angeben. Weitere Informationen finden Sie unter Unterstützung für native Parameter für Amazon Machine Image-IDs.

> **Important**
>
> Bei Erfolg gibt der Befehl die Versionsnummer des Parameters zurück. Ausnahme: Wenn Sie `aws:ec2:image` als Datentyp angegeben haben, bedeutet eine neue Versionsnummer in der Antwort nicht, dass der Parameterwert bereits validiert wurde. Weitere Informationen finden Sie unter Unterstützung für native Parameter für Amazon Machine Image-IDs.

Im folgenden Beispiel werden einem Parameter die Tags zweier Schlüssel-Wert-Paare hinzugefügt.

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\"}]
```

Im folgenden Beispiel wird eine Parameterhierarchie im Namen verwendet, um einen `String`-Klartext-Parameter zu erstellen. Er gibt die Versionsnummer des Parameters zurück. Weitere Informationen zu Parameterhierarchien finden Sie unter Arbeiten mit Parameterhierarchien.

Linux & macOS

Parameter nicht in einer Hierarchie

```
aws ssm put-parameter \
    --name "golden-ami" \
    --type "String" \
    --value "ami-12345abcdeEXAMPLE"
```

Parameter in einer Hierarchie

```
aws ssm put-parameter \
    --name "/amis/linux/golden-ami" \
    --type "String" \
    --value "ami-12345abcdeEXAMPLE"
```

Windows

Parameter nicht in einer Hierarchie

```
aws ssm put-parameter ^
    --name "golden-ami" ^
    --type "String" ^
    --value "ami-12345abcdeEXAMPLE"
```

Parameter in einer Hierarchie

```
aws ssm put-parameter ^
    --name "/amis/windows/golden-ami" ^
    --type "String" ^
    --value "ami-12345abcdeEXAMPLE"
```

3. Führen Sie den folgenden Befehl aus, um den letzten Parameterwert anzuzeigen und die Details des neuen Parameters zu überprüfen.

```
aws ssm get-parameters --names "/Test/IAD/helloWorld"
```

Das System gibt unter anderem folgende Informationen zurück

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

Führen Sie den folgenden Befehl aus, um den Parameterwert zu ändern. Er gibt die Versionsnummer des Parameters zurück.

```
aws ssm put-parameter --name "/Test/IAD/helloWorld" --value "My updated 1st parameter"
 --type String --overwrite
```

Führen Sie den folgenden Befehl aus, um den Verlauf der Parameterwerte anzuzeigen.

```
aws ssm get-parameter-history --name "/Test/IAD/helloWorld"
```

Führen Sie den folgenden Befehl aus, um diesen Parameter in einem Befehl zu verwenden.

```
aws ssm send-command --document-name "AWS-RunShellScript" --parameters '{"commands":
["echo {{ssm:/Test/IAD/helloWorld}}"]}' --targets "Key=instanceids,Values=instance-ids"
```

Führen Sie den folgenden Befehl aus, wenn Sie nur den Parameterwert abrufen möchten.

```
aws ssm get-parameter --name testDataTypeParameter --query "Parameter.Value"
```

Führen Sie den folgenden Befehl aus, wenn Sie nur den Parameterwert mithilfe von `get-parameters` abrufen möchten.

```
aws ssm get-parameters --names "testDataTypeParameter" --query "Parameters[*].Value"
```

Führen Sie den folgenden Befehl aus, um die Metadaten zum Parameter anzuzeigen.

```
aws ssm describe-parameters --filters "Key=Name,Values=/Test/IAD/helloWorld"
```

> **Note**
>
> Der Name muss großgeschrieben werden.

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Parameters": [
        {
            "Name": "helloworld",
            "Type": "String",
            "LastModifiedUser": "arn:aws:iam::123456789012:user/JohnDoe",
            "LastModifiedDate": 1494529763.156,
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Erstellen eines **`StringList`**-Parameters (AWS CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

   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**
   >
   > Bei Erfolg gibt der Befehl die Versionsnummer des Parameters zurück.

   In diesem Beispiel werden die Tags zweier Schlüssel-Wert-Paare an ein Parameter hinzugefügt. (Führen Sie, abhängig von der Art des Betriebssystems auf Ihrem lokalen Computer, einen der folgenden Befehle aus. Die von einer lokalen Windows-Maschine auszuführende Version enthält die Escape-Zeichen ("\"), mit denen Sie den Befehl von Ihrem Befehlszeilen-Tool aus ausführen.)

   Im folgenden Beispiel für `StringList` wird eine Parameterhierarchie verwendet.

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**
>
> Die Elemente einer `StringList` müssen durch ein Komma (,) getrennt werden. Sie können keine anderen Satzzeichen oder Sonderzeichen als Escape-Zeichen für Elemente in der Liste verwenden. Verwenden Sie den Typ `String`, wenn ein Parameterwert ein Komma erfordert.

3. Führen Sie den Befehl `get-parameters` aus, um die Details zu einem Parameter zu überprüfen. Zum Beispiel:

```
aws ssm get-parameters --name "/IAD/ERP/Oracle/addUsers"
```

Erstellen eines SecureString-Parameters (AWS CLI)

Gehen Sie folgendermaßen vor, um einen `SecureString`-Parameter zu erstellen. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

> **Important**
>
> Nur der Wert eines `SecureString`-Parameters wird verschlüsselt. Der Name des Parameters, die Beschreibung und andere Eigenschaften sind nicht verschlüsselt.

> **Important**
>
> Parameter Store unterstützt nur KMS-Schlüssel zur symmetrischen Verschlüsselung. Sie können keinen KMS-Schlüssel zur asymmetrischen Verschlüsselung verwenden, um Ihre Parameter zu verschlüsseln. Wie Sie feststellen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, erfahren Sie unter Erkennen symmetrischer und asymmetrischer Schlüssel im AWS Key Management Service-Entwicklerhandbuch.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie einen der folgenden Befehle aus, um einen Parameter zu erstellen, der den Datentyp `SecureString` verwendet.

   Linux & macOS

   Erstellen eines **`SecureString`**-Parameters mit dem standardmäßigen Von AWS verwalteter Schlüssel

   ```
   aws ssm put-parameter \
       --name "parameter-name" \
       --value "parameter-value" \
       --type "SecureString"
   ```

   Erstellen eines **`SecureString`**-Parameters, der einen vom Kunden verwalteten Schlüssel verwendet

   ```
   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"
   ```

   Erstellen eines **`SecureString`**-Parameters, der einen benutzerdefinierten AWS KMS-Schlüssel verwendet

```
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

Erstellen eines **`SecureString`**-Parameters mit dem standardmäßigen Von AWS verwalteter Schlüssel

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "parameter-value" ^
    --type "SecureString"
```

Erstellen eines **`SecureString`**-Parameters, der einen vom Kunden verwalteten Schlüssel verwendet

```
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"
```

Erstellen eines **`SecureString`**-Parameters, der einen benutzerdefinierten AWS KMS-Schlüssel verwendet

```
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"
```

Wenn Sie einen `SecureString`-Parameter erstellen, indem Sie den Von AWS verwalteter Schlüssel-Schlüssel in Ihrem Konto und Ihrer Region verwenden, müssen Sie keinen Wert für den `--key-id`-Parameter angeben.

> **Note**
>
> Wenn Sie den AWS KMS key verwenden möchten, der Ihrem AWS-Konto und Ihrer AWS-Region zugewiesen wurde, müssen Sie den `key-id`-Parameter in dem Befehl entfernen. Weitere Informationen zum Konfigurieren einer Regel in AWS KMS keys finden Sie unter AWS Key Management Service im AWS Key Management Service-Entwicklerhandbuch.

Wenn Sie statt des Ihrem Konto zugewiesenen Von AWS verwalteter Schlüssel einen kundenverwalteten Schlüssel verwenden möchten, müssen Sie den Schlüssel mithilfe des `--key-id`-Parameters angeben. Der Parameter unterstützt die folgenden KMS-Parameterformate.

- Beispiel für Schlüssel-Amazon-Ressourcenname (ARN):

  `arn:aws:kms:us-east-2:123456789012:key/key-id`

- Beispiel für den Alias-ARN:

  `arn:aws:kms:us-east-2:123456789012:alias/alias-name`

- Beispiel: Key-ID

  `12345678-1234-1234-1234-123456789012`

- Beispiel für den Aliasnamen:

  `alias/MyAliasName`

Sie können einen kundenverwalteten Schlüssel erstellen, indem Sie die AWS Management Console oder die AWS KMS-API verwenden. Mit den folgenden AWS CLI-Befehlen wird in der aktuellen AWS-Region Ihres AWS-Konto ein kundenverwalteter Schlüssel erstellt.

```
aws kms create-key
```

Verwenden Sie einen Befehl im folgenden Format, um einen `SecureString`-Parameter mit dem Schlüssel zu erstellen, den Sie gerade generiert haben.

Im folgenden Beispiel wird ein verschleierter Name (`3l3vat3131`) für einen Passwortparameter und einen AWS KMS key verwendet.

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. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen.

Wenn Sie keinen `with-decryption`-Parameter bzw. den `no-with-decryption`-Parameter angeben, gibt der Befehl eine verschlüsselte GUID zurück.

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. Führen Sie den folgenden Befehl aus, um die Metadaten zum Parameter anzuzeigen.

   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. Führen Sie den folgenden Befehl aus, um den Parameterwert zu ändern, wenn Sie keinen vom Kunden verwalteten AWS KMS key verwenden.

   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
   ```

   –oder–

   Führen Sie einen der folgenden Befehle aus, um den Parameterwert zu ändern, wenn Sie einen vom Kunden verwalteten AWS KMS key verwenden.

   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. Führen Sie den folgenden Befehl aus, um den letzten Parameterwert anzuzeigen.

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. Führen Sie den folgenden Befehl aus, um den Verlauf der Parameterwerte anzuzeigen.

   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**
>
> Sie können einen Parameter mit einem verschlüsselten Wert manuell erstellen. Da der Wert in diesem Fall bereits verschlüsselt ist, müssen Sie den `SecureString`-Parametertyp nicht auswählen. Wenn Sie `SecureString` dennoch auswählen, wird Ihr Parameter zweifach verschlüsselt.

Alle `SecureString`-Werte werden standardmäßig als verschlüsselter Text angezeigt. Um einen `SecureString`-Wert entschlüsseln zu können, muss ein Benutzer die Berechtigung zum Aufruf der AWS KMS-API-Operation Decrypt besitzen. Weitere Informationen zur Konfiguration der AWS KMS-Zugriffskontrolle finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS im AWS Key Management Service-Entwicklerhandbuch.

> **Important**
>
> Wenn Sie den KMS-Schlüsselalias für den KMS-Schlüssel, der zum Verschlüsseln eines Parameters verwendet wird, ändern, müssen Sie auch den Schlüsselalias aktualisieren, mit dem der Parameter AWS KMS referenziert. Dies gilt nur für den KMS-Schlüsselalias; die Schlüssel-ID, die ein Alias anfügt, bleibt unverändert, es sei denn, Sie löschen den gesamten Schlüssel.

Erstellen eines mehrzeiligen Parameters (AWS CLI)

Sie können die AWS CLI verwenden, um einen Parameter mit Zeilenumbrüchen zu erstellen. Verwenden Sie Zeilenumbrüche, um den Text in längere Parameterwerte aufzuteilen, um die Lesbarkeit zu verbessern, oder aktualisieren Sie beispielsweise den Inhalt von Parametern mit mehreren Absätzen für eine Webseite. Sie können den Inhalt in eine JSON-Datei einschließen und die `--cli-input-json`-Option verwenden, indem Sie Zeilenumbruchzeichen wie `\n` verwenden, wie im folgenden Beispiel gezeigt.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um einen mehrzeiligen Parameter zu erstellen.

   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
   ```

   Im folgenden Beispiel werden die Inhalte der Datei `MultiLineParameter.json` angezeigt.

   ```
   {
       "Value": "<para>Paragraph One</para>\n<para>Paragraph Two</para>
   \n<para>Paragraph Three</para>"
   }
   ```

Der gespeicherte Parameterwert wird wie folgt gespeichert.

```
<para>Paragraph One</para>
<para>Paragraph Two</para>
<para>Paragraph Three</para>
```

Erstellen eines Systems Manager-Parameters (Tools for Windows PowerShell)

Sie können mit AWS Tools for Windows PowerShell `String`-, `StringList`- und `SecureString`-Parametertypen erstellen. Warten Sie nach dem Löschen eines Parameters mindestens 30 Sekunden, um einen Parameter mit dem gleichen Namen zu erstellen.

Parameter können nicht referenziert oder in den Werten anderer Parameter verschachtelt werden. Sie können `{{}}` oder `{{ssm:parameter-name}}` nicht in einen Parameterwert aufnehmen.

> **Note**
>
> Parameter sind nur in den AWS-Region verfügbar, in denen sie erstellt wurden.

Themen

- Erstellen eines String-Parameters (Tools for Windows PowerShell)
- Erstellen eines StringList-Parameters (Tools for Windows PowerShell)
- Erstellen eines SecureString-Parameters (Tools for Windows PowerShell)

Erstellen eines **`String`**-Parameters (Tools for Windows PowerShell)

1. Installieren und konfigurieren Sie die AWS Tools for PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS Tools for PowerShell.

2. Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen, der einen Klartext-Wert enthält. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

   ```
   Write-SSMParameter `
       -Name "parameter-name" `
       -Value "parameter-value" `
       -Type "String"
   ```

–oder–

Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen, der eine Amazon Machine Image (AMI)-ID als Parameterwert enthält.

> **Note**
>
> Um einen Parameter mit einem Tag zu erstellen, erstellen Sie den service.model.tag vorher als Variable. Ein Beispiel.
>
> ```
> $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
```

Die Option `-DataType` muss nur angegeben werden, wenn Sie einen Parameter erstellen, der eine AMI-ID enthält. Für alle anderen Parameter lautet der Standarddatentyp `text`. Weitere Informationen finden Sie unter Unterstützung für native Parameter für Amazon Machine Image-IDs.

Im folgenden Beispiel wird eine Parameterhierarchie verwendet.

```
Write-SSMParameter `
    -Name "/IAD/Web/SQL/IPaddress" `
    -Value "99.99.99.999" `
    -Type "String" `
    -Tags $tag
```

3. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen.

```
(Get-SSMParameterValue -Name "the-parameter-name-you-specified").Parameters
```

Erstellen eines **`StringList`**-Parameters (Tools for Windows PowerShell)

1. Installieren und konfigurieren Sie die AWS Tools for PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS Tools for PowerShell.

2. Führen Sie den folgenden Befehl aus, um einen StringList-Parameter zu erstellen. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

   > **Note**
   >
   > Um einen Parameter mit einem Tag zu erstellen, erstellen Sie den service.model.tag vorher als Variable. Ein Beispiel.
   >
   > ```
   > $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
   ```

   Bei Erfolg gibt der Befehl die Versionsnummer des Parameters zurück.

   Ein Beispiel.

   ```
   Write-SSMParameter `
       -Name "stringlist-parameter" `
       -Value "Milana,Mariana,Mark,Miguel" `
       -Type "StringList" `
       -Tags $tag
   ```

   > **Note**
   >
   > Die Elemente einer `StringList` müssen durch ein Komma (,) getrennt werden. Sie können keine anderen Satzzeichen oder Sonderzeichen als Escape-Zeichen

für Elemente in der Liste verwenden. Verwenden Sie den Typ `String`, wenn ein Parameterwert ein Komma erfordert.

3. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen.

```
(Get-SSMParameterValue -Name "the-parameter-name-you-specified").Parameters
```

Erstellen eines SecureString-Parameters (Tools for Windows PowerShell)

Bevor Sie einen `SecureString`-Parameter erstellen, informieren Sie sich über die Voraussetzungen für diese Art von Parameter. Weitere Informationen finden Sie unter Erstellen eines SecureString-Parameters (AWS CLI).

Important

Nur der Wert eines `SecureString`-Parameters wird verschlüsselt. Der Name des Parameters, die Beschreibung und andere Eigenschaften sind nicht verschlüsselt.

Important

Parameter Store unterstützt nur KMS-Schlüssel zur symmetrischen Verschlüsselung. Sie können keinen KMS-Schlüssel zur asymmetrischen Verschlüsselung verwenden, um Ihre Parameter zu verschlüsseln. Wie Sie feststellen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, erfahren Sie unter Erkennen symmetrischer und asymmetrischer Schlüssel im AWS Key Management Service-Entwicklerhandbuch.

1. Installieren und konfigurieren Sie die AWS Tools for PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS Tools for PowerShell.

2. Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

> **Note**
>
> Um einen Parameter mit einem Tag zu erstellen, erstellen Sie zuerst den service.model.tag als Variable. Ein Beispiel.
>
> ```
> $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
```

Bei Erfolg gibt der Befehl die Versionsnummer des Parameters zurück.

> **Note**
>
> Zum Verwenden des Von AWS verwalteter Schlüssel, der Ihrem Konto zugewiesen wurde, müssen Sie den `-KeyId`-Parameter aus dem Befehl entfernen.

Im folgenden Beispiel wird ein verschleierter Name (3l3vat3131) für einen Passwortparameter und einen Von AWS verwalteter Schlüssel verwendet.

```
Write-SSMParameter `
    -Name "/Finance/Payroll/3l3vat3131" `
    -Value "P@sSwW)rd" `
    -Type "SecureString"`
    -Tags $tag
```

3. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen.

```
(Get-SSMParameterValue -Name "the-parameter-name-you-specified" -WithDecryption
 $true).Parameters
```

Alle `SecureString`-Werte werden standardmäßig als verschlüsselter Text angezeigt. Um einen `SecureString`-Wert entschlüsseln zu können, muss ein Benutzer die Berechtigung zum Aufruf der AWS KMS-API-Operation Decrypt besitzen. Weitere Informationen zur Konfiguration der AWS KMS-Zugriffskontrolle finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS im AWS Key Management Service-Entwicklerhandbuch.

> **Important**
>
> Wenn Sie den KMS-Schlüsselalias für den KMS-Schlüssel, der zum Verschlüsseln eines Parameters verwendet wird, ändern, müssen Sie auch den Schlüsselalias aktualisieren, mit dem der Parameter AWS KMS referenziert. Dies gilt nur für den KMS-Schlüsselalias; die Schlüssel-ID, die ein Alias anfügt, bleibt unverändert, es sei denn, Sie löschen den gesamten Schlüssel.

Nach Systems Manager-Parametern suchen

Wenn Sie viele Parameter in Ihrem Konto haben, kann es schwierig sein, Informationen zu nur einem oder einigen wenigen Parametern gleichzeitig zu finden. In diesem Fall können Sie Filterwerkzeuge verwenden, um mithilfe von Suchkriterien nach den gewünschten Parametern zu suchen. Sie können die AWS Systems Manager Konsole, die AWS Command Line Interface (AWS CLI), die oder die DescribeParametersAPI verwenden AWS Tools for PowerShell, um nach Parametern zu suchen.

Themen

- Suchen nach einem Parameter (Konsole)
- Suchen nach einem Parameter (AWS CLI)

Suchen nach einem Parameter (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.

3. Wählen Sie das Suchfeld und die gewünschte Suchmethode aus. Zum Beispiel `Type` oder `Name`.
4. Geben Sie Informationen für den ausgewählten Suchtyp an. Zum Beispiel:
   - Wenn Sie nach `Type` suchen, wählen Sie `String`, `StringList` oder `SecureString` aus.
   - Wenn Sie nach `Name` suchen, wählen Sie `contains`, `equals` oder `begins-with` aus und geben Sie den Parameternamen ganz oder teilweise ein.

   > **Note**
   >
   > In der Konsole ist `contains` der Standardsuchtyp für `Name`.

5. Drücken Sie Enter.

Die Liste der Parameter wird mit den Ergebnissen Ihrer Suche aktualisiert.

Suchen nach einem Parameter (AWS CLI)

Verwenden Sie den Befehl `describe-parameters`, um Informationen zu einem oder mehreren Parametern in der AWS CLI anzuzeigen.

Die folgenden Beispiele zeigen verschiedene Optionen, mit denen Sie Informationen zu den Parametern in Ihrem anzeigen können AWS-Konto. Weitere Informationen zu diesen Optionen finden Sie unter describe-parameters im AWS Command Line Interface -Handbuch.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Ersetzen Sie die Beispielwerte in den folgenden Befehlen durch Werte, die in Ihrem Konto erstellten Parametern entsprechen.

   Linux & macOS

```
aws ssm describe-parameters \
    --parameter-filters "Key=Name,Values=MyParameterName"
```

Windows

```
aws ssm describe-parameters ^
    --parameter-filters "Key=Name,Values=MyParameterName"
```

> **Note**
>
> Für `describe-parameters` ist `Equals` der Standardsuchtyp für `Name`. In den Parameterfiltern ist die Angabe von `"Key=Name,Values=MyParameterName"` identisch mit der Angabe `"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**
>
> Im letzten Beispiel steht *key-id für die ID* eines AWS Key Management Service (AWS KMS) -Schlüssels, der zur Verschlüsselung eines in Ihrem Konto erstellten

`SecureString` Parameters verwendet wird. Alternativ können Sie eingeben, **`alias/aws/ssm`** um den AWS KMS Standardschlüssel für Ihr Konto zu verwenden. Weitere Informationen finden Sie unter Erstellen eines SecureString-Parameters (AWS CLI).

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
    "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": []
        }
    ]
}
```

Zuweisen von Parameterrichtlinien

Parameterrichtlinien unterstützen Sie bei der Verwaltung einer wachsenden Menge von Parametern, indem Sie einem Parameter bestimmte Kriterien zuweisen können, wie etwa Ablaufdatum oder Time to Live (Gültigkeitsdauer). Parameterrichtlinien sind besonders hilfreich, um Sie zu zwingen, Passwörter und Konfigurationsdaten zu aktualisieren oder zu löschen, die in gespeichert sindParameter Store, eine Funktion von AWS Systems Manager. Parameter Storebietet die folgenden Arten von Richtlinien: `Expiration`Expiration`Notification`, und`NoChange`Notification`.

> **Note**
>
> Um Lebenszyklen für die Passwortrotation zu implementieren, verwenden Sie. AWS Secrets Manager Sie können Datenbankanmeldeinformationen, API-Schlüssel und andere geheime Informationen mit Secrets Manager während ihres gesamten Lebenszyklus mühelos rotieren, verwalten und abfragen. Weitere Informationen finden Sie unter Was ist? AWS Secrets Manager im AWS Secrets Manager Benutzerhandbuch.

Parameter Store erzwingt Parameterrichtlinien durch asynchrone, periodische Scans. Nachdem Sie eine Richtlinie erstellt haben, müssen Sie weitere Aktionen ausführen, um die Richtlinie zu erzwingen. Parameter Store führt die von der Richtlinie definierte Aktion gemäß den von Ihnen angegebenen Kriterien unabhängig aus.

> **Note**
>
> Parameterrichtlinien sind nur verfügbar für Parameter, die das Kontingent für erweiterte Parameter verwenden. Weitere Informationen finden Sie unter Verwalten von Parameterstufen.

Eine Parameterrichtlinie ist ein JSON-Array, wie in der folgenden Tabelle gezeigt. Sie können eine Richtlinie zuweisen, wenn Sie einen neuen erweiterten Parameter erstellen, oder Sie können eine Richtlinie anwenden, indem Sie einen Parameter aktualisieren. Parameter Store unterstützt die folgenden Arten von Parameterrichtlinien.

Richtlinie	Details	Beispiele
Ablauf	Diese Richtlinie löscht den Parameter. Sie können ein bestimmtes Datum und eine bestimmte Uhrzeit im Format `ISO_INSTANT` oder `ISO_OFFSET_DATE_TIME` angeben. Wenn Sie den Zeitpunkt für das Löschen des Parameters ändern möchten, aktualisieren Sie die Richtlinie. Das Aktualisieren eines Parameters hat keine Auswirkungen auf das Ablaufdatum oder die Uhrzeit der angefügten Richtlinie. Wenn das Ablaufdatum und die Uhrzeit erreicht ist, löscht Parameter Store den Parameter. **Note** Für das Beispiel wird das Format `ISO_INSTANT` verwendet. Sie können auch ein Datum und eine Uhrzeit im Format `ISO_OFFSET_DATE_TIME` angeben. Hier ist ein Beispiel: `2019-11-0`	`{ "Type": "Expiration", "Version": "1.0", "Attributes": { "Timestamp": "2018-12-02T21:34:33.000Z" } }`

Richtlinie	Details	Beispiele
	1T22:13:4 8.87+10:30:00 .	
ExpirationNotification	Diese Richtlinie löst ein Ereignis in Amazon EventBridge (EventBridge) aus, das Sie über den Ablauf informiert. Mithilfe dieser Richtlinie können Sie Benachrichtigungen erhalten, bevor die Ablaufzeit erreicht ist, und zwar in Einheiten von Tagen oder Stunden.	<pre>{ "Type": "ExpirationNotification", "Version": "1.0", "Attributes": { "Before": "15", "Unit": "Days" } }</pre>

Richtlinie	Details	Beispiele
NoChangeNotification	Diese Richtlinie löst ein Ereignis aus, EventBridge wenn ein Parameter für einen bestimmten Zeitraum nicht geändert wurde. Dieser Richtlinientyp ist beispielsweise nützlich, wenn ein Passwort in einem bestimmten Zeitraum geändert werden muss. Diese Richtlinie bestimmt anhand des `LastModifiedTime` -Attributs des Parameters, wann eine Benachrichtigung gesendet wird. Wenn Sie einen Parameter ändern oder bearbeiten, setzt das System den Benachrichtigungszeitraum basierend auf dem neuen Wert für `LastModifiedTime` zurück.	<pre>{ "Type": "NoChangeNotification", "Version": "1.0", "Attributes": { "After": "20", "Unit": "Days" } }</pre>

Sie können einem Parameter mehrere Richtlinien zuweisen. Sie können beispielsweise `ExpirationNotification` Richtlinien zuweisen`Expiration`, sodass das System ein EventBridge Ereignis auslöst, um Sie über das bevorstehende Löschen eines Parameters zu informieren. Sie können einem Parameter maximal zehn (10) Richtlinien zuweisen.

Das folgende Beispiel zeigt die Anforderungssyntax für eine PutParameterAPI-Anfrage, die einem neuen `SecureString` Parameter mit dem Namen vier Richtlinien zuweist. `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"
            }
        }
    ]
}
```

Hinzufügen von Richtlinien zu einem vorhandenen Parameter

Dieser Abschnitt enthält Informationen zum Hinzufügen von Richtlinien zu einem vorhandenen Parameter mithilfe der AWS Systems Manager Konsole, der AWS Command Line Interface (AWS

CLI) und AWS Tools for Windows PowerShell . Weitere Informationen zum Erstellen eines neuen Parameters mit Richtlinien finden Sie unter Erstellen von Systems Manager-Parametern.

Themen

- Hinzufügen von Richtlinien zu einem vorhandenen Parameter (Konsole)
- Hinzufügen von Richtlinien zu einem vorhandenen Parameter (AWS CLI)
- Fügen Sie Richtlinien zu einem vorhandenen Parameter hinzu (Tools für Windows PowerShell)

Hinzufügen von Richtlinien zu einem vorhandenen Parameter (Konsole)

Gehen Sie wie folgt vor, um Richtlinien zu einem vorhandenen Parameter über die Systems Manager-Konsole hinzufügen.

So fügen Sie einem vorhandenen Parameter Richtlinien hinzu

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie die Option neben dem Parameter, den Sie aktualisieren möchten, um Richtlinien einzuschließen, und klicken Sie anschließend auf Edit (Bearbeiten).
4. Wählen Sie Advanced (Erweitert) aus.
5. (Optional) Wählen Sie im Abschnitt Parameter policies (Parameterrichtlinien) die Option Enabled (Aktiviert) aus. Sie können ein Ablaufdatum und ein oder mehrere Benachrichtigungsrichtlinien für diesen Parameter angeben.
6. Wählen Sie Änderungen speichern aus.

> **Important**
>
> - Parameter Store behält Richtlinien für einen Parameter bei, bis Sie entweder die Richtlinien mit neuen Richtlinien überschreiben oder die Richtlinien entfernen.
> - Um alle Richtlinien aus einem vorhandenen Parameter zu entfernen, bearbeiten Sie den Parameter und wenden Sie eine leere Richtlinie mithilfe von eckigen und geschweiften Klammern wie folgt an: `[{}]`
> - Wenn Sie einem Parameter mit Richtlinien eine neue Richtlinie hinzufügen, überschreibt Systems Manager die dem Parameter angefügten Richtlinien. Die vorhandenen Richtlinien

werden gelöscht. Wenn Sie einem Parameter mit einer oder mehrere Richtlinien eine neue Richtlinie hinzufügen möchten, müssen Sie die ursprünglichen Richtlinien kopieren und einfügen, die neue Richtlinie eingeben und Ihre Änderungen speichern.

Hinzufügen von Richtlinien zu einem vorhandenen Parameter (AWS CLI)

Gehen Sie wie folgt vor, um einem vorhandenen Parameter mit der AWS CLI Richtlinien hinzuzufügen.

So fügen Sie einem vorhandenen Parameter Richtlinien hinzu

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl zum Hinzufügen von Richtlinien zu einem vorhandenen Parameter aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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}]"
   ```

Hier sehen Sie ein Beispiel mit einer Ablaufrichtlinie, mit der der Parameter nach 15 Tagen gelöscht wird. Das Beispiel enthält auch eine Benachrichtigungsrichtlinie, die fünf (5) Tage vor dem Löschen des Parameters ein EventBridge Ereignis generiert. Außerdem umfasst es eine `NoChangeNotification`-Richtlinie für den Fall, dass an diesem Parameter nach 60 Tagen keine Änderungen vorgenommen werden. Im folgenden Beispiel wird ein verschleierter Name (`3l3vat3131`) für ein Passwort und einen AWS Key Management Service (AWS KMS key) verwendet. Weitere Informationen zu AWS KMS keys finden Sie unter AWS Key Management Service Konzepte im AWS Key Management Service Entwicklerhandbuch.

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. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen. Ersetzen Sie *`Parametername`* durch Ihre eigenen Informationen.

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

- Parameter Store behält Richtlinien für einen Parameter bei, bis Sie entweder die Richtlinien mit neuen Richtlinien überschreiben oder die Richtlinien entfernen.
- Um alle Richtlinien aus einem vorhandenen Parameter zu entfernen, bearbeiten Sie den Parameter und wenden Sie eine leere Richtlinie mithilfe von eckigen und geschweiften Klammern an. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Zum Beispiel:

  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 "[{}]"
  ```

- Wenn Sie einem Parameter mit Richtlinien eine neue Richtlinie hinzufügen, überschreibt Systems Manager die dem Parameter angefügten Richtlinien. Die vorhandenen Richtlinien werden gelöscht. Wenn Sie einem Parameter mit einer oder mehrere Richtlinien eine neue

Richtlinie hinzufügen möchten, müssen Sie die ursprünglichen Richtlinien kopieren und einfügen, die neue Richtlinie eingeben und Ihre Änderungen speichern.

Fügen Sie Richtlinien zu einem vorhandenen Parameter hinzu (Tools für Windows PowerShell)

Gehen Sie wie folgt vor, um mithilfe von Tools für Windows einem vorhandenen Parameter Richtlinien hinzuzufügen PowerShell. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

So fügen Sie einem vorhandenen Parameter Richtlinien hinzu

1. Öffnen Sie Tools für Windows PowerShell und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen anzugeben. Sie müssen entweder über Administratorrechte in Amazon Elastic Compute Cloud (Amazon EC2) verfügen oder Ihnen müssen die entsprechenden Berechtigungen in AWS Identity and Access Management (IAM) erteilt worden sein.

   ```
   Set-AWSCredentials `
       -AccessKey access-key-name `
       -SecretKey secret-key-name
   ```

2. Führen Sie den folgenden Befehl aus, um die Region für Ihre PowerShell Sitzung festzulegen. Im Beispiel wird die Region USA Ost (Ohio) (us-east-2) verwendet.

   ```
   Set-DefaultAWSRegion `
       -Region us-east-2
   ```

3. Führen Sie den folgenden Befehl zum Hinzufügen von Richtlinien zu einem vorhandenen Parameter aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   ```
   Write-SSMParameter `
       -Name "parameter name" `
       -Value "parameter value" `
       -Type "parameter type" `
       -Policies "[{policies-enclosed-in-brackets-and-curly-braces}]" `
       -Overwrite
   ```

   Hier sehen Sie ein Beispiel mit einer Ablaufrichtlinie, mit der der Parameter am 13. Mai 2020 um Mitternacht (GMT) gelöscht wird. Das Beispiel enthält auch eine Benachrichtigungsrichtlinie,

die fünf (5) Tage vor dem Löschen des Parameters ein EventBridge Ereignis generiert. Außerdem umfasst es eine `NoChangeNotification`-Richtlinie für den Fall, dass an diesem Parameter nach 60 Tagen keine Änderungen vorgenommen werden. Im folgenden Beispiel wird ein verschleierter Name (`3l3vat3131`) für einen Passwortparameter und einen Von AWS verwalteter Schlüssel verwendet.

```
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. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen. Ersetzen Sie *Parametername* durch Ihre eigenen Informationen.

```
(Get-SSMParameterValue -Name "parameter name").Parameters
```

Important

- Parameter Store behält Richtlinien für einen Parameter bei, bis Sie entweder die Richtlinien mit neuen Richtlinien überschreiben oder die Richtlinien entfernen.
- Um alle Richtlinien aus einem vorhandenen Parameter zu entfernen, bearbeiten Sie den Parameter und wenden Sie eine leere Richtlinie mithilfe von eckigen und geschweiften Klammern an. Zum Beispiel:

  ```
  Write-SSMParameter `
      -Name "parameter name" `
      -Value "parameter value" `
      -Type "parameter type" `
      -Policies "[{}]"
  ```

- Wenn Sie einem Parameter mit Richtlinien eine neue Richtlinie hinzufügen, überschreibt Systems Manager die dem Parameter angefügten Richtlinien. Die vorhandenen Richtlinien

werden gelöscht. Wenn Sie einem Parameter mit einer oder mehrere Richtlinien eine neue Richtlinie hinzufügen möchten, müssen Sie die ursprünglichen Richtlinien kopieren und einfügen, die neue Richtlinie eingeben und Ihre Änderungen speichern.

Arbeiten mit Parameterhierarchien

Das Verwalten Dutzender oder Hunderter Parameter als unsortierte Liste ist zeitaufwendig und fehleranfällig. Außerdem kann es sich als schwierig erweisen, für eine bestimmte Aufgabe den korrekten Parameter zu bestimmen. Sie könnten versehentlich den falschen Parameter verwenden, oder Sie erstellen möglicherweise mehrere Parameter, die dieselben Konfigurationsdaten verwenden.

Mit Parameterhierarchien können Sie -Parameter leichter organisieren und verwalten. Bei einer Hierarchie handelt es sich um einen Parameternamen mit einem Pfad, den Sie mit Schrägstrichen (/) definieren.

Themen

- Beispiele für Parameterhierarchien
- Abfragen von Parametern in einer Hierarchie
- Einschränken des Zugriffs auf Parameter Store-API-Operationen
- Verwalten von Parametern mithilfe von Hierarchien (AWS CLI)

Beispiele für Parameterhierarchien

Im folgendne Beispiel werden drei Hierarchieebenen im Namen verwendet. Damit wird Folgendes identifiziert:

```
/Environment/Type of computer/Application/Data
```

```
/Dev/DBServer/MySQL/db-string13
```

Sie können eine Hierarchie mit maximal 15 Ebenen erstellen. Wir empfehlen, dass Sie Hierarchien erstellen, die eine vorhandene hierarchische Struktur in Ihrer Umgebung abbilden, wie in den folgenden Beispielen gezeigt:

- Ihre Umgebung für kontinuierliche Integration und kontinuierliche Bereitstellung (CI/CD-Workflows)

  ```
  /Dev/DBServer/MySQL/db-string
  ```

```
/Staging/DBServer/MySQL/db-string

/Prod/DBServer/MySQL/db-string
```

- Die Anwendungen, die Container verwenden

```
/MyApp/.NET/Libraries/my-password
```

- Die Unternehmensstruktur

```
/Finance/Accountants/UserList

/Finance/Analysts/UserList

/HR/Employees/EU/UserList
```

Parameterhierarchien standardisieren die Möglichkeiten für die Erstellung von Parameter und vereinfachen mit der Zeit die Verwaltung von Parametern. Eine Parameterhierarchie kann außerdem dazu beitragen, den richtigen Parameter für eine Konfigurationsaufgabe zu bestimmen. Auf diese Weise können Sie vermeiden, dass mehrere Parameter mit denselben Konfigurationsdaten erstellt werden.

Sie können eine Hierarchie erstellen, mit der Sie wie in den folgenden Beispielen gezeigt Parameter über verschiedene Umgebungen hinweg freigeben können; hier werden Passwörter in Entwicklungs- und Staging-Umgebungen verwendet.

```
/DevTest/MyApp/database/my-password
```

Sie könnten anschließend ein eindeutiges Passwort für Ihre produktive Umgebung erstellen, wie im folgenden Beispiel gezeigt:

```
/prod/MyApp/database/my-password
```

Sie müssen dabei nicht unbedingt eine Parameterhierarchie angeben. Sie können Parameter auf Ebene 1 erstellen. Diese werden als Root-Parameter bezeichnet. Um die Abwärtskompatibilität zu gewährleisten, sind alle Parameter, die in Parameter Store erstellt wurden, bevor Hierarchien eingeführt wurden, Root-Parameter. Die Systeme behandelt die folgenden beiden Parameter als Root-Parameter.

```
/parameter-name
```

`parameter-name`

Abfragen von Parametern in einer Hierarchie

Ein weiterer Vorteil der Verwendung von Hierarchien ist die Möglichkeit zur Abfrage aller Parameter innerhalb einer Hierarchie mithilfe der API-Operation GetParametersByPath. Wenn Sie beispielsweise den folgenden Befehl in der AWS Command Line Interface (AWS CLI) ausführen, gibt das System alle Parameter auf der IIS-Ebene zurück.

```
aws ssm get-parameters-by-path --path /Dev/Web/IIS
```

Sie können den entschlüsselten `SecureString`-Parameter in einer Hierarchie anzeigen, indem Sie den Pfad und den `--with-decryption`-Parameter angeben, wie im folgenden Beispiel gezeigt.

```
aws ssm get-parameters-by-path --path /Prod/ERP/SAP --with-decryption
```

Einschränken des Zugriffs auf Parameter Store-API-Operationen

Sie können mit AWS Identity and Access Management (IAM)-Richtlinien den Benutzerzugriff auf Parameter Store-API-Operationen und -Inhalte bereitstellen oder einschränken.

In der folgenden Beispielrichtlinie wird Benutzern zunächst Zugriff gewährt, um die API-Aktion `PutParameter` für alle Parameter im AWS-Konto 123456789012 in der Region USA Ost (Ohio) (us-east-2) auszuführen. Anschließend wird jedoch verhindert, dass Benutzer die Werte vorhandener Parameter ändern, da die Option `0verwrite` für die Operation `PutParameter` ausdrücklich abgelehnt wird. Benutzer, denen diese Richtlinie zugewiesen ist, können daher Parameter erstellen, vorhandene Parameter jedoch nicht ändern.

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

Verwalten von Parametern mithilfe von Hierarchien (AWS CLI)

In dieser Anleitung wird beschrieben, wie Sie mit Parametern und Parameterhierarchien arbeiten können, indem Sie die AWS CLI verwenden.

So verwalten Sie Parameter mithilfe von Hierarchien

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen, der den `allowedPattern`-Parameter und den Parametertyp `String` verwendet. Das zulässige Muster in diesem Beispiel bedeutet, dass der Wert für den Parameter zwischen 1 und 4 Zeichen lang sein muss.

   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
```

Der Befehl gibt die Versionsnummer des Parameters zurück.

3. Führen Sie den folgenden Befehl aus und versuchen Sie, den gerade erstellten Parameter mit einem neuen Wert zu überschreiben.

   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
   ```

   Das System gibt den folgenden Fehler zurück, da der neue Wert die Anforderungen des zulässigen Musters nicht erfüllt, das Sie im letzten Schritt angegeben haben.

   ```
   An error occurred (ParameterPatternMismatchException) when calling the PutParameter
    operation: Parameter value, cannot be validated against allowedPattern: \d{1,4}
   ```

4. Führen Sie den folgenden Befehl aus, um einen `SecureString`-Parameter zu erstellen, der den Datentyp Von AWS verwalteter Schlüssel verwendet. Das zulässige Muster in diesem Beispiel bedeutet, dass der Benutzer beliebige Zeichen eingeben kann, und der Wert zwischen 8 und 20 Zeichen lang sein muss.

   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. Führen Sie die folgenden Befehle aus, um mehrere Parameter zu erstellen, die die Hierarchiestruktur aus dem letzten Schritt verwenden.

   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. Führen Sie den folgenden Befehl aus, um den Wert zweier Parameter abzurufen.

   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. Führen Sie den folgenden Befehl aus, um alle Parameter auf einer bestimmten Ebene abzufragen.

   Linux & macOS

   ```
   aws ssm get-parameters-by-path \
       --path "/MyService/Test"
   ```

   Windows

   ```
   aws ssm get-parameters-by-path ^
       --path "/MyService/Test"
   ```

8. Führen Sie den folgenden Befehl aus, um zwei Parameter zu löschen.

   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"
```

Arbeiten mit Parameterbezeichnungen

Eine Parameter-Bezeichnung ist ein benutzerdefinierter Alias, mit dem Sie verschiedene Versionen eines Parameters verwalten können. Wenn Sie einen Parameter ändern, AWS Systems Manager wird automatisch eine neue Version gespeichert und die Versionsnummer um eins erhöht. Dank einer Bezeichnung können Sie sich den Zweck einer Parameterversion merken, wenn mehrere Versionen vorhanden sind.

Nehmen wir beispielsweise an, Sie haben einen Parameter mit dem Namen `/MyApp/DB/ConnectionString`. Der Wert des Parameters ist eine Verbindungszeichenfolge mit einem MySQL-Server in einer lokalen Datenbank einer Testumgebung. Nachdem Sie die Anwendung aktualisiert haben, möchten Sie festlegen, dass der Parameter eine Verbindungszeichenfolge für eine Produktionsdatenbank verwendet. Sie ändern den Wert von `/MyApp/DB/ConnectionString`. Systems Manager erstellt automatisch Version 2 mit der neuen Verbindungszeichenfolge. Damit Sie sich den Zweck der einzelnen Versionen besser merken können, fügen Sie jedem Parameter eine Bezeichnung an. Für Version eins fügen Sie die Bezeichnung Test an. Für Version zwei fügen Sie die Bezeichnung Production an.

Sie können Bezeichnungen von einer Version eines Parameters in eine andere Version verschieben. Wenn Sie beispielsweise Version drei des Parameters `/MyApp/DB/ConnectionString` mit einer Verbindungszeichenfolge für eine neue Produktionsdatenbank erstellen, können Sie die Bezeichnung Production von Version zwei des Parameters zu Version drei des Parameters verschieben.

Parameterbezeichnungen stellen eine einfache Alternative zu Parameter-Tags dar. Ihre Organisation verfügt u. U. über strenge Richtlinien für Tags, die auf verschiedene AWS -Ressourcen angewendet werden müssen. Im Gegensatz dazu ist eine Bezeichnung einfach eine Textzuordnung für eine bestimmte Version eines Parameters.

Ähnlich wie Tags können Sie Parameter mithilfe von Bezeichnungen abfragen. Sie können eine Liste bestimmter Parameterversionen anzeigen, die alle dieselbe Bezeichnung verwenden, wenn Sie Ihren

Parametersatz mithilfe der GetParametersByPathAPI-Operation abfragen, wie weiter unten in diesem Abschnitt beschrieben.

> **Note**
>
> Wenn Sie einen Befehl ausführen, der eine Version eines Parameters angibt, die nicht existiert, schlägt der Befehl fehl. Es greift nicht auf den letzten oder Standardwert des Parameters zurück.

Anforderungen und Einschränkungen für Bezeichnungen

Für Parameterbezeichnungen gelten die folgenden Anforderungen und Einschränkungen:

- Für eine Version eines Parameters sind maximal 10 Bezeichnungen zulässig.
- Es ist nicht möglich, die gleiche Bezeichnung verschiedenen Versionen desselben Parameters anzufügen. Wenn Version 1 des Parameters beispielsweise die Bezeichnung Production hat, können Sie Production nicht an Version 2 anfügen.
- Sie können eine Bezeichnung von einer Version eines Parameters zu einer anderen Version verschieben.
- Es ist nicht möglich, eine Bezeichnung festzulegen, wenn Sie einen Parameter erstellen. Sie müssen eine Bezeichnung einer bestimmten Version eines Parameters anfügen.
- Wenn Sie eine Parameterbezeichnung nicht mehr verwenden möchten, können Sie sie zu einer anderen Version eines Parameters verschieben oder sie löschen.
- Eine Bezeichnung darf höchstens 100 Zeichen lang sein.
- Bezeichnungen können Buchstaben (Unterscheidung nach Groß- und Kleinschreibung), Ziffern, Punkte (.), Bindestriche (-) und Unterstriche (_) enthalten.
- Bezeichnungen dürfen nicht mit einer Zahl, "aws" oder "ssm" (keine Unterscheidung nach Groß- und Kleinschreibung) beginnen. Wenn eine Bezeichnung diese Anforderungen nicht erfüllt, wird sie der Parameterversion nicht angefügt und vom System in der Liste `InvalidLabels` angezeigt.

Themen

- Arbeiten mit Parameterbezeichnungen (Konsole)
- Arbeiten mit Parameterbezeichnungen (AWS CLI)

Arbeiten mit Parameterbezeichnungen (Konsole)

In diesem Abschnitt wird beschrieben, wie Sie die folgenden Aufgaben mithilfe der Systems Manager-Konsole durchführen.

- Erstellen einer Parameterbeschriftung (Konsole)
- Anzeigen von Bezeichnungen, die einem Parameter angefügt sind (Konsole)
- Verschieben einer Parameterbezeichnung (Konsole)
- Löschen von Parameterbezeichnungen (Konsole)

Erstellen einer Parameterbeschriftung (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie einer bestimmten Version eines vorhandenen Parameters über die Systems Manager-Konsole eine Bezeichnung anfügen. Es ist nicht möglich, eine Bezeichnung anzufügen, wenn Sie einen neuen Parameter erstellen.

Anfügen einer Bezeichnung an die aktuelle Version eines Parameters

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, um die Detailseite für diesen Parameter anzuzeigen.
4. Wählen Sie die Registerkarte History (Verlauf) aus.
5. Wählen Sie die Parameterversion aus, der Sie eine Bezeichnung anfügen möchten.
6. Klicken Sie auf Verwalten von Bezeichnungen.
7. Klicken Sie auf Hinzufügen einer neuen Bezeichnung.
8. Geben Sie den Namen der Bezeichnung in das Textfeld ein. Wählen Sie Add new label (Neue Bezeichnung hinzufügen) aus, um weitere Bezeichnungen hinzuzufügen. Sie können maximal zehn Bezeichnungen anfügen.
9. Wenn Sie fertig sind, wählen Sie Änderungen speichern aus.

Anzeigen von Bezeichnungen, die einem Parameter angefügt sind (Konsole)

Für eine Parameterversion sind maximal 10 Bezeichnungen zulässig. Im folgenden Verfahren wird beschrieben, wie Sie alle Bezeichnungen, die einer Parameterversion angefügt sind, mithilfe der Systems Manager-Konsole anzeigen.

Anzeigen von Bezeichnungen, die einer Parameterversion angefügt sind

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, um die Detailseite für diesen Parameter anzuzeigen.
4. Wählen Sie die Registerkarte History (Verlauf) aus.
5. Suchen Sie die Parameterversion, für die Sie die angefügten Bezeichnungen anzeigen möchten. Die Spalte Labels (Bezeichnungen) enthält alle Bezeichnungen, die der Parameterversion angefügt sind.

Verschieben einer Parameterbezeichnung (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie eine Parameterbezeichnung zu einer anderen Version desselben Parameters mithilfe der Systems Manager-Konsole verschieben.

So verschieben Sie eine Bezeichnung zu einer Parameterversion

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, um die Detailseite für diesen Parameter anzuzeigen.
4. Wählen Sie die Registerkarte History (Verlauf) aus.
5. Wählen Sie die Parameterversion aus, deren Bezeichnung Sie verschieben möchten.
6. Klicken Sie auf Verwalten von Bezeichnungen.
7. Klicken Sie auf Hinzufügen einer neuen Bezeichnung.
8. Geben Sie den Namen der Bezeichnung in das Textfeld ein.
9. Wenn Sie fertig sind, wählen Sie Änderungen speichern aus.

Löschen von Parameterbezeichnungen (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie über die Systems Manager-Konsole eine oder mehrere Parameterbezeichnungen löschen.

So löschen Sie Bezeichnungen aus einem Parameter

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, um die Detailseite für diesen Parameter anzuzeigen.
4. Wählen Sie die Registerkarte History (Verlauf) aus.
5. Wählen Sie die Parameterversion aus, deren Bezeichnungen Sie löschen möchten.
6. Klicken Sie auf Verwalten von Bezeichnungen.
7. Klicken Sie neben jeder Bezeichnung, die Sie löschen möchten, auf Remove (Entfernen).
8. Wenn Sie fertig sind, wählen Sie Änderungen speichern aus.
9. Bestätigen Sie, dass Ihre Änderungen korrekt sind, geben Sie `Confirm` in das Textfeld ein und wählen Sie Bestätigen aus.

Arbeiten mit Parameterbezeichnungen (AWS CLI)

In diesem Abschnitt wird beschrieben, wie Sie die folgenden Aufgaben mithilfe der AWS Command Line Interface (AWS CLI) durchführen.

- Erstellen einer neuen Parameterbezeichnung (AWS CLI)
- Anzeigen der Bezeichnungen für einen Parameter (AWS CLI)
- Anzeigen einer Liste von Parametern, denen eine Bezeichnung zugewiesen ist (AWS CLI)
- Verschieben einer Parameterbezeichnung (AWS CLI)
- Löschen von Parameterbezeichnungen (AWS CLI)

Erstellen einer neuen Parameterbezeichnung (AWS CLI)

Im folgenden Verfahren wird beschrieben, wie Sie einer bestimmten Version eines vorhandenen Parameters über die AWS CLI eine Bezeichnung anfügen. Es ist nicht möglich, eine Bezeichnung anzufügen, wenn Sie einen neuen Parameter erstellen.

So erstellen Sie eine neue Parameterbezeichnung

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um eine Liste der Parameter anzuzeigen, für die Sie über die Berechtigung zum Anfügen einer Bezeichnung verfügen.

   > **Note**
   >
   > Parameter sind nur dort verfügbar AWS-Region , wo sie erstellt wurden. Falls Sie einen Parameter, dem Sie eine Bezeichnung anfügen möchten, nicht finden können, prüfen Sie Ihre Region.

   ```
   aws ssm describe-parameters
   ```

   Notieren Sie den Namen eines Parameters, dem Sie eine Bezeichnung anfügen möchten.

3. Führen Sie den folgenden Befehl aus, um alle Versionen des Parameters anzuzeigen.

   ```
   aws ssm get-parameter-history --name "parameter-name"
   ```

   Notieren Sie die Parameterversion, der Sie eine Bezeichnung anfügen möchten.

4. Führen Sie den folgenden Befehl aus, um anhand der Versionsnummer Informationen zu einem Parameter abzurufen.

   ```
   aws ssm get-parameters --names "parameter-name:version-number"
   ```

   Ein Beispiel.

   ```
   aws ssm get-parameters --names "/Production/SQLConnectionString:3"
   ```

5. Führen Sie einen der folgenden Befehle aus, um einer Parameterversion eine Bezeichnung anzufügen. Wenn Sie mehrere Bezeichnungen anzufügen, müssen Sie die Namen der Bezeichnungen durch ein Leerzeichen trennen.

Anfügen einer Bezeichnung an die aktuelle Version eines Parameters

```
aws ssm label-parameter-version --name parameter-name  --labels label-name
```

Anfügen einer Bezeichnung an eine bestimmte Version eines Parameters

```
aws ssm label-parameter-version --name parameter-name --parameter-version version-number --labels label-name
```

Hier sind einige Beispiele.

```
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**
>
> Wenn die Ausgabe die Bezeichnung zeigt, die Sie in der Liste `InvalidLabels` erstellt haben, entspricht die Bezeichnung nicht den weiter oben in diesem Thema beschriebenen Anforderungen. Überprüfen Sie die Anforderungen und versuchen Sie es erneut. Wenn die Liste `InvalidLabels` leer ist, wurde Ihre Bezeichnung der Version des Parameters erfolgreich angefügt.

6. Sie können die Details des Parameters entweder mithilfe einer Versionsnummer oder eines Bezeichnungsnamens anzeigen. Führen Sie den folgenden Befehl aus und geben Sie die im vorherigen Schritt erstellte Bezeichnung an.

```
aws ssm get-parameter --name parameter-name:label-name --with-decryption
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "Parameter": {
        "Version": version-number,
        "Type": "parameter-type",
```

```
        "Name": "parameter-name",
        "Value": "parameter-value",
        "Selector": ":label-name"
    }
}
```

> **Note**
>
> Selector (Auswahl) in der Ausgabe ist entweder die Versionsnummer oder die Bezeichnung, die Sie im Eingabefeld Name angegeben haben.

Anzeigen der Bezeichnungen für einen Parameter (AWS CLI)

Sie können den GetParameterHistoryAPI-Vorgang verwenden, um den vollständigen Verlauf und alle mit einem bestimmten Parameter verknüpften Labels anzuzeigen. Oder Sie können den GetParametersByPathAPI-Vorgang verwenden, um eine Liste aller Parameter anzuzeigen, denen ein bestimmtes Label zugewiesen wurde.

Um Beschriftungen für einen Parameter mithilfe der GetParameterHistory API-Operation anzuzeigen

1. Führen Sie den folgenden Befehl aus, um eine Liste der Parameter anzuzeigen, für die Sie Bezeichnungen anzeigen können.

   > **Note**
   >
   > Parameter sind nur in den Regionen verfügbar, in denen sie erstellt wurden. Falls Sie einen Parameter, für den Sie eine Bezeichnung verschieben möchten, nicht finden können, prüfen Sie Ihre Region.

   ```
   aws ssm describe-parameters
   ```

   Notieren Sie sich den Namen des Parameters, dessen Bezeichnungen Sie anzeigen möchten.

2. Führen Sie den folgenden Befehl aus, um alle Versionen des Parameters anzuzeigen.

   ```
   aws ssm get-parameter-history --name parameter-name --with-decryption
   ```

Das System gibt unter anderem folgende Informationen zurück

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

Anzeigen einer Liste von Parametern, denen eine Bezeichnung zugewiesen ist (AWS CLI)

Sie können den GetParametersByPathAPI-Vorgang verwenden, um eine Liste aller Parameter in einem Pfad anzuzeigen, denen eine bestimmte Bezeichnung zugewiesen wurde.

Führen Sie den folgenden Befehl aus, um eine Liste der Parameter in einem Pfad anzuzeigen, denen eine bestimmte Bezeichnung zugeordnet wurde. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
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
```

Das System gibt unter anderem folgende Informationen zurück In diesem Beispiel durchsuchte der Benutzer den `/Config`-Pfad.

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

Verschieben einer Parameterbezeichnung (AWS CLI)

Im folgenden Verfahren wird beschrieben, wie Sie eine Parameterbezeichnung zu einer anderen Version desselben Parameters verschieben.

So verschieben Sie eine Parameterbezeichnung

1. Führen Sie den folgenden Befehl aus, um alle Versionen des Parameters anzuzeigen. Ersetzen Sie *Parametername* durch Ihre eigenen Informationen.

   ```
   aws ssm get-parameter-history \
       --name "parameter name"
   ```

Beachten Sie die Parameterversionen, aus denen Sie die Bezeichnung verschieben möchten.

2. Führen Sie den folgenden Befehl aus, um eine vorhandene Bezeichnung einer anderen Version eines Parameters zuzuweisen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws ssm label-parameter-version \
    --name parameter name \
    --parameter-version version number \
    --labels name-of-existing-label
```

> **Note**
>
> Wenn Sie eine vorhandene Bezeichnung zur neuesten Version eines Parameters verschieben möchten, entfernen Sie `--parameter-version` aus dem Befehl.

Löschen von Parameterbezeichnungen (AWS CLI)

Im folgenden Verfahren wird beschrieben, wie Sie Parameterbezeichnungen mithilfe der AWS CLI löschen.

So löschen Sie eine Parameterbezeichnung

1. Führen Sie den folgenden Befehl aus, um alle Versionen des Parameters anzuzeigen. Ersetzen Sie *Parametername* durch Ihre eigenen Informationen.

```
aws ssm get-parameter-history \
    --name "parameter name"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Parameters": [
        {
            "Name": "foo",
            "DataType": "text",
            "LastModifiedDate": 1607380761.11,
            "Labels": [
                "l3",
```

```
                "12"
            ],
            "Value": "test",
            "Version": 1,
            "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
            "Policies": [],
            "Tier": "Standard",
            "Type": "String"
        },
        {
            "Name": "foo",
            "DataType": "text",
            "LastModifiedDate": 1607380763.11,
            "Labels": [
                "11"
            ],
            "Value": "test",
            "Version": 2,
            "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
            "Policies": [],
            "Tier": "Standard",
            "Type": "String"
        }
    ]
}
```

Notieren Sie die Parameterversion, für die Sie eine oder mehrere Bezeichnungen löschen möchten.

2. Führen Sie den folgenden Befehl aus, um die Bezeichnungen zu löschen, die Sie aus diesem Parameter auswählen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws ssm unlabel-parameter-version \
    --name parameter name \
    --parameter-version version \
    --labels label 1,label 2,label 3
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "InvalidLabels": ["invalid"],
```

```
    "DeletedLabels" : ["Prod"]
}
```

Arbeiten mit Parameterversionen

Jedes Mal, wenn Sie den Wert eines Parameters bearbeiten, erstellt Parameter Store, eine Funktion von AWS Systems Manager , eine neue Version des Parameters und behält die vorherigen Versionen bei. Zu Beginn der Erstellung eines Parameters weist Parameter Store diesem Parameter die Version 1 zu. Wenn Sie den Wert des Parameters ändern, erhöht Parameter Store die Versionsnummer automatisch um eins. Sie können die Details, einschließlich der Werte, aller Versionen im Verlauf eines Parameters anzeigen.

Sie können auch die Version eines Parameters angeben, der in API-Befehlen und SSM-Dokumenten verwendet werden soll. Beispiel: `ssm:MyParameter:3`. Sie können einen Parameternamen und eine bestimmte Versionsnummer in API-Aufrufen und SSM-Dokumenten angeben. Wenn Sie keine Versionsnummer angeben, verwendet das System automatisch die neueste Version. Wenn Sie die Nummer für eine nicht vorhandene Version angeben, gibt das System einen Fehler zurück, anstatt auf die neueste oder Standardversion des Parameters zurückzugreifen.

Sie können Parameterversionen verwenden, um zu sehen, wie oft ein Parameter im Lauf eines bestimmten Zeitraums geändert wurde. Parameterversionen bieten auch eine Schutzebene, wenn ein Parameterwert versehentlich geändert wird.

Sie können maximal 100 Versionen eines Parameters erstellen verwalten. Nachdem Sie 100 Versionen eines Parameters erstellt haben, wird jedes Mal, wenn Sie eine neue Version erstellen, die älteste Version des Parameters aus dem Verlauf entfernt, um Platz für die neue Version zu schaffen.

Eine Ausnahme ist, wenn bereits 100 Parameterversionen im Verlauf vorhanden sind und der ältesten Version eines Parameters eine Parameterbezeichnung zugewiesen wird. In diesem Fall wird diese Version nicht aus dem Verlauf entfernt, und die Anforderung, eine neue Parameterversion zu erstellen, schlägt fehl. Diese Schutzmaßnahme soll verhindern, dass Parameterversionen mit ihnen zugewiesenen geschäftskritischen Bezeichnungen gelöscht werden. Um mit dem Erstellen neuer Parameter fortzufahren, verschieben Sie die Bezeichnung zuerst von der ältesten Version des Parameters in eine neuere Version, um sie in Ihren Operationen verwenden zu können. Informationen zum Verschieben von Parameterbezeichnungen finden Sie unter Verschieben einer Parameterbezeichnung (Konsole) und Verschieben einer Parameterbezeichnung (AWS CLI).

Das folgende Verfahren zeigt, wie Sie einen Parameter bearbeiten und dann überprüfen, ob Sie eine neue Version erstellt haben. Sie können die Befehle `get-parameter` und `get-parameters`

verwenden, um Parameterversionen anzuzeigen. Beispiele zur Verwendung dieser Befehle finden Sie unter GetParameterund GetParametersin der AWS Systems Manager API-Referenz

Themen

- Erstellen einer neuen Version eines Parameters (Konsole)
- Verweisen auf eine Parameterversion

Erstellen einer neuen Version eines Parameters (Konsole)

Sie können die Systems Manager-Konsole verwenden, um eine neue Version eines Parameters zu erstellen und den Versionsverlauf eines Parameters anzuzeigen.

So erstellen Sie eine neue Version eines Parameters

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, den Sie vorher erstellt haben. Weitere Informationen zum Erstellen eines neuen Parameters finden Sie unter Erstellen von Systems Manager-Parametern.
4. Wählen Sie Bearbeiten aus.
5. Geben Sie im Feld Value einen neuen Wert ein und klicken Sie auf Save changes.
6. Wählen Sie den Namen des Parameters aus, den Sie gerade aktualisiert haben. Prüfen Sie auf der Registerkarte Overview, dass die Versionsnummer um 1 erhöht wurde, und überprüfen Sie den neuen Wert.
7. Um den Verlauf aller Versionen eines Parameters anzuzeigen, wählen Sie die Registerkarte History (Verlauf) aus.

Verweisen auf eine Parameterversion

Sie können in Befehlen, API-Aufrufen und SSM-Dokumenten mithilfe des folgenden Formats auf spezifische Parameterversionen verweisen: ssm: *`parameter-name`*`:`*`version-number`*.

Im folgenden Beispiel verwendet die Amazon Elastic Compute Cloud (Amazon EC2) `run-instances command` Version 3 des Parameters `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**
>
> Das Verwenden von `resolve` und einem Parameterwert ist nur mit der Option `--image-id` und einem Parameter, der ein Amazon Machine Image (AMI) als Wert enthält, möglich. Weitere Informationen finden Sie unter Unterstützung für native Parameter für Amazon Machine Image-IDs.

Hier ist ein Beispiel für die Angabe von Version 2 eines Parameters mit dem Namen `MyRunCommandParameter` in einem SSM-Dokument.

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}}"
                ]
            }
        }
    ]
}
```

Mit gemeinsam genutzten Parametern arbeiten

Die gemeinsame Nutzung erweiterter Parameter vereinfacht die Verwaltung von Konfigurationsdaten in einer Umgebung mit mehreren Konten. Sie können Ihre Parameter zentral speichern und verwalten und sie mit anderen teilen AWS-Konten , die sie referenzieren müssen.

Parameter Storelässt sich in AWS Resource Access Manager (AWS RAM) integrieren, um die erweiterte gemeinsame Nutzung von Parametern zu ermöglichen. AWS RAM ist ein Dienst, der es Ihnen ermöglicht, Ressourcen mit anderen zu teilen AWS-Konten oder über AWS Organizations.

Mit können Sie Ressourcen AWS RAM, die Ihnen gehören, gemeinsam nutzen, indem Sie eine gemeinsame Nutzung erstellen. Eine Ressourcenfreigabe gibt an, welche Ressourcen gemeinsam genutzt werden sollen, welche Berechtigungen gewährt werden sollen und mit welchen Verbrauchern diese gemeinsam genutzt werden sollen. Zu den Verbrauchern können gehören:

- AWS-Konten Spezifisch innerhalb oder außerhalb seiner Organisation in AWS Organizations
- Eine Organisationseinheit innerhalb ihrer Organisation in AWS Organizations
- Ihre gesamte Organisation ist in AWS Organizations

Weitere Informationen zu AWS RAM finden Sie im AWS RAM Benutzerhandbuch.

In diesem Thema wird erklärt, wie Sie Parameter, deren Eigentümer Sie sind, gemeinsam nutzen und wie Sie Parameter verwenden, die mit Ihnen gemeinsam genutzt werden.

Inhalt

- Voraussetzungen für die gemeinsame Nutzung von Parametern
- Einen Parameter teilen
- Beenden Sie die gemeinsame Nutzung eines gemeinsam genutzten Parameters
- Identifizieren gemeinsam genutzter Parameter
- Zugreifen auf gemeinsam genutzte Parameter
- Berechtigungssätze für die gemeinsame Nutzung von Parametern
- Maximaler Durchsatz für gemeinsam genutzte Parameter
- Preisgestaltung für gemeinsam genutzte Parameter
- Kontoübergreifender Zugriff für geschlossene AWS-Konten

Voraussetzungen für die gemeinsame Nutzung von Parametern

Die folgenden Voraussetzungen müssen erfüllt sein, bevor Sie Parameter aus Ihrem Konto teilen können:

- Um einen Parameter gemeinsam zu nutzen, müssen Sie ihn in Ihrem besitzen AWS-Konto. Sie können einen Parameter, der mit Ihnen geteilt wurde, nicht teilen.

- Um einen Parameter gemeinsam nutzen zu können, muss er sich in der erweiterten Parameterebene befinden. Hinweise zu Parameterschichten finden Sie unterVerwalten von Parameterstufen. Hinweise zum Ändern eines vorhandenen Standardparameters in einen erweiterten Parameter finden Sie unterÄndern eines Standardparameters in einen fortgeschrittenen Parameter.
- Um einen `SecureString` Parameter gemeinsam zu nutzen, muss er mit einem vom Kunden verwalteten Schlüssel verschlüsselt werden, und Sie müssen den Schlüssel separat weitergeben AWS Key Management Service. Von AWS verwaltete Schlüssel kann nicht geteilt werden. Mit der Standardeinstellung verschlüsselte Parameter Von AWS verwalteter Schlüssel können aktualisiert werden, sodass stattdessen ein vom Kunden verwalteter Schlüssel verwendet wird. AWS KMS Schlüsseldefinitionen finden Sie unter AWS KMS Konzepte im AWS Key Management Service Entwicklerhandbuch.
- Um einen Parameter mit Ihrer Organisation oder einer Organisationseinheit gemeinsam zu nutzen AWS Organizations, müssen Sie das Teilen mit aktivieren AWS Organizations. Weitere Informationen finden Sie unter Freigabe für AWS Organizations aktivieren im AWS RAM - Benutzerhandbuch.

Einen Parameter teilen

Um einen Parameter gemeinsam zu nutzen, müssen Sie ihn zu einer Ressourcenfreigabe hinzufügen. Eine Ressourcenfreigabe ist eine AWS RAM Ressource, mit der Sie Ihre Ressourcen gemeinsam nutzen können AWS-Konten. Eine Ressourcenfreigabe gibt die freizugebenden Ressourcen und die Konsumenten an, für die sie freigegeben werden.

Wenn Sie einen Parameter, dessen Eigentümer Sie sind, mit anderen teilen AWS-Konten, können Sie zwischen zwei AWS verwalteten Berechtigungen wählen, die Sie den Benutzern gewähren möchten. Weitere Informationen finden Sie unter Berechtigungssätze für die gemeinsame Nutzung von Parametern.

Wenn Sie Teil einer Organisation sind AWS Organizations und das Teilen innerhalb Ihrer Organisation aktiviert ist, können Sie Verbrauchern in Ihrer Organisation von der AWS RAM Konsole aus Zugriff auf den gemeinsamen Parameter gewähren. Andernfalls erhalten Verbraucher eine Einladung zur Teilnahme an Resource Share und erhalten Zugriff auf den gemeinsamen Parameter, nachdem sie die Einladung angenommen haben.

Sie können einen Parameter, den Sie besitzen, mithilfe der AWS RAM Konsole oder der teilen AWS CLI.

> **Note**
>
> Sie können einen Parameter zwar mithilfe der Systems Manager PutResourcePolicy API-Operation teilen, wir empfehlen jedoch, stattdessen AWS Resource Access Manager (AWS RAM) zu verwenden. Dies liegt daran, dass für die Verwendung des Parameters der zusätzliche Schritt `PutResourcePolicy` erforderlich ist, den Parameter mithilfe der AWS RAM PromoteResourceShareCreatedFromPolicyAPI-Operation auf einen standardmäßigen Resource Share hochzustufen. Andernfalls wird der Parameter nicht von der Systems Manager DescribeParametersManager-API-Operation zurückgegeben, die die `--shared` Option verwendet.

Um einen Parameter, den Sie besitzen, mithilfe der AWS RAM Konsole gemeinsam zu nutzen

Weitere Informationen finden Sie unter Erstellen einer gemeinsamen Ressource AWS RAM im AWS RAM Benutzerhandbuch.

Treffen Sie beim Abschluss des Verfahrens die folgenden Auswahlen:

- Wählen Sie auf der Seite Schritt 1 unter Ressourcen die Option für jeden Parameter in der erweiterten Parameterebene aus`Parameter Store Advanced Parameter`, den Sie gemeinsam nutzen möchten, und aktivieren Sie dann das Kästchen.
- Wählen Sie auf der Seite Schritt 2 für Verwaltete Berechtigungen die Berechtigung aus, die Verbrauchern gewährt werden soll, wie weiter Berechtigungssätze für die gemeinsame Nutzung von Parametern unten in diesem Thema beschrieben.

Wählen Sie je nach Ihren Zielen für die gemeinsame Nutzung von Parametern weitere Optionen aus.

Um einen Parameter, den Sie besitzen, zu teilen, verwenden Sie den AWS CLI

Verwenden Sie den create-resource-shareBefehl, um Parameter zu einer neuen Ressourcenfreigabe hinzuzufügen.

Verwenden Sie den associate-resource-shareBefehl, um einer vorhandenen Ressourcenfreigabe Parameter hinzuzufügen.

Im folgenden Beispiel wird eine neue Ressourcenfreigabe erstellt, um Parameter mit Verbrauchern in einer Organisation und in einem Einzelkonto gemeinsam zu nutzen.

```
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"
```

Beenden Sie die gemeinsame Nutzung eines gemeinsam genutzten Parameters

Wenn Sie die gemeinsame Nutzung eines gemeinsam genutzten Parameters beenden, kann das Verbraucherkonto nicht mehr auf den Parameter zugreifen.

Um die gemeinsame Nutzung eines Parameters, dessen Eigentümer Sie sind, zu beenden, müssen Sie ihn aus der Ressourcenfreigabe entfernen. Hierzu können Sie die Systems Manager-Konsole, die AWS RAM -Konsole oder die AWS CLI verwenden.

Um die gemeinsame Nutzung eines Parameters, dessen Eigentümer Sie sind, über die AWS RAM Konsole zu beenden

Weitere Informationen finden Sie unter Aktualisieren einer gemeinsam genutzten Ressource AWS RAM im AWS RAM Benutzerhandbuch.

Um die gemeinsame Nutzung eines Parameters, dessen Eigentümer Sie sind, zu beenden, verwenden Sie AWS CLI

Verwenden Sie den Befehl disassociate-resource-share.

Identifizieren gemeinsam genutzter Parameter

Eigentümer und Verbraucher können gemeinsam genutzte Parameter anhand der identifizieren AWS CLI.

Um gemeinsam genutzte Parameter mit dem zu identifizieren AWS CLI

Um gemeinsam genutzte Parameter mit dem zu identifizieren AWS CLI, können Sie zwischen dem Systems Manager `describe-parameters` Manager-Befehl und dem AWS RAM `list-resources` Befehl wählen.

Wenn Sie die `--shared` Option with verwenden`describe-parameters`, gibt der Befehl die Parameter zurück, die mit Ihnen gemeinsam genutzt werden.

Im Folgenden wird ein Beispiel gezeigt:

```
aws ssm describe-parameters --shared
```

Zugreifen auf gemeinsam genutzte Parameter

Verbraucher können mithilfe der AWS Befehlszeilentools und AWS SDKs auf gemeinsam genutzte Parameter zugreifen. Bei Verbraucherkonten sind Parameter, die mit diesem Konto gemeinsam genutzt werden, nicht auf der Seite Meine Parameter enthalten.

CLI-Beispiel: Zugreifen auf gemeinsam genutzte Parameterdetails mit dem AWS CLI

Um mit dem auf Details gemeinsam genutzter Parameter zuzugreifen AWS CLI, können Sie die get-parametersBefehle get-parameteroder verwenden. Sie müssen den vollständigen Parameter-ARN als angeben, um den `--name` Parameter von einem anderen Konto abzurufen.

Im Folgenden wird ein Beispiel gezeigt.

```
aws ssm get-parameter \
    --name arn:aws:ssm:us-east-2:123456789012:parameter/MySharedParameter
```

Unterstützte und nicht unterstützte Integrationen für gemeinsam genutzte Parameter

Derzeit können Sie gemeinsam genutzte Parameter in den folgenden Integrationsszenarien verwenden:

- AWS CloudFormation Vorlagenparameter
- Die Lambda-Erweiterung AWS Parameters and Secrets
- Startvorlagen für Amazon Elastic Compute Cloud (EC2)
- Werte für `ImageID` mit dem RunInstances EC2-Befehl zum Erstellen von Instances aus einem Amazon Machine Image () AMI
- Abrufen von Parameterwerten in Runbooks for Automation, eine Funktion von Systems Manager

Die folgenden Szenarien und integrierten Dienste unterstützen derzeit nicht die Verwendung gemeinsam genutzter Parameter:

- Parameter in Befehlen inRun Command, eine Funktion von Systems Manager
- AWS CloudFormation dynamische Verweise
- Die Werte von Umgebungsvariablen in AWS CodeBuild

- Die Werte der Umgebungsvariablen in AWS App Runner
- Der Wert eines Geheimnisses in Amazon Elastic Container Service

Berechtigungssätze für die gemeinsame Nutzung von Parametern

Benutzerkonten erhalten nur Lesezugriff auf die Parameter, die Sie mit ihnen teilen. Der Verbraucher kann den Parameter nicht aktualisieren oder löschen. Der Verbraucher kann den Parameter nicht mit einem dritten Konto teilen.

Wenn Sie eine Ressourcenfreigabe AWS Resource Access Manager für die gemeinsame Nutzung Ihrer Parameter erstellen, können Sie aus zwei AWS verwalteten Berechtigungssätzen wählen, um diesen schreibgeschützten Zugriff zu gewähren:

AWSRAMDefaultPermissionSSMParameterReadOnly

Zulässige Aktionen:`DescribeParameters`,, `GetParameter` `GetParameters`

AWSRAMPermissionSSMParameterReadOnlyWithHistory

Zulässige Aktionen:`DescribeParameters`,`GetParameter`,`GetParameters`, `GetParameterHistory`

Wenn Sie die Schritte unter Erstellen einer gemeinsamen Ressource AWS RAM im AWS RAM Benutzerhandbuch ausführen, wählen Sie `Parameter Store Advanced Parameters` als Ressourcentyp und eine dieser verwalteten Berechtigungen aus, je nachdem, ob Benutzer den Parameterverlauf einsehen sollen oder nicht.

Maximaler Durchsatz für gemeinsam genutzte Parameter

Systems Manager begrenzt den maximalen Durchsatz (Transaktionen pro Sekunde) für die Operationen GetParameterund GetParameters. Der Durchsatz wird auf der Ebene der einzelnen Konten durchgesetzt. Daher kann jedes Konto, das einen gemeinsamen Parameter verwendet, seinen maximal zulässigen Durchsatz nutzen, ohne von anderen Konten beeinflusst zu werden. Weitere Informationen zum maximalen Durchsatz für Parameter finden Sie in den folgenden Themen:

- Erhöhung des Parameter Store Durchsatzes
- Systems Manager Manager-Dienstkontingente in der Allgemeine Amazon Web Services-Referenz.

Preisgestaltung für gemeinsam genutzte Parameter

Kontoübergreifendes Teilen ist nur in der erweiterten Parameterstufe verfügbar. Für erweiterte Parameter fallen Gebühren zum aktuellen Preis für den Speicher und die API-Nutzung für jeden erweiterten Parameter an. Die Speicherung der erweiterten Parameter wird dem Eigentümerkonto in Rechnung gestellt. Für jedes nutzende Konto, das einen API-Aufruf an einen gemeinsam genutzten erweiterten Parameter tätigt, wird die Nutzung des Parameters in Rechnung gestellt.

Wenn Konto A beispielsweise einen erweiterten Parameter erstelltMyAdvancedParameter, werden diesem Konto 0,05 USD pro Monat für die Speicherung des Parameters berechnet.

Konto A wird dann MyAdvancedParameter mit Konto B und Konto C geteilt. Während eines Monats tätigen die drei Konten Anrufe beiMyAdvancedParameter. In der folgenden Tabelle sind die Gebühren aufgeführt, die für sie je nach Anzahl der von ihnen getätigten Anrufe anfallen würden.

> **Note**
>
> Die Gebühren in der folgenden Tabelle dienen nur der Veranschaulichung. Informationen zur Überprüfung der aktuellen Preise finden Sie unter AWS Systems Manager Preise für Parameter Store.

Account	Anzahl der Anrufe	Gebühren
Konto A (Besitzkonto)	10.000 Anrufe	• Erweiterter Parameter speicher für einen Monat: USD 0,05 • 10.000 Anrufe bisMyAdvancedParameter : USD 0,05 • Insgesamt: 0,10 USD
Konto B (verbrauchendes Konto)	20.000 Anrufe	• 20.000 Anrufe anMyAdvancedParameter : USD 0,10 • Insgesamt: 0,10 USD

Account	Anzahl der Anrufe	Gebühren
Konto C (verbrauchendes Konto)	30.000 Anrufe	• 30.000 Anrufe an `MyAdvancedParameter` : 0,15 USD • Insgesamt: 0,15 USD

Kontoübergreifender Zugriff für geschlossene AWS-Konten

Wenn der AWS-Konto , der einen gemeinsamen Parameter besitzt, geschlossen wird, verlieren alle Benutzerkonten den Zugriff auf den gemeinsamen Parameter. Wenn das Konto, das Eigentümer ist, innerhalb von 90 Tagen nach der Schließung des Kontos wieder geöffnet wird, erhalten die verbrauchenden Konten wieder Zugriff auf die zuvor gemeinsam genutzten Parameter. Weitere Informationen zur Wiedereröffnung eines Kontos während der Zeit nach der Schließung findest du im Referenzhandbuch unter Zugriff auf dein Konto, AWS-Konto nachdem du es geschlossen hast.AWS Account Management

Arbeiten mit Parametern unter Verwendung von Run Command-Befehlen

Sie können mit Parametern in arbeitenRun Command, eine Fähigkeit von AWS Systems Manager. Weitere Informationen finden Sie unter AWS Systems Manager Run Command.

Ausführen eines String-Parameters (Konsole)

Das folgende Verfahren führt Sie durch die Schritte zum Ausführen eines Befehls, der einen `String`-Parameter verwendet.

Ausführen eines String-Parameters mithilfe der Parameter Store

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option `AWS-RunPowerShellScript` (Windows) oder `AWS-RunShellScript` (Linux) aus.
5. Geben Sie für Command parameters (Befehlsparameter) Folgendes ein: **`echo {{ssm:parameter-name}}`**. Zum Beispiel: **`echo {{ssm:/Test/helloWorld}}`**.

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Other parameters (Weitere Parameter):

   - Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket

(Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

10. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.
11. Wählen Sie Ausführen aus.
12. Wählen Sie auf der Seite Command ID (Befehls-ID) im Bereich Targets and outputs (Ziele und Ausgaben) auf die Schaltfläche neben der ID eines Knotens, auf dem Sie den Befehl ausgeführt haben, und wählen Sie dann View output (Ausgabe anzeigen). Vergewissern Sie sich, dass der Befehl den Wert ausgibt, den Sie für den Parameter angegeben haben, z. B. **`This is my first parameter`**.

Ausführen eines Parameters (AWS CLI)

Beispiel 1: Einfacher Befehl

Der folgende Beispielbefehl enthält einen Systems Manager-Parameter mit der Bezeichnung `DNS-IP`. Der Wert dieses Parameters entspricht der IP-Adresse eines Knotens. In diesem Beispiel wird ein AWS Command Line Interface (AWS CLI) -Befehl verwendet, um den Parameterwert wiederzugeben.

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "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": []
    }
}
```

Nachdem die Ausführung eines Befehls abgeschlossen ist, können Sie mit den folgenden Befehlen weitere Informationen dazu anzeigen:

- get-command-invocation – Zeigt detaillierte Informationen zur Befehlsausführung an.
- list-command-invocations – Zeigt den Status der Befehlsausführung auf einem bestimmten verwalteten Knoten an.

- list-commands – Zeigt den Status der Befehlsausführung in verwalteten Knoten an.

Beispiel 2: Einen **`SecureString`**-Parameterwert entschlüsseln

Der nächste Beispielbefehl verwendet einen `SecureString` Parameter mit dem Namen SecurePassword. Mit dem im `parameters`-Feld verwendeten Befehl wird der Wert des `SecureString`-Parameters abgerufen und entschlüsselt. Anschließend wird das lokale Administratorpasswort zurückgesetzt, ohne dass das Passwort als Klartext übergeben wird.

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
```

Beispiel 3: Auf einen Parameter in einem SSM-Dokument verweisen

Sie können Systems Manager-Parameter auch im Abschnitt Parameters in einem SSM-Dokument referenzieren, wie im folgenden Beispiel gezeigt.

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

Verwechseln Sie die ähnliche Syntax für lokale Parameter, die im `runtimeConfig`-Abschnitt von SSM-Dokumenten verwendet werden, nicht mit Parameter Store-Parametern. Ein lokaler Parameter ist nicht dasselbe wie ein Systems Manager-Parameter. Sie können lokale Parameter daran erkennen, dass diese (im Gegensatz zu Systems Manager-Parametern) über kein `ssm:`-Präfix verfügen.

```
"runtimeConfig":{
        "aws:runShellScript":{
            "properties":[
                {
                    "id":"0.aws:runShellScript",
                    "runCommand":"{{ commands }}",
                    "workingDirectory":"{{ workingDirectory }}",
                    "timeoutSeconds":"{{ executionTimeout }}"
```

> **Note**
>
> SSM-Dokumente unterstützen keine Referenzen auf `SecureString`-Parameter. Um `SecureString`-Parameter beispielsweise mit Run Command verwenden zu können, müssen Sie daher den Parameterwert vor der Übergabe an Run Command wie in den folgenden Beispielen gezeigt abrufen.

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
```

Unterstützung für native Parameter für Amazon Machine Image-IDs

Wenn Sie einen `String`-Parameter erstellen, können Sie einen Datentyp als `aws:ec2:image` angeben, um sicherzustellen, dass der eingegebene Parameterwert ein gültiges Amazon Machine Image (AMI)-ID-Format aufweist.

Durch die Unterstützung von AMI-ID-Formaten können Sie vermeiden, dass alle Skripts und Vorlagen jedes Mal mit einer neuen ID aktualisiert werden, wenn sich das AMI ändert, das Sie in Ihren Prozessen verwenden möchten. Sie können einen Parameter mit dem Datentyp `aws:ec2:image` erstellen und für seinen Wert die ID eines AMI eingeben. Dies ist das AMI, von dem Sie neue Instances erstellen möchten. Anschließend verweisen Sie in Ihren Vorlagen, Befehlen und Skripts auf diesen Parameter.

Sie können beispielsweise den Parameter angeben, der die bevorzugte AMI-ID beim Ausführen des Amazon Elastic Compute Cloud (Amazon EC2) `run-instances`-Befehls enthält.

> **Note**
>
> Der Benutzer, der diesen Befehl ausführt, muss über AWS Identity and Access Management (IAM-) Berechtigungen verfügen, die den `ssm:GetParameters` API-Vorgang einschließen, damit der Parameterwert validiert werden kann. Andernfalls schlägt die Parametererstellung fehl.

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
```

Sie können auch Ihr gewünschtes AMI wählen, wenn Sie mit der Amazon EC2-Konsole eine Instance erstellen. Weitere Informationen finden Sie unter Verwenden eines Systems Manager Manager-Parameters AMI im Amazon EC2 EC2-Benutzerhandbuch.

Wenn Sie ein anderes AMI in Ihrem Workflow zur Instance-Erstellung verwenden möchten, müssen Sie nur den Parameter mit dem neuen AMI-Wert aktualisieren und Parameter Store überprüft erneut, ob die ID im richtigen Format eingegeben wurde.

Erteilen Sie Berechtigungen zum Erstellen eines Parameters aus dem Datentyp **`aws:ec2:image`**

Mithilfe von AWS Identity and Access Management (IAM-) Richtlinien können Sie Benutzern den Zugriff auf Parameter Store API-Operationen und -Inhalte gewähren oder einschränken.

Um einen `aws:ec2:image` Datentypparameter zu erstellen, muss der Benutzer `ssm:PutParameter` sowohl als auch über `ec2:DescribeImages` Berechtigungen verfügen.

Die folgende Beispielrichtlinie erteilt Benutzern die Berechtigung zum Aufrufen der API-Operation `PutParameter` für `aws:ec2:image`. Dies bedeutet, dass der Benutzer einen Parameter des Datentyps `aws:ec2:image`zum System hinzufügen kann.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:PutParameter",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeImages",
            "Resource": "*"
        }
    ]
}
```

Funktionsweise der AMI-Formatvalidierung

Wenn Sie `aws:ec2:image` als Datentyp für einen Parameter angeben, erstellt Systems Manager den Parameter nicht sofort. Stattdessen wird eine asynchrone Validierungsoperation ausgeführt, um sicherzustellen, dass der Parameterwert die Formatierungsanforderungen für eine AMI-ID erfüllt und das angegebene AMI in Ihrem AWS-Konto verfügbar ist.

Eine Parameterversionsnummer wird möglicherweise generiert, bevor die Validierungsoperation abgeschlossen ist. Die Operation wird möglicherweise nicht abgeschlossen, auch wenn eine Parameterversionsnummer geniert wird.

Um zu überprüfen, ob Ihre Parameter erfolgreich erstellt wurden, empfehlen wir, Amazon EventBridge zu verwenden, um Ihnen Benachrichtigungen über Ihre `create` und die `update`

Parameteroperationen zu senden. Diese Benachrichtigungen melden, ob eine Parameteroperation erfolgreich war oder nicht. Wenn eine Operation fehlschlägt, enthält die Benachrichtigung eine Fehlermeldung, die den Grund für den Fehler angibt.

```
{
    "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"
    }
}
```

Informationen zum Abonnieren von Parameter Store Veranstaltungen in finden Sie EventBridge unterEinrichten von Benachrichtigungen oder Auslöseraktionen basierend auf Parameter Store-Ereignissen.

Löschen von Systems-Manager-Parametern

In diesem Thema wird beschrieben, wie Sie Parameter löschen, die Sie inParameter Store, einer Funktion von, erstellt haben AWS Systems Manager.

Um einen Parameter zu löschen (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Aktivieren Sie auf der Registerkarte My parameters (Meine Parameter) das Kontrollkästchen neben jedem zu löschenden Parameter.

4. Wählen Sie Löschen aus.
5. Wählen Sie im Bestätigungs-Dialogfeld die Option Delete parameters (Parameter löschen).

Um einen Parameter zu löschen (AWS CLI)

- Führen Sie den folgenden Befehl aus:

```
aws ssm delete-parameter --name "my-parameter"
```

  Ersetzen Sie *my-parameter* durch den Namen Ihres Parameters, der gelöscht werden soll.

  Informationen zu allen Optionen, die für den `delete-parameter` Befehl verfügbar sind, finden Sie delete-parameterim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Arbeiten mit öffentlichen Parametern

Einige AWS-Services veröffentlichen Informationen über häufig verwendete Artefakte als AWS Systems Manager öffentliche Parameter. Der Amazon Elastic Compute Cloud (Amazon EC2)-Service beispielsweise veröffentlicht Informationen zu Amazon Machine Images (AMIs) als öffentliche Parameter.

Themen in diesem Leitfaden

- Auffinden von öffentlichen Parametern
- Aufrufen von öffentlichen AMI-Parametern
- Aufrufen der ECS-optimierten öffentlichen AMI-Parameter
- Aufrufen der EKS-optimierten öffentlichen AMI-Parameter
- Aufrufen öffentlicher Parameter für Regionen AWS-Services, Endpunkte, Availability Zones, lokale Zonen und Wellenlängenzonen

Verwandte AWS Blogbeiträge

- Query for AWS-Regionen, Endpoints und mehr unter Verwendung AWS Systems ManagerParameter Store
- Abfragen nach den aktuellen Amazon Linux AMI-IDs mit AWS Systems ManagerParameter Store
- Abfragen nach dem aktuellen Windows-AMI mit AWS Systems ManagerParameter Store

Auffinden von öffentlichen Parametern

Sie können mithilfe der Parameter Store-Konsole oder der AWS Command Line Interface nach öffentlichen Parametern suchen.

Ein öffentlicher Parametername beginnt mit `aws/service/list`. Der nächste Teil des Namens entspricht dem Service, dem dieser Parameter gehört.

Im Folgenden finden Sie eine Liste einiger Services, die öffentliche Parameter bereitstellen:

- `ami-amazon-linux-latest`
- `ami-windows-latest`
- `appmesh`
- `aws-for-fluent-bit`
- `bottlerocket`
- `canonical`
- `cloud9`
- `datasync`
- `debian`
- `ecs`
- `eks`
- `freebsd`
- `global-infrastructure`
- `marketplace`
- `storagegateway`

Nicht alle öffentlichen Parameter werden für alle veröffentlicht AWS-Region.

Auffinden von öffentlichen Parametern mithilfe der Parameter Store-Konsole

Sie müssen mindestens einen Parameter in Ihrem AWS-Konto und haben, AWS-Region bevor Sie mit der Konsole nach öffentlichen Parametern suchen können.

Auffinden von öffentlichen Parametern mithilfe der Konsole

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie die Registerkarte Öffentliche Parameter aus.
4. Wählen Sie das Dropdown-Menü Einen Service auswählen aus. Wählen Sie den Service aus, dessen Parameter Sie verwenden möchten.
5. (Optional) Filtern Sie die Parameter, die dem ausgewählten Dienst gehören, indem Sie weitere Informationen in die Suchleiste eingeben.
6. Wählen Sie den zu verwendenden öffentlichen Parameter aus.

Suchen nach öffentlichen Parametern mit dem AWS CLI

Verwenden Sie `describe-parameters`, um öffentliche Parameter zu entdecken.

Verwenden Sie `get-parameters-by-path`, um den tatsächlichen Pfad für einen Service zu erhalten, der unter `/aws/service/list` gelistet ist. Um den Pfad des Services abzurufen, entfernen Sie `/list` aus dem Pfad. Beispielsweise wird `/aws/service/list/ecs` zu `/aws/service/ecs`.

Um eine Liste von öffentlichen Parametern abzurufen, die verschiedenen Services in Parameter Store gehören, führen Sie den folgenden Befehl aus.

```
aws ssm get-parameters-by-path --path /aws/service/list
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

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

Wenn Sie Parameter anzeigen möchten, die einem bestimmten Service gehören, wählen Sie den Service aus der Liste aus, die nach dem Ausführen des vorherigen Befehls erstellt wurde. Dann machen Sie einen `get-parameters-by-path`-Aufruf nach dem Namen Ihres gewünschten Services.

z. B. `/aws/service/global-infrastructure`. Der Pfad kann einstufig sein (ruft nur Parameter auf, die genau den angegebenen Werten entsprechen) oder rekursiv (enthält Elemente im Pfad über das hinaus, was Sie angegeben haben).

> **Note**
>
> Der `/aws/service/global-infrastructure` Pfad wird nicht für Abfragen in allen Regionen unterstützt. Weitere Informationen finden Sie unter Aufrufen öffentlicher Parameter für Regionen AWS-Services, Endpunkte, Availability Zones, lokale Zonen und Wellenlängenzonen.

Wenn für den von Ihnen angegebenen Dienst keine Ergebnisse zurückgegeben werden, fügen Sie das `--recursive`-Flag hinzu, und führen Sie den Befehl erneut aus.

```
aws ssm get-parameters-by-path --path /aws/service/global-infrastructure
```

Dadurch werden alle Parameter im Besitz von `global-infrastructure` ausgegeben. Im Folgenden wird ein Beispiel gezeigt.

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

Sie können auch Parameter, die einem bestimmten Service gehören, anzeigen, indem Sie den `Option:BeginsWith`-Filter verwenden.

```
aws ssm describe-parameters --parameter-filters "Key=Name, Option=BeginsWith, Values=/
aws/service/ami-amazon-linux-latest"
```

Der Befehl gibt Informationen wie die folgenden zurück. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

```
{
    "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**
>
> Die zurückgegebenen Parameter können unterschiedlich sein, wenn Sie `Option=BeginsWith` verwenden, da es ein anderes Suchmuster verwendet.

Aufrufen von öffentlichen AMI-Parametern

Öffentliche Parameter von Amazon Elastic Compute Cloud Amazon Machine Image (Amazon EC2AMI) () sind für Amazon Linux 1, Amazon Linux 2, Amazon Linux 2023 (AL2023) und über die folgenden Windows Server Pfade verfügbar:

- Amazon Linux 1, Amazon Linux 2 und Amazon Linux 2023: `/aws/service/ami-amazon-linux-latest`
- Windows Server: `/aws/service/ami-windows-latest`

Aufrufen AMI öffentlicher Parameter für Amazon Linux 1, Amazon Linux 2 und Amazon Linux 2023

Sie können eine Liste aller aktuellen Amazon Linux 1, Amazon Linux 2 und Amazon Linux 2023 (AL2023) AMIs anzeigen, AWS-Region indem Sie den folgenden Befehl in der AWS Command Line Interface (AWS CLI) verwenden.

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "/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"
]
```

Sie können weitere Details zu diesen AMIs, einschließlich der AMI-IDs und Amazon-Ressourcennamen (ARNs) anzeigen, indem Sie den folgenden Befehl ausführen.

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
```

region steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

Der Befehl gibt Informationen wie die folgenden zurück. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

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

Sie können Details zu einer bestimmten Datei anzeigen, AMI indem Sie den GetParametersAPI-Vorgang mit dem vollständigen AMI Namen, einschließlich des Pfads, verwenden. Hier sehen Sie ein Beispiel für einen Befehl.

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
```

Der Befehl gibt die folgenden Informationen zurück.

```
{
    "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": []
}
```

Aufrufen von öffentlichen AMI-Parametern für Windows Server

Sie können eine Liste aller Windows Server AMIs aktuellen Einträge anzeigen, AWS-Region indem Sie den folgenden Befehl in der verwenden 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
```

Der Befehl gibt Informationen wie die folgenden zurück. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

```
[
    "/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
]
```

Sie können weitere Details zu diesen AMIs, einschließlich der AMI-IDs und Amazon-Ressourcennamen (ARNs) anzeigen, indem Sie den folgenden Befehl ausführen.

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
```

Region steht für den Bezeichner einer Region AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

Der Befehl gibt Informationen wie die folgenden zurück. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

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


Sie können Details zu einer bestimmten Datei anzeigen, AMI indem Sie den GetParametersAPI-Vorgang mit dem vollständigen AMI Namen, einschließlich des Pfads, verwenden. Hier sehen Sie ein Beispiel für einen Befehl.

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
```

Der Befehl gibt die folgenden Informationen zurück.

```
{
    "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": []
}
```

Aufrufen der ECS-optimierten öffentlichen AMI-Parameter

Der Amazon Elastic Container Service (Amazon ECS)-Service veröffentlicht den Namen des neuesten von Amazon ECS optimierten Amazon Machine Images (AMIs) als öffentlichen Parameter. Benutzer sollten dieses AMI beim Erstellen eines neuen Amazon Elastic Compute Cloud (Amazon EC2)-Clusters für Amazon ECS verwenden, da das optimierte AMIs Fehlerbehebungen und Funktionsaktualisierungen enthält.

Führen Sie den folgenden Befehl aus, um den Namen des neuesten von Amazon ECS optimierten AMI für Amazon Linux 2 anzuzeigen. Befehle für andere Betriebssysteme finden Sie unter Amazon ECS-optimierte AMI-Metadaten abrufen im Entwicklerhandbuch für 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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "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": []
}
```

Aufrufen der EKS-optimierten öffentlichen AMI-Parameter

Der Amazon Elastic Kubernetes Service (Amazon EKS)-Service veröffentlicht den Namen des neuesten von Amazon EKS optimiertenAmazon Machine Image (AMI) als öffentlichen Parameter. Benutzer sollten beim Hinzufügen von Knoten zu einem Amazon EKS-Cluster dieses AMI verwenden, da neue Versionen Kubernetes-Patches und -Sicherheitsupdates enthalten. Um sicherzustellen, dass Sie das neueste AMI verwenden, mussten Sie bisher die Amazon EKS-Dokumentation überprüfen und Bereitstellungsvorlagen oder Ressourcen manuell mit der neuen AMI-ID aktualisieren.

Führen Sie den folgenden Befehl aus, um den Namen des neuesten von Amazon EKS optimierten AMI für Amazon Linux 2 anzuzeigen.

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "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": []
}
```

Aufrufen öffentlicher Parameter für Regionen AWS-Services, Endpunkte, Availability Zones, lokale Zonen und Wellenlängenzonen

Sie können die öffentlichen Parameter Service AWS-Region, Endpoint, Availability und Wavelength Zones aufrufen, indem Sie den folgenden Pfad verwenden.

`/aws/service/global-infrastructure`

Note

Derzeit `/aws/service/global-infrastructure` wird der Pfad AWS-Regionen nur für Abfragen in den folgenden Bereichen unterstützt:

- USA Ost (Nord-Virginia): (us-east-1)
- USA Ost (Ohio): (us-east-2)
- USA West (Nordkalifornien) (us-west-1)
- USA West (Oregon): (us-west-2)
- Asien-Pazifik (Hongkong) (ap-east-1)
- Asien-Pazifik (Mumbai): (ap-south-1)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Kanada (Zentral): (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Europa (Stockholm) (eu-north-1)
- Südamerika (São Paulo) (sa-east-1)

Wenn Sie in einer anderen Handelsregion arbeiten, können Sie in Ihrer Abfrage eine unterstützte Region angeben, um die Ergebnisse anzuzeigen. Wenn Sie beispielsweise in der Region Canada West (Calgary) (ca-west-1) arbeiten, könnten Sie in Ihrer Abfrage Canada (Central) (ca-central-1) angeben:

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/regions \
    --region ca-central-1
```

Aktiv anzeigen AWS-Regionen

Sie können eine Liste aller aktiven anzeigen, AWS-Regionen indem Sie den folgenden Befehl in der AWS Command Line Interface (AWS CLI) verwenden.

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "/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"
]
```

Ansicht verfügbar AWS-Services

Mit dem folgenden Befehl können Sie eine vollständige Liste aller verfügbaren anzeigen AWS-Services und sie in alphabetischer Reihenfolge sortieren. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

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(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
[
     "/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",
```

Unterstützte Regionen anzeigen für AWS-Service

Sie können sich eine Liste ansehen AWS-Regionen , wo ein Service verfügbar ist. In diesem Beispiel wird AWS Systems Manager (`ssm`) verwendet.

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "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"
]
```

Anzeigen des regionalen Endpunkts für einen Service

Sie können einen regionalen Endpunkt für einen Service anzeigen, indem Sie den folgenden Befehl ausführen. Mit diesem Befehl wird die Region USA Ost (Ohio) (us-east-2) abgefragt.

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
"ssm.us-east-2.amazonaws.com"
```

Anzeigen der vollständigen Details zu Availability Zones

Sie können den folgenden Befehl verwenden, um Availability Zones anzuzeigen

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/
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
{
```

```
    "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"
        }
    ]
}
```

Anzeigen ausschließlich der Namen von Availability Zones

Sie können den folgenden Befehl verwenden, um ausschließlich die Namen von Availability Zones anzuzeigen.

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(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
[
    "/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"
```

Anzeigen der Namen von Availability Zones in einer einzelnen Region

Sie können den folgenden Befehl verwenden, um die Namen der Availability Zones in einer Region (in diesem Beispiel `us-east-2`) anzuzeigen.

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(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "/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"
```

Anzeigen ausschließlich der ARNs von Availability Zones

Sie können den folgenden Befehl verwenden, um ausschließlich die Amazon-Ressourcennamen (ARN) von Availability Zones anzuzeigen.

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(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
[
    "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",
```

Anzeigen der Details zu lokalen Zonen

Sie können den folgenden Befehl verwenden, um lokale Zonen anzuzeigen.

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
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

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

Anzeigen von Details zu Wavelength Zones

Sie können den folgenden Befehl verwenden, um Wavelength Zones anzuzeigen.

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
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

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

Anzeigen aller Parameter und Werte unter einer lokalen Zone

Sie können den folgenden Befehl verwenden, um alle Parameterdaten für eine lokale Zone anzuzeigen.

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"
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

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

Anzeigen ausschließlich der Namen von Parametern für lokale Zonen

Sie können den folgenden Befehl verwenden, um ausschließlich die Namen der Parameter für lokale Zonen anzuzeigen.

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(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "/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"
]
```

Walkthroughs für Parameter Store

Der Walkthrough in diesem Abschnitt zeigt, wie Sie Parameter mit Parameter Store, einer Funktion von AWS Systems Manager, in einer Testumgebung erstellen, speichern und ausführen können. Diese Anleitungen zeigen Ihnen die Verwendung von Parameter Store mit anderen Systems Manager-Funktionen. Sie können Parameter Store auch mit anderen AWS-Services verwenden. Weitere Informationen finden Sie unter Was ist ein Parameter?.

Inhalt

- Erstellen eines SecureString-Parameters und Verknüpfen eines Knotens mit einer Domain (PowerShell)

- Verwenden von Parameter Store-Parametern in Amazon Elastic Kubernetes Service

Erstellen eines SecureString-Parameters und Verknüpfen eines Knotens mit einer Domain (PowerShell)

Dieser Walkthrough zeigt Ihnen das Verknüpfen eines Windows Server-Knotens mit einer Domain mittels AWS Systems Manager `SecureString`-Parametern und Run Command. In der Anleitung werden typische Domain-Parameter verwendet, z. B. der Domain-Name und ein Benutzername für die Domain. Diese Werte werden als unverschlüsselte Zeichenfolgen weitergegeben. Das Passwort für die Domain wird unter Verwendung eines Von AWS verwalteter Schlüsselverschlüsselt und als verschlüsselte Zeichenfolge übergeben.

Voraussetzungen

In dieser Anleitung wird davon ausgegangen, dass Sie Ihren Domain-Namen und die DNS-Server-IP-Adresse in der DHCP-Optionsliste, die Ihrer Amazon VPC zugeordnet ist, bereits angegeben haben. Informationen finden Sie unter Arbeiten mit DHCP-Optionslisten im Amazon VPC-Benutzerhandbuch.

So erstellen Sie einen **`SecureString`**-Parameter und verknüpfen einen Knoten mit einer Domain

1. Geben Sie die Parameter mithilfe von AWS Tools for Windows PowerShell in das System ein.

   Ersetzen Sie in den folgenden Befehlen jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

   ```
   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**
   >
   > Nur der Wert eines `SecureString`-Parameters wird verschlüsselt. Der Name des Parameters, die Beschreibung und andere Eigenschaften sind nicht verschlüsselt.

2. Fügen Sie die folgenden AWS Identity and Access Management (IAM)-Richtlinien an die IAM-Rollenberechtigungen für Ihren Knoten an:

   - AmazonSSMManagedInstanceCore – Erforderlich. Diese AWS-verwaltete Richtlinie erlaubt es einem verwalteten Knoten, Systems-Manager-Service-Kernfunktionalität zu verwenden.

- AmazonSSMDirectoryServiceAccess – Erforderlich. Diese AWS-verwaltete Richtlinie erlaubt SSM Agent in Ihrem Namen den Zugriff auf AWS Directory Service für Anforderungen zum Beitritt zur Domain von dem verwalteten Knoten.
- Eine benutzerdefinierte Richtlinie für S3-Bucket-Zugriff – Erforderlich. SSM Agent, der sich auf Ihrem Knoten befindet und Systems-Manager-Aufgaben ausführt, erfordert Zugriff auf bestimmte Amazon Simple Storage Service (Amazon S3)-Buckets im Besitz von Amazon. In der benutzerdefinierten S3-Bucket-Richtlinie, die Sie erstellen, können Sie auch Zugriff auf Ihre eigenen S3-Buckets gewähren, die für Systems Manager-Operationen benötigt werden.

  Beispiele: Sie können Ausgabe für Run Command-Befehle oder Session Manager-Sitzungen in einen S3-Bucket schreiben und diese Ausgabe dann zu einem späteren Zeitpunkt für Auditing-Zwecke oder zur Fehlerbehebung nutzen. Sie speichern Zugriffsskripts oder benutzerdefinierte Patch-Baseline-Listen in einem S3-Bucket und verweisen dann auf das Skript oder die Liste, wenn Sie einen Befehl ausführen oder wenn eine Patch-Baseline angewendet wird.

  Weitere Informationen zum Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen Amazon S3-Bucket finden Sie unter Erstellen einer benutzerdefinierten S3-Bucket-Richtlinie für ein Instance-Profil

  > **Note**
  >
  > Die Speicherung von Ausgabeprotokolldaten in einem S3-Bucket ist optional. Wenn Sie sich jedoch hierzu entschlossen haben, sollte die Funktion zu Beginn des Systems Manager-Konfigurationsprozesses eingerichtet werden. Weitere Informationen finden Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service.

- CloudWatchAgentServerPolicy – Optional. Diese AWS-verwaltete Richtlinie erlaubt es Ihnen, den CloudWatch-Agenten auf verwalteten Knoten auszuführen. Diese Richtlinie ermöglicht es, Informationen auf einem Knoten zu lesen und sie in Amazon CloudWatch zu schreiben. Ihr Instance-Profil benötigt diese Richtlinie nur, wenn Sie Services wie Amazon EventBridge oder CloudWatch Logs verwenden.

  > **Note**
  >
  > Die Verwendung von CloudWatch- und EventBridge-Features ist optional. Es wird aber empfohlen, sie am Anfang Ihres Systems Manager-Konfigurationsprozesses

einzurichten, wenn Sie sich für deren Verwendung entschieden haben. Weitere Informationen finden Sie im Amazon EventBridge-Benutzerhandbuch und dem Amazon CloudWatch Logs-Benutzerhandbuch.

3. Bearbeiten Sie die IAM-Rolle, die dem Knoten zugeordnet ist, und fügen Sie die folgende Richtlinie hinzu. Diese Richtlinie erteilt dem Knoten Berechtigungen, um die `kms:Decrypt`- und `ssm:CreateDocument`-API aufrufen zu können.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "ssm:CreateDocument"
            ],
            "Resource": [
                "arn:aws:kms:region:account-id:key/kms-key-id"
            ]
        }
    ]
}
```

4. Kopieren Sie den folgenden JSON-Text in einen Texteditor und speichern Sie die Datei unter dem Namen `JoinInstanceToDomain.json` am folgenden Speicherort: `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. Führen Sie den folgenden Befehl in Tools for Windows PowerShell aus, um ein neues SSM-Dokument zu erstellen.

```
$json = Get-Content C:\temp\JoinInstanceToDomain | Out-String
New-SSMDocument -Name JoinInstanceToDomain -Content $json -DocumentType Command
```

6. Führen Sie den folgenden Befehl in Tools for Windows PowerShell aus, um den Knoten mit der Domain zu verknüpfen.

```
Send-SSMCommand -InstanceId instance-id -DocumentName JoinInstanceToDomain
```

Wenn der Befehl erfolgreich ausgeführt wurde, sieht das Ergebnis im System in etwa wie folgt aus:

```
WARNING: The changes will take effect after you restart the computer EC2ABCD-
EXAMPLE.
Domain join succeeded, restarting
Computer is part of example.local, exiting
```

Wenn der Befehl nicht erfolgreich ausgeführt wurde, sieht das Ergebnis im System in etwa wie folgt aus:

```
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.
```

Verwenden von Parameter Store-Parametern in Amazon Elastic Kubernetes Service

Um Geheimnisse aus Secrets Manager und Parameter aus Parameter Store Dateien anzuzeigen, die in Amazon EKS-Pods gemountet sind, können Sie den AWS Secrets and Configuration Provider (ASCP) für den Kubernetes Secrets Store CSI-Treiber verwenden. (Parameter Storeist eine Fähigkeit von.) AWS Systems Manager Das ASCP funktioniert mit Amazon Elastic Kubernetes Service (Amazon EKS) 1.17+. AWS Fargate (Fargate) Knotengruppen werden nicht unterstützt.

Mit dem ASCP können Sie Parameter abrufen, die in Parameter Store gespeichert und verwaltete werden. Dann können Sie die Parameter in Ihren Workloads verwenden, die auf Amazon EKS ausgeführt werden. Wenn Ihr Paramter mehrere Schlüssel/Wert-Paare im JSON-Format enthält, können Sie optional auswählen, welche in Amazon EKS bereitgestellt werden sollen. Der ASCP verwendet JMESPath-Syntax, um die Schlüssel/Wert-Paare in Ihrem Parameter abzufragen.

Sie können AWS Identity and Access Management (IAM) -Rollen und -Richtlinien verwenden, um den Zugriff auf Ihre Parameter auf bestimmte Amazon EKS-Pods in einem Cluster zu beschränken. Der ASCP ruft die Pod-Identität ab und tauscht die Identität gegen eine IAM-Rolle. ASCP übernimmt die IAM-Rolle des Pods. Dann kann es Parameter von Parameter Store abrufen, die für diese Rolle autorisiert sind.

Informationen zur Integration von Secrets Manager in Amazon EKS finden Sie unter Secrets Manager-Secrets in Amazon Elastic Kubernetes Service verwenden.

Installieren des ASCP

Das ASCP ist GitHub im secrets-store-csi-driver-provider-aws-Repository verfügbar. Das Repository enthält auch YAML-Beispieldateien zum Erstellen und Mounten eines Secrets. Sie installieren zuerst den Kubernetes-Secrets-Store-CSI-Treiber und dann den ASCP.

So installieren Sie den Kubernetes-Secrets-Store-CSI-Treiber und den ASCP

1. Führen Sie die folgenden Befehle aus, um den Kubernetes-Secrets-Store-CSI-Treiber zu installieren. Eine vollständige Installationsanweisung finden Sie unter Installation im Kubernetes-Secrets-Store-CSI-Treiberhandbuch. Weitere Informationen zur Installation von Helm finden Sie unter Verwenden von Helm mit Amazon EKS.

   ```
   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
   ```

2. Verwenden Sie zur Installation des ASCP die YAML-Datei im Bereitstellungsverzeichnis des Repositorys. GitHub Informationen zur Installation von `kubectl` finden Sie im Abschnitt Installieren der `kubectl`.

   ```
   kubectl apply -f https://raw.githubusercontent.com/aws/secrets-store-csi-driver-provider-aws/main/deployment/aws-provider-installer.yaml
   ```

Schritt 1: Einrichten der Zugriffssteuerung

Um Ihrem Amazon-EKS-Pod Zugriff auf Parameter in Parameter Store zu gewähren, erstellen Sie zunächst eine Richtlinie, die den Zugriff auf die Parameter einschränkt, auf die der Pod zugreifen muss. Erstellen Sie dann eine IAM role for service account (IAM-Rolle für Dienstkonto) und fügen Sie die Richtlinie an diese an. Weitere Informationen zum Einschränken des Zugriffs auf Systems Manager-Parameter mithilfe von IAM-Richtlinien finden Sie unter Einschränken des Zugriffs auf Systems Manager-Parameter mithilfe von IAM-Richtlinien.

> **Note**
>
> Bei Verwendung von Parameter Store-Parametern, wird die Berechtigung `ssm:GetParameters` in der Richtlinie benötigt.

Der ASCP ruft die Pod-Identität ab und tauscht sie gegen die IAM-Rolle. ASCP übernimmt die IAM-Rolle des Pods, wodurch er Zugriff auf die von Ihnen autorisierten Parameter erhält. Andere Container können nur auf die Parameter zugreifen, wenn Sie diese auch der IAM-Rolle zuordnen.

Schritt 2: Mounten von Parametern in Amazon EKS

Um Parameter in Amazon EKS wie Dateien im Dateisystem anzuzeigen, erstellen Sie eine `SecretProviderClass`-YAML-Datei mit Informationen zu Ihren Parametern und dem Mounten der Parameter im Amazon-EKS-Pod.

`SecretProviderClass` muss sich im gleichen Namespace wie der Amazon-EKS-Pod befinden, auf den verwiesen wird.

`SecretProviderClass`

Die `SecretProviderClass`-YAML-Datei hat folgendes Format:

```
apiVersion: secrets-store.csi.x-k8s.io/v1alpha1
kind: SecretProviderClass
metadata:
   name: <NAME>
spec:
  provider: aws
  parameters:
```

parameters (Parameter)

Enthält die Details der Mounting-Anfrage.

objects

Eine Zeichenfolge, die eine YAML-Deklaration der bereitzustellenden Parameter enthält. Wir empfehlen, eine mehrzeilige YAML-Zeichenfolge oder ein Pipe-Zeichen (|) zu verwenden.

objectName (Objektname)

Der Anzeigename des Parameters. Dies wird der Dateiname des Parameters im Amazon-EKS-Pod, es sei denn, Sie geben `objectAlias` an. Dabei muss Parameter Store der `Name` des Parameters sein und kann kein vollständiger Amazon-Ressourcenname (ARN) sein.

jmesPath

(Optional) Eine Zuordnung der Schlüssel im JSON-kodierten Parameter zu den Dateien, die in Amazon EKS bereitgestellt werden sollen. Das folgende Beispiel zeigt, wie ein JSON-kodierter Parameter aussieht.

```
{
    "username" : "myusername",
    "password" : "mypassword"
}
```

Die Schlüssel sind `username` und `password`. Der Wert, der mit `username` verbunden ist, ist `myusername`, und der Wert, der mit `password` verbunden ist, ist `mypassword`.

Pfad

Der Schlüssel im Parameter.

objectAlias

Der Dateiname, der im Amazon-EKS-Pod bereitgestellt werden soll.

objectType

Dies ist ein Pflichtfeld für Parameter Store. Verwenden Sie `ssmparameter`.

objectAlias

(Optional) Der Dateiname des Parameters im Amazon-EKS-Pod. Wenn Sie dieses Feld nicht angeben, wird `objectName` als Dateiname angezeigt.

objectVersion (Objektversion)

Optional: Die Versionsnummer des Parameters. Es wird empfohlen, dieses Feld nicht zu verwenden, da Sie es jedes Mal aktualisieren müssen, wenn Sie den Parameter aktualisieren. Standardmäßig wird die neueste Version verwendet. Für Parameter Store-Parameter können Sie `objectVersion`oder `objectVersionLabel` verwenden, aber nicht beides.

objectVersionLabel

(Optional) Die Parameterbeschriftung für die Version. Die Standardversion ist die neueste Version. Für Parameter Store-Parameter können Sie `objectVersion`oder `objectVersionLabel` verwenden, aber nicht beides.

Region

(Optional) Der Wert AWS-Region des Parameters. Wenn Sie dieses Feld nicht verwenden, sucht der ASCP die Region aus der Anmerkung auf dem Knoten. Diese Suche steigert den Overhead

von Mounting-Anfragen. Daher wird empfohlen, die Region für Cluster mit einer großen Anzahl von Pods anzugeben.

pathTranslation (Pfadangabe)

(Optional) Ein einzelnes Ersetzungszeichen, das verwendet werden soll, wenn der Dateiname (`objectName` oder`objectAlias`) das Pfadtrennzeichen enthält, z. B. Schrägstrich (/) unter Linux. Wenn ein Parametername das Pfadtrennzeichen enthält, kann ASCP keine eingehängte Datei mit diesem Namen erstellen. Stattdessen können Sie das Pfadtrennzeichen durch ein anderes Zeichen ersetzen, das Sie in dieses Feld eingeben. Wenn Sie dieses Feld nicht verwenden, ist der Standardwert ein Unterstrich (_), d. h. `My/Path/Parameter` wird als `My_Path_Parameter` bereitgestellt.

Um die Zeichenersetzung zu verhindern, geben Sie die Zeichenfolge `False` ein.

Beispiel

Die folgende Beispielkonfiguration zeigt eine `SecretProviderClass` mit einer Parameter Store-Parameterressource.

```
apiVersion: secrets-store.csi.x-k8s.io/v1alpha1
kind: SecretProviderClass
metadata:
    name: aws-secrets
spec:
    provider: aws
    parameters:
        objects: |
            - objectName: "MyParameter"
              objectType: "ssmparameter"
```

Schritt 3: Aktualisieren der Bereitstellungs-YAML

Aktualisieren Sie Ihre Bereitstellungs-YAML, damit sie die `secrets-store.csi.k8s.io`-Treiber verwendet und auf die`SecretProviderClass`-Ressource verweist, die im vorherigen Schritt erstellt wurde. Dadurch wird sichergestellt, dass Ihr Cluster den Secrets-Store-CSI-Treiber verwendet.

Im Folgenden finden Sie eine Beispiel-Bereitstellungs-YAML mit einer `SecretProviderClass` mit dem Namen `aws-secrets`.

```
volumes:
```

```
  - name: secrets-store-inline
    csi:
      driver: secrets-store.csi.k8s.io
      readOnly: true
      volumeAttributes:
        secretProviderClass: "aws-secrets"
```

Tutorial: Erstellen Sie einen Parameter und mounten Sie ihn in einem Amazon-EKS-Pod

In diesem Tutorial erstellen Sie einen Beispiel-Parameter in Parameter Store mounten den Parameter dann in einem Amazon-EKS-Pod und stellen ihn bereit.

Bevor Sie beginnen, installieren Sie den ASCP. Weitere Informationen finden Sie unter the section called “Installieren des ASCP”.

Ein Secret erstellen und mounten

1. Legen Sie den AWS-Region und den Namen Ihres Clusters als Shell-Variablen fest, damit Sie sie in `bash` Befehlen verwenden können. Geben Sie für *`Region`* den Ort ein, AWS-Region in dem Ihr Amazon EKS-Cluster ausgeführt wird. Geben Sie unter *`Cluster-Name`* einen Namen für Ihren Cluster ein.

   ```
   REGION=region
   CLUSTERNAME=clustername
   ```

2. Erstellen Sie einen Test-Parameter.

   ```
   aws ssm put-parameter --name "MyParameter" --value "EKS parameter" --type String --
   region "$REGION"
   ```

3. Erstellen Sie eine Ressourcenrichtlinie für den Pod, die den Zugriff auf den Parameter beschränkt, den Sie im vorherigen Schritt erstellt haben. Verwenden Sie für *`parameter-arn`* den ARN des Parameters. Speichern Sie den Richtlinien-ARN in einer Shell-Variablen. Um den Parameter-ARN abzurufen, verwenden Sie `get-parameter`.

   ```
   POLICY_ARN=$(aws --region "$REGION" --query Policy.Arn --output text iam create-
   policy --policy-name nginx-parameter-deployment-policy --policy-document '{
       "Version": "2012-10-17",
       "Statement": [ {
           "Effect": "Allow",
           "Action": ["ssm:GetParameter", "ssm:GetParameters"],
   ```

```
        "Resource": ["parameter-arn"]
    } ]
}')
```

4. Erstellen Sie einen IAM OpenID Connect (OIDC)-Anbieter für den Cluster, wenn Sie noch keinen haben. Weitere Informationen finden Sie unter Erstellen eines IAM-OIDC-Anbieters für Ihren Cluster.

```
eksctl utils associate-iam-oidc-provider --region="$REGION" --
cluster="$CLUSTERNAME" --approve # Only run this once
```

5. Erstellen Sie das Dienstkonto, das der Pod verwendet, und ordnen Sie die Ressourcenrichtlinie, die Sie in Schritt 3 erstellt haben, diesem Dienstkonto zu. Für dieses Tutorial verwenden Sie für den Namen des Dienstkontos. nginx-deployment-sa Weitere Informationen finden Sie unter Erstellen einer IAM-Rolle für ein Servicekonto.

```
eksctl create iamserviceaccount --name nginx-deployment-sa --region="$REGION" --
cluster "$CLUSTERNAME" --attach-policy-arn "$POLICY_ARN" --approve --override-
existing-serviceaccounts
```

6. Erstellen Sie `SecretProviderClass`, um anzugeben, welcher Parameter im Pod gemounted werden soll. Der folgende Befehl verwendet den Dateispeicherort einer `SecretProviderClass`-Datei mit dem Namen `ExampleSecretProviderClass.yaml`. Informationen zum Erstellen Ihrer eigenen `SecretProviderClass` finden Sie unter the section called “`SecretProviderClass`”.

```
kubectl apply -f ./ExampleSecretProviderClass.yaml
```

7. Ihr Pod bereitstellen Der folgende Befehl verwendet eine Bereitstellungsdatei mit dem Namen `ExampleDeployment.yaml`. Informationen zum Erstellen Ihrer eigenen `SecretProviderClass` finden Sie unter the section called “Schritt 3: Aktualisieren der Bereitstellungs-YAML”.

```
kubectl apply -f ./ExampleDeployment.yaml
```

8. Um zu überprüfen, ob der Parameter ordnungsgemäß gemounted wurde, verwenden Sie den folgenden Befehl und bestätigen Sie, dass Ihr Parameterwert angezeigt wird.

```
kubectl exec -it $(kubectl get pods | awk '/nginx-deployment/{print $1}' | head -1)
 cat /mnt/secrets-store/MyParameter; echo
```

Der Parameterwert wird angezeigt.

```
"EKS parameter"
```

Fehlerbehebung

Sie können die meisten Fehler anzeigen, indem Sie die Pod-Bereitstellung beschreiben.

Fehlermeldungen für Ihren Container anzeigen

1. Erstellen Sie mit dem folgenden Befehl eine Liste der Pod-Namen. Wenn Sie nicht den Standard-Namespace verwenden, verwenden Sie `-n <NAMESPACE>`.

   ```
   kubectl get pods
   ```

2. Um den Pod zu beschreiben, geben Sie im folgenden Befehl für *pod-id* die Pod-ID aus den Pods an, die Sie im vorherigen Schritt gefunden haben. Wenn Sie nicht den Standard-Namespace verwenden, verwenden Sie `-n <NAMESPACE>`.

   ```
   kubectl describe pod/pod-id
   ```

Fehler für den ASCP anzeigen

- Um weitere Informationen in den Anbieterprotokollen zu finden, verwenden Sie im folgenden Befehl für *pod-id die ID* des csi-secrets-store-provider-aws-Pods.

  ```
  kubectl -n kube-system get pods
  kubectl -n kube-system logs pod/pod-id
  ```

Prüfen und Protokollieren von Parameter Store-Aktivitäten

AWS CloudTrail erfasst über die AWS Systems Manager-Konsole, die AWS Command Line Interface (AWS CLI) und das Systems Manager SDK ausgeführte API-Aufrufe. Sie können die Informationen in der CloudTrail-Konsole oder in einem Amazon Simple Storage Service (Amazon S3)-Bucket anzeigen. Alle CloudTrail-Protokolle in Ihrem Konto verwenden nur ein Bucket. Weitere Informationen zum Anzeigen und Verwenden von CloudTrail-Protokollen von Systems Manager-Aktivitäten

finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail. Weitere Informationen zu den Prüfungs- und Protokollierungsoptionen für Systems Manager finden Sie unter Überwachung AWS Systems Manager.

Fehlerbehebung für Parameter Store

Verwenden Sie die folgenden Informationen, um Probleme mit , einer Funktion von Parameter Store, zu beheben AWS Systems Manager.

Fehlerbehebung bei der Erstellung von **`aws:ec2:image`**-Parametern

Verwenden Sie die folgenden Informationen, um Probleme bei der Erstellung von `aws:ec2:image`-Datentypparametern zu beheben.

Keine Berechtigung zum Erstellen einer Instance

Problem : Sie versuchen, eine Instance mit einem -`aws:ec2:image`Parameter zu erstellen, erhalten jedoch eine Fehlermeldung wie „Sie sind nicht berechtigt, diesen Vorgang auszuführen“.

- Lösung : Sie verfügen nicht über alle Berechtigungen, die zum Erstellen einer EC2-Instance mit einem Parameterwert erforderlich sind, z. B. über Berechtigungen für `ec2:RunInstances`, `ec2:DescribeImages`und `ssm:GetParameter`. Wenden Sie sich an einen Benutzer mit Administratorberechtigungen in Ihrer Organisation, um die erforderlichen Berechtigungen anzufordern.

EventBridge meldet die Fehlermeldung „Ressource kann nicht beschrieben werden“

Problem: Sie haben einen Befehl ausgeführt, um einen `aws:ec2:image`-Parameter zu erstellen, die Parametererstellung ist jedoch fehlgeschlagen. Sie erhalten eine Benachrichtigung von Amazon EventBridge , die die Ausnahme „Ressource kann nicht beschrieben werden“ meldet.

Lösung: Diese Meldung kann auf Folgendes hinweisen:

- Sie verfügen nicht über alle für die `ec2:DescribeImages`-API-Operation erforderlichen Berechtigungen oder Sie haben keine Berechtigung für den Zugriff auf das spezifische Image, auf das im Parameter verwiesen wird. Wenden Sie sich an einen Benutzer mit Administratorberechtigungen in Ihrer Organisation, um die erforderlichen Berechtigungen anzufordern.

- Die Amazon Machine Image (AMI)-ID, die Sie als Parameterwert eingegeben haben, ist ungültig. Stellen Sie sicher, dass Sie die ID eines eingebenAMI, der im aktuellen AWS-Region und Konto verfügbar ist, in dem Sie arbeiten.

Es ist kein neuer **`aws:ec2:image`**-Parameter verfügbar

Problem: Sie haben gerade einen Befehl ausgeführt, um einen `aws:ec2:image`-Parameter zu erstellen, und eine Versionsnummer wurde gemeldet, der Parameter ist jedoch nicht verfügbar.

- Lösung: Wenn Sie den Befehl zum Erstellen eines Parameters ausführen, der den Datentyp `aws:ec2:image` verwendet, wird sofort eine Versionsnummer für den Parameter generiert. Das Parameterformat muss jedoch validiert werden, bevor der Parameter verfügbar ist. Dieser Vorgang kann einige Minuten dauern. Wenn Sie die Parametererstellung- und -validierung überwachen möchten, können Sie Folgendes tun:
  - Verwenden Sie EventBridge , um Ihnen Benachrichtigungen über Ihre - `create` und -`update`Parameteroperationen zu senden. Diese Benachrichtigungen melden, ob eine Parameteroperation erfolgreich war oder nicht. Informationen zum Abonnieren von Parameter Store Ereignissen in finden Sie EventBridgeunter Einrichten von Benachrichtigungen oder Auslöseraktionen basierend auf Parameter Store-Ereignissen.
  - Aktualisieren Sie im Abschnitt Parameter Store der Systems Manager-Konsole regelmäßig die Parameterliste, um nach den neuen oder aktualisierten Parameterdetails zu suchen.
  - Verwenden Sie den Befehl GetParameter, um nach dem neuen oder aktualisierten Parameter zu suchen. Beispielsweise mithilfe der AWS Command Line Interface (AWS CLI):

    ```
    aws ssm get-parameter name MyParameter
    ```

    Für einen neuen Parameter wird die Meldung `ParameterNotFound` zurückgegeben, bis der Parameter validiert wurde. Für einen vorhandenen Parameter, den Sie aktualisieren, werden Informationen zur neuen Version erst erfasst, wenn der Parameter validiert wurde.

  Wenn Sie versuchen, den Parameter erneut zu erstellen oder zu aktualisieren, bevor der Validierungsprozess abgeschlossen ist, meldet das System, dass die Validierung noch läuft. Wenn der Parameter nicht erstellt oder aktualisiert wurde, können Sie es fünf Minuten nach dem ersten Versuch erneut versuchen.

AWS Systems Manager Verwaltung von Änderungen

AWS Systems Manager bietet die folgenden Funktionen, um Änderungen an Ihren AWS Ressourcen vorzunehmen.

Themen

- AWS Systems Manager Change Manager
- AWS Systems Manager-Automatisierung
- AWS Systems Manager Change Calendar
- AWS Systems Manager Maintenance Windows

AWS Systems Manager Change Manager

Change Manager, eine Funktion von AWS Systems Manager, ist ein Change-Management-Framework für Unternehmen, mit dem betriebliche Änderungen an Ihrer Anwendungskonfiguration und Infrastruktur angefordert, genehmigt, implementiert und gemeldet werden können. Wenn Sie ein einziges delegiertes Administratorkonto verwenden AWS Organizations, können Sie Änderungen an mehreren Stellen AWS-Konten und übergreifend verwalten. AWS-Regionen Alternativ können Sie mit einem lokalen Konto Änderungen für einen einzigen AWS-Konto verwalten. Wird Change Manager für die Verwaltung von Änderungen sowohl an AWS Ressourcen als auch an lokalen Ressourcen verwendet. Um mit Change Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Change Manager aus.

Mit Change Manager können Sie vorab genehmigte Änderungsvorlagen verwenden, um Änderungsprozesse für Ihre Ressourcen zu automatisieren und unbeabsichtigte Ergebnisse bei betrieblichen Änderungen zu vermeiden. Jede Änderungsvorlage gibt Folgendes an:

- Eine oder mehrere Automation-Runbooks, aus denen ein Benutzer beim Erstellen einer Änderungsanforderung auswählen kann. Die Änderungen an Ihren Ressourcen werden in Automation-Runbooks definiert. Sie können benutzerdefinierte Runbooks oder AWS -verwaltete Runbooks in die von Ihnen erstellten Änderungsvorlagen aufnehmen. Wenn ein Benutzer einen Änderungsantrag erstellt, kann er auswählen, welches der verfügbaren Runbooks in die Anforderung aufgenommen werden soll. Darüber hinaus können Sie Änderungsvorlagen erstellen, mit denen der Benutzer, der die Anforderung stellt, jedes beliebige Runbook in der Änderungsanforderung angeben kann.

- Die Benutzer im Konto, die Änderungsanforderungen überprüfen müssen, die mit dieser Änderungsvorlage vorgenommen wurden.
- Das Amazon Simple Notification Service (Amazon SNS)-Thema, das verwendet wird, um zugewiesene Genehmiger darüber zu informieren, dass ein Änderungsantrag zur Überprüfung bereit ist.
- Der CloudWatch Amazon-Alarm, der zur Überwachung des Runbook-Workflows verwendet wird.
- Das Amazon SNS-Thema, das verwendet wird, um Benachrichtigungen über Statusänderungen für Änderungsanforderungen zu senden, die mit der Änderungsvorlage erstellt werden.
- Die Tags, die auf die Änderungsvorlage angewendet werden sollen, um die Änderungsvorlagen zu kategorisieren und zu filtern.
- Ob aus der Änderungsvorlage erstellte Änderungsanträge ohne Genehmigungsschritt ausgeführt werden können (automatisch genehmigte Anforderungen).

Durch die Integration mitChange Calendar, einer weiteren Funktion von Systems Manager, können Sie Änderungen Change Manager auch sicher implementieren und gleichzeitig Terminkonflikte bei wichtigen Geschäftsereignissen vermeiden. Change ManagerIntegration mit AWS Organizations und Unterstützung AWS IAM Identity Center bei der Verwaltung von Änderungen in Ihrem gesamten Unternehmen von einem einzigen Konto aus unter Verwendung Ihres vorhandenen Identitätsmanagementsystems. Sie können den Änderungsfortschritt von Change Manager überwachen und betriebliche Änderungen in Ihrer gesamten Organisation prüfen und für bessere Transparenz und Rechenschaftspflicht sorgen.

Change Manager ergänzt die Sicherheitskontrollen Ihrer Continuous Integration(CI)-Praktiken und Ihrer Continuous Delivery (CD)-Methodik. Change Manager ist nicht für Änderungen gedacht, die im Rahmen eines automatisierten Release-Prozesses wie einer CI/CD-Pipeline vorgenommen werden, es sei denn, es ist eine Ausnahme oder Genehmigung erforderlich.

Funktionsweise von Change Manager

Wenn eine Standard- oder Notfalländerung benötigt wird, erstellt jemand in der Organisation einen Änderungsantrag, der auf einer der Änderungsvorlagen basiert, die für die Verwendung in Ihrer Organisation oder Ihrem Konto erstellt wurden.

Wenn die angeforderte Änderung manuelle Genehmigungen erfordert, benachrichtigt Change Manager die angegebenen Genehmiger durch eine Amazon SNS Benachrichtigung, dass ein Änderungsantrag zur Überprüfung bereit ist. Sie können Genehmiger für Änderungsanforderungen

in der Änderungsvorlage benennen oder Benutzer Genehmiger in der Änderungsanforderung selbst benennen lassen. Sie können verschiedenen Vorlagen verschiedene Prüfer zuweisen. Ordnen Sie beispielsweise einen Benutzer, eine Benutzergruppe oder eine AWS Identity and Access Management (IAM)-Rolle zu, die Anforderungen für Änderungen an verwalteten Knoten genehmigen muss, und eine andere Benutzer-, Gruppen- oder IAM-Rolle für Datenbankänderungen. Wenn die Änderungsvorlage automatische Genehmigungen zulässt und die Benutzerrichtlinie eines Anforderers dies nicht verbietet, kann der Benutzer das Automation-Runbook für seine Anfrage auch ohne Überprüfungsschritt ausführen (mit Ausnahme von Ereignissen zum Einfrieren von Änderungen).

Für jede Änderungsvorlage können Sie bis zu fünf Genehmigerebenen hinzufügen. Sie können beispielsweise verlangen, dass technische Prüfer eine Änderungsanforderung, die aus einer Änderungsvorlage erstellt wurde, zuerst genehmigen und dann eine zweite Genehmigungsebene von einem oder mehreren Managern anfordern.

Change Manager ist in AWS Systems Manager Change Calendar integriert. Wenn eine angeforderte Änderung genehmigt wird, ermittelt das System zunächst, ob die Anforderung mit anderen geplanten Geschäftsvorgängen in Konflikt steht. Wenn ein Konflikt erkannt wird, kann Change Manager die Änderung blockieren oder zusätzliche Genehmigungen erfordern, bevor Sie den Runbook-Workflow starten. Beispielsweise können Sie Änderungen nur während der Geschäftszeiten erlauben, um sicherzustellen, dass Teams zur Verfügung stehen, um unerwartete Probleme zu verwalten. Für alle Änderungen, die außerhalb dieser Zeiten ausgeführt werden sollen, können Sie eine Genehmigung für die Verwaltung auf höherer Ebene in Form vonChange–Freeze-Genehmigern fordern. Bei Notfalländerungen kann Change Manager den Schritt der Überprüfung von Change Calendar für Konflikte oder Blockieren von Ereignissen überspringen, nachdem eine Änderungsanforderung genehmigt wurde.

Wenn es an der Zeit ist, eine genehmigte Änderung zu implementieren Change Manager führt das Automation-Runbook aus, das in der zugeordneten Änderungsanforderung angegeben ist. Nur die Vorgänge, die in genehmigten Änderungsanforderungen definiert sind, sind zulässig, wenn Runbook-Workflows ausgeführt werden. Dieser Ansatz hilft Ihnen, unbeabsichtigte Ergebnisse zu vermeiden, während Änderungen implementiert werden.

Zusätzlich zum Einschränken der Änderungen, die bei der Ausführung eines Runbook-Workflows vorgenommen werden können, hilft Ihnen Change Manager außerdem beim Steuern der Gleichzeitigkeit und der Fehlerschwellenwerte. Sie legen fest, wie viele Ressourcen ein Runbook-Workflow gleichzeitig ausführen kann, auf wie vielen Konten die Änderung gleichzeitig ausgeführt werden kann und wie viele Fehler vor dem Beenden des Prozesses zugelassen werden und (wenn

das Runbook ein Rollback-Skript enthält) zurückgesetzt werden sollen. Sie können den Fortschritt der vorgenommenen Änderungen auch mithilfe von CloudWatch Alarmen überwachen.

Nachdem ein Runbook-Workflow abgeschlossen wurde, können Sie Details zu den vorgenommenen Änderungen überprüfen. Diese Details beinhalten den Grund für einen Änderungsantrag, welche Änderungsvorlage verwendet wurde, wer die Änderungen angefordert und genehmigt hat und wie die Änderungen implementiert wurden.

Weitere Informationen

Einführung von AWS Systems ManagerChange Manager auf dem AWS -News Blog

Welche Vorteile bietet Change Manager meinen Vorgängen?

Change Manager bietet folgende Vorteile:

- Reduzieren Sie das Risiko von Service-Unterbrechungen und Ausfallzeiten

  Change Manager kann Betriebsänderungen sicherer machen, indem sichergestellt wird, dass nur genehmigte Änderungen implementiert werden, wenn ein Runbook-Workflow ausgeführt wird. Sie können ungeplante und nicht überprüfte Änderungen blockieren. Change Manager hilft Ihnen, die Arten von unbeabsichtigten Ergebnissen zu vermeiden, die durch menschliche Fehler verursacht werden, die kostspielige Zeit an Forschung und Rückverfolgung erfordern.
- Detailliertes Prüfung und Berichterstattung zu Änderungshistorien

  Change Manager bietet Rechenschaftspflicht mit einer konsistenten Möglichkeit, Änderungen, die in Ihrem Unternehmen vorgenommen wurden, zu melden und zu prüfen, die Absicht der Änderungen und Details darüber, wer sie genehmigt und implementiert hat.
- Vermeiden Sie Konflikte oder Verstöße

  Change Manager kann Zeitplankonflikte wie Feiertagsereignisse oder neue Produktstarts basierend auf dem aktiven Änderungskalender für Ihre Organisation erkennen. Sie können die Ausführung von Runbook-Workflows nur während der Geschäftszeiten oder nur mit zusätzlichen Genehmigungen erlauben.
- Anpassung der Änderungsanforderungen an Ihr sich änderndes Geschäft

  In verschiedenen Geschäftsperioden können Sie unterschiedliche Anforderungen an das Änderungsmanagement stellen. Beispielsweise können Sie während der end-of-month

Berichterstattung, in der Steuersaison oder in anderen kritischen Geschäftsperioden Änderungen blockieren oder für Änderungen, die unnötige betriebliche Risiken mit sich bringen könnten, die Genehmigung der Geschäftsleitung einholen.

- Zentrale Verwaltung von Änderungen über Konten hinweg

  Durch die Integration mit Organizations ermöglicht Change Manager Ihnen die Verwaltung von Änderungen in allen Ihren Organisationseinheiten (OUs) von einem einzigen delegierten Administratorkonto aus. Sie können Change Manager für die Verwendung mit Ihrer gesamten Organisation oder nur mit einigen Ihrer Organisationseinheiten aktivieren.

An wen richtet sich Change Manager?

Change Managerist für die folgenden AWS Kunden und Organisationen geeignet:

- Jeder AWS Kunde, der die Sicherheit und Steuerung betrieblicher Änderungen an seinen Cloud- oder lokalen Umgebungen verbessern möchte.
- Organisationen, die die Zusammenarbeit und Transparenz über alle Teams hinweg verbessern, die Anwendungsverfügbarkeit durch Vermeidung von Ausfallzeiten verbessern und das mit manuellen und sich wiederholenden Aufgaben verbundene Risiko verringern möchten.
- Organisationen, die bewährte Methoden für das Änderungsmanagement einhalten müssen.
- Kunden, die eine vollständig überprüfbare Historie der Änderungen an ihrer Anwendungskonfiguration oder -infrastruktur benötigen.

Was sind die Hauptfeatures von Change Manager?

Zu den wichtigsten Features von Change Manager gehört Folgendes:

- Integrierte Unterstützung für bewährte Methoden für das Änderungsmanagement

  Mit Change Manager können Sie ausgewählte bewährte Methoden für das Änderungsmanagement auf Ihre Vorgänge anwenden. Sie können folgende Optionen aktivieren:

  - Überprüfen Sie Change Calendar, um zu sehen, ob Ereignisse derzeit eingeschränkt sind, sodass Änderungen nur während der offenen Kalenderperioden vorgenommen werden.
  - Zulassen von Änderungen bei eingeschränkten Ereignissen mit zusätzlichen Genehmigungen von Change-Freeze-Genehmigungsberechtigten.
  - Erfordern, dass CloudWatch Alarme für alle Änderungsvorlagen angegeben werden.

  - Verlangen Sie, dass alle in Ihrem Konto erstellten Änderungsvorlagen geprüft und genehmigt werden müssen, bevor sie zur Erstellung von Änderungsaufträgen verwendet werden können.
- Verschiede Genehmigungspfade für geschlossene Kalenderperioden und Notänderungsanträge

  Sie können eine Option zulassen, um Change Calendar auf eingeschränkte Ereignisse zu prüfen und genehmigte Änderungsanforderungen zu blockieren, bis das Ereignis abgeschlossen ist. Sie können jedoch auch eine zweite Gruppe von Genehmigern bestimmen, die Change-Freeze-Genehmiger, die eine Änderung auch dann zulassen können, wenn der Kalender geschlossen ist. Sie können auch Notfalländerungsvorlagen erstellen. Änderungsaufträge, die aus einer Notfalländerungsvorlage erstellt wurden, erfordern weiterhin regelmäßige Genehmigungen, unterliegen jedoch keinen Kalenderbeschränkungen und erfordern keine Change-Freeze-Freigaben.
- Steuern Sie, wie und wann Runbook-Workflows gestartet werden

  Runbook-Workflows können nach einem Zeitplan oder nach Abschluss der Genehmigungen gestartet werden (vorbehaltlich der Kalendereinschränkungsregeln).
- Integrierte Unterstützung für Benachrichtigungen

  Geben Sie an, wer in Ihrer Organisation Änderungsvorlagen und Änderungsanforderungen prüfen und genehmigen soll. Weisen Sie einer Änderungsvorlage ein Amazon SNS-Thema zu, um Benachrichtigungen an die Abonnenten des Themas über Statusänderungen für Änderungsanträge zu senden, die mit dieser Änderungsvorlage erstellt wurden.
- Integration mit AWS Systems Manager Change Calendar

  Change Manager ermöglicht es Administratoren, Zeitplanänderungen während bestimmter Zeiträume einzuschränken. Sie können beispielsweise eine Richtlinie erstellen, die Änderungen nur während der Geschäftszeiten zulässt, um sicherzustellen, dass das Team für Probleme verfügbar ist. Sie können Änderungen auch bei wichtigen Geschäftsereignissen einschränken. Beispielsweise können Einzelhandelsunternehmen Änderungen bei großen Verkaufsereignissen einschränken. Sie können auch während eingeschränkter Zeiträume zusätzliche Genehmigungen verlangen.
- Integration mit AWS IAM Identity Center und Active Directory-Unterstützung

  Mit der IAM-Identity-Center-Integration können Mitglieder Ihrer Organisation auf AWS-Konten zugreifen und ihre Ressourcen mithilfe von Systems Manager basierend auf einer gemeinsamen Benutzeridentität verwalten. Mit IAM Identity Center können Sie Ihren Benutzern Zugriff auf Konten über AWS hinweg gewähren.

Durch die Integration mit Active Directory können Benutzer in Ihrem Active Directory-Konto als Genehmiger für Änderungsvorlagen zugewiesen werden, die für Ihre Change Manager-Vorgänge erstellt wurden.

- Integration mit CloudWatch Amazon-Alarmen

  Change Managerist in CloudWatch Alarme integriert. Change Managerlauscht während des Runbook-Workflows auf CloudWatch Alarme und ergreift alle für den Alarm definierten Aktionen, einschließlich des Sendens von Benachrichtigungen.

- Integration mit Lake AWS CloudTrail

  Durch die Einrichtung eines Ereignisdatenspeichers in AWS CloudTrail Lake können Sie überprüfbare Informationen zu den Änderungen einsehen, die durch Änderungsanforderungen vorgenommen wurden, die in Ihrem Konto oder Ihrer Organisation ausgeführt werden. Die gespeicherten Ereignisinformationen enthalten u. a. folgende Details:

  - Die API-Aktionen, die ausgeführt wurden
  - Die für diese Aktionen enthaltenen Anforderungsparameter
  - Der Benutzer, der die Aktion ausgeführt hat
  - Die Ressourcen, die während des Vorgangs aktualisiert wurden

- Integration mit AWS Organizations

  Mit den kontenübergreifenden Funktionen von Organizations können Sie ein delegiertes Administratorkonto für die Verwaltung von Change Manager-Vorgängen in OUs in Ihrer Organisation verwenden. In Ihrem Organizations-Verwaltungskonto können Sie angeben, welches Konto das delegierte Administratorkonto sein soll. Sie können auch steuern, in welchen Ihrer OUs Change Manager verwendet werden kann.

Entstehen Kosten für die Verwendung von Change Manager?

Ja. Change Managerwird auf einer bestimmten pay-per-use Basis berechnet. Sie zahlen nur das, was Sie nutzen. Weitere Informationen finden Sie unter AWS Systems Manager -Preisgestaltung.

Was sind die primären Komponenten von Change Manager?

Change Manager-Komponenten, die Sie zum Verwalten des Änderungsprozesses in Ihrer Organisation oder Ihrem Konto verwenden, umfassen Folgendes:

Delegiertes Administratorkonto

Wenn Sie den Change Manager in einer Organisation verwenden, verwenden Sie ein delegiertes Administratorkonto. Dies ist der AWS-Konto , der als Konto für die Verwaltung von Betriebsaktivitäten in Systems Manager festgelegt ist, einschließlich Change Manager. Das delegierte Administratorkonto verwaltet Änderungsaktivitäten in Ihrer gesamten Organisation. Wenn Sie Ihre Organisation für die Verwendung mit dem Change Manager einrichten, geben Sie an, welche Ihrer Konten in dieser Rolle verwendet werden. Das delegierte Administratorkonto muss das einzige Mitglied der Organisationseinheit (OU) sein, der es zugewiesen ist. Das delegierte Administratorkonto ist nicht erforderlich, wenn Sie es AWS-Konto nur Change Manager mit einem Konto verwenden.

> **Important**
>
> Wenn Sie den Change Manager in einer Organisation verwenden, empfehlen wir, Änderungen immer über das delegierte Administratorkonto vorzunehmen. Obwohl Sie Änderungen von anderen Konten in der Organisation vornehmen, werden diese Änderungen nicht im delegierten Administratorkonto gemeldet oder können nicht angezeigt werden.

Änderungsvorlage

Eine Änderungsvorlage ist eine Sammlung von Konfigurationseinstellungen in Change Manager, die beispielsweise erforderliche Genehmigungen, verfügbare Runbooks und Benachrichtigungsoptionen für Änderungsanforderungen definiert.

Sie können verlangen, dass die von Benutzern in Ihrer Organisation oder Ihrem Konto erstellten Änderungsvorlagen einen Genehmigungsprozess durchlaufen, bevor sie verwendet werden können.

Change Manager unterstützt zwei Arten von Änderungsvorlagen. Bei einer genehmigten Änderungsanforderung, die auf einer Notfalländerungsvorlage basiert, kann die angeforderte Änderung auch dann vorgenommen werden, wenn Sperrereignisse in Change Calendar vorliegen. Bei einer genehmigten Änderungsanforderung, die auf einer standardmäßigen Änderungsvorlage basiert, kann die angeforderte Änderung nicht vorgenommen werden, wenn Blockierungsereignisse in Change Calendar vorhanden sind, es sei denn, es werden zusätzliche Genehmigungen von bestimmten Genehmigern für Change-Freeze-Ereignisse erhalten.

Änderungsanforderung

Eine Änderungsanforderung ist eine Anforderung Change Manager zur Ausführung eines Automatisierungs-Runbooks, das eine oder mehrere Ressourcen in Ihren AWS oder lokalen Umgebungen aktualisiert. Ein Änderungsantrag wird mit einer Änderungsvorlage erstellt.

Wenn Sie eine Änderungsanforderung erstellen, müssen ein oder mehrere Genehmiger in Ihrer Organisation oder Ihrem Konto die Anforderung überprüfen und genehmigen. Ohne die erforderlichen Genehmigungen kann der Runbook-Workflow, der die angeforderten Änderungen anwendet, nicht ausgeführt werden.

Im System sind Änderungsanforderungen eine Art von OpsItem Eingabe. AWS Systems Manager OpsCenter Jedoch werden OpsItems des Typs `/aws/changerequest` nicht in OpsCenter angezeigt. Als OpsItems unterliegen Änderungsanforderungen den gleichen erzwungenen Kontingenten wie andere OpsItems-Typen.

Um eine Änderungsanforderung programmgesteuert zu erstellen, rufen Sie außerdem nicht die `CreateOpsItem`-API-Operation auf. Verwenden Sie anstelle die `StartChangeRequestExecution`-API-Operation. Anstatt sofort ausgeführt zu werden, muss die Änderungsanforderung genehmigt werden, und es dürfen keine blockierenden Ereignisse in Change Calendar vorliegen, um zu verhindern, dass der Workflow ausgeführt wird. Wenn Genehmigungen empfangen wurden und der Kalender nicht gesperrt ist (oder die Berechtigung erteilt wurde, blockierende Kalenderereignisse zu umgehen), kann die `StartChangeRequestExecution`-Aktion abgeschlossen werden.

Runbook-Workflow

Ein Runbook-Workflow ist der Prozess der angeforderten Änderungen, die an den Zielressourcen in Ihrer Cloud oder On-Premises-Umgebung vorgenommen werden. Jede Änderungsanforderung bestimmt ein einziges Automation-Runbook, das zur Durchführung der angeforderten Änderung verwendet werden soll. Der Runbook-Workflow tritt auf, nachdem alle erforderlichen Genehmigungen erteilt wurden und keine Sperrereignisse in Change Calendar vorliegen. Wenn die Änderung für ein bestimmtes Datum und eine bestimmte Uhrzeit geplant wurde, beginnt der Runbook-Workflow erst nach der Planung, selbst wenn alle Genehmigungen eingegangen sind und der Kalender nicht blockiert ist.

Themen

- Einrichten von Change Manager

- Arbeiten mit Change Manager
- Prüfen und Protokollieren von Change Manager-Aktivitäten
- Fehlerbehebung für Change Manager

Einrichten von Change Manager

Sie können Change Manager, eine -Funktion von AWS Systems Manager, nutzen, um Änderungen für eine gesamte Organisation zu verwalten, wie in AWS Organizations oder für einen einzelnen AWS-Konto konfiguriert.

Wenn Sie Change Manager mit einer Organisation verwenden, beginnen Sie mit dem Thema Einrichten von Change Manager für eine Organisation (Management-Konto)und fahren Sie dann mit Konfigurieren von Change Manager-Optionen und bewährten Methoden fort.

Wenn Sie Change Manager mit einem einzelnen Konto verwenden, fahren Sie direkt mit Konfigurieren von Change Manager-Optionen und bewährten Methoden fort.

> **Note**
>
> Wenn Sie mit Nutzung von Change Manager mit einem einzigen Konto beginnen, dieses Konto aber später zu einer Organisationseinheit hinzugefügt wird, für die Change Manager genehmigt ist, werden Ihre individuellen Kontoeinstellungen nicht berücksichtigt.

Themen

- Einrichten von Change Manager für eine Organisation (Management-Konto)
- Konfigurieren von Change Manager-Optionen und bewährten Methoden
- Konfigurieren von Rollen und Berechtigungen für Change Manager
- Steuern des Zugriffs auf Runbook-Workflows für automatische Genehmigung

Einrichten von Change Manager für eine Organisation (Management-Konto)

Die Aufgaben in diesem Thema geltenChange Manager, wenn Sie eine Funktion von AWS Systems Manager, mit einer Organisation verwenden, die in eingerichtet ist AWS Organizations. Wenn Sie es Change Manager nur mit einer einzigen verwenden möchten AWS-Konto, fahren Sie mit dem Thema fortKonfigurieren von Change Manager-Optionen und bewährten Methoden.

Führen Sie die Aufgaben in diesem Abschnitt in einem aus AWS-Konto , der als Verwaltungskonto in Organizations dient. Weitere Informationen zum Verwaltungskonto und zu anderen Organizations-Konzepten finden Sie unter AWS Organizations -Terminologie und Konzepte.

Wenn Sie Organizations aktivieren und Ihr Konto als Verwaltungskonto angeben müssen, bevor Sie fortfahren, siehe Creating and managing an organization (Erstellen und Verwalten einer Organisation) im AWS Organizations -Benutzerhandbuch.

> **Note**
>
> Dieser Einrichtungsvorgang kann in den folgenden Fällen nicht ausgeführt werden AWS-Regionen:
>
> - Europa (Mailand) (eu-south-1)
> - Naher Osten (Bahrain) (me-south-1)
> - Afrika (Kapstadt) (af-south-1)
> - Asien-Pazifik (Hongkong) (ap-east-1)
>
> Stellen Sie sicher, dass Sie für dieses Verfahren in einer anderen Region in Ihrem Verwaltungskonto arbeiten.

Während des Einrichtungsvorgangs führen Sie die folgenden Hauptaufgaben in ausQuick Setup, mit einer Fähigkeit von AWS Systems Manager.

- Aufgabe 1: Registrieren eines delegierten Administrators für Ihre Organisation

  Die änderungsbezogenen Aufgaben, die mit Change Manager ausgeführt werden, werden in einem Ihrer Mitgliedskonten verwaltet, das Sie als delegiertes Administratorkonto angeben. Das delegierte Administratorkonto, das Sie für Change Manager registrieren wird zum delegierten Administratorkonto für alle Systems Manager-Vorgänge. (Möglicherweise haben Sie Administratorkonten für andere delegiert AWS-Services). Ihr delegiertes Administratorkonto für Change Manager, das nicht mit Ihrem Verwaltungskonto identisch ist, verwaltet Änderungsaktivitäten in Ihrer gesamten Organisation, einschließlich Änderungsvorlagen, Änderungsanforderungen und Genehmigungen für jede. Im delegierten Administratorkonto geben Sie auch andere Konfigurationsoptionen für Ihre Change Manager-Operationen an.

> **Important**
>
> Das delegierte Administratorkonto muss das einzige Mitglied der Organisationseinheit (OU) sein, der es in Organizations zugewiesen ist.

- Aufgabe 2: Definieren und Angeben von Runbook-Zugriffsrichtlinien für Änderungsanfordererrollen oder benutzerdefinierte Auftragsfunktionen, die Sie für Ihre Change Manager-Operationen verwenden möchten.

  Um Änderungsanforderungen in erstellen zu könnenChange Manager, müssen Benutzern in Ihren Mitgliedskonten AWS Identity and Access Management (IAM) -Berechtigungen erteilt werden, die es ihnen ermöglichen, nur auf die Automatisierungs-Runbooks und Änderungsvorlagen zuzugreifen, die Sie ihnen zur Verfügung stellen.

  > **Note**
  >
  > Wenn ein Benutzer einen Änderungsantrag erstellt, wählt er zunächst eine Änderungsvorlage aus. Diese Änderungsvorlage stellt möglicherweise mehrere Runbooks zur Verfügung, der Benutzer kann jedoch nur ein Runbook für den jeweiligen Änderungsantrag auswählen. Änderungsvorlagen können auch so konfiguriert werden, dass Benutzer jedes verfügbare Runbook in ihre Anforderungen aufnehmen können.

  Um die erforderlichen Berechtigungen zu erteilen, verwendet Change Manager das Konzept von job functions (Auftragsfunktionen), die auch von IAM verwendet wird. Im Gegensatz zu den AWS -verwalteten Richtlinien für Auftragsfunktionen in IAM geben Sie sowohl die Namen Ihrer Change Manager-Auftragsfunktionen und die IAM-Berechtigungen für diese Auftragsfunktionen an.

  Wenn Sie eine Auftragsfunktion konfigurieren, empfiehlt es sich, eine benutzerdefinierte Richtlinie zu erstellen und nur die Berechtigungen bereitzustellen, die zum Ausführen von Änderungsverwaltungsaufgaben erforderlich sind. Sie können beispielsweise Berechtigungen angeben, die Benutzer basierend auf den von Ihnen definierten Auftragsfunktionen auf diesen bestimmten Satz von Runbooks beschränken.

  Sie können beispielsweise eine Auftragsfunktion mit dem Namen `DBAdmin` erstellen. Für diese Auftragsfunktion können Sie nur Berechtigungen erteilen, die für Runbooks erforderlich sind, die

sich auf Amazon DynamoDB-Datenbanken beziehen, z. B. `AWS-CreateDynamoDbBackup` und `AWSConfigRemediation-DeleteDynamoDbTable`.

Als weiteres Beispiel möchten Sie einigen Benutzern möglicherweise nur die Berechtigungen erteilen, die zum Arbeiten mit Runbooks im Zusammenhang mit Amazon Simple Storage Service (Amazon S3)-Buckets erforderlich sind, z. B. `AWS-ConfigureS3BucketLogging`und `AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock`.

Der Konfigurationsprozess in Quick Setup für Change Manager stellt außerdem eine Reihe vollständiger Administratorberechtigungen für Systems Manager zur Verfügung, die Sie auf eine von Ihnen erstellte Administratorrolle anwenden können.

Jede Change Manager Quick Setup-Konfiguration, die Sie bereitstellen, erstellt eine Auftragsfunktion in Ihrem delegierten Administratorkonto mit Berechtigungen zum Ausführen von Change Manager-Vorlagen und Automation-Runbooks in den von Ihnen ausgewählten Organisationseinheiten. Sie können bis zu 15 Quick Setup-Konfigurationen für Change Manager erstellen.

- Aufgabe 3: Wählen Sie aus, welche Mitgliedskonten in Ihrer Organisation mit Change Manager verwendet werden sollen

  Sie können Change Manager mit allen Mitgliedskonten in allen Organisationseinheiten verwenden, die in Organizations eingerichtet sind, und in allen AWS-Regionen in denen sie arbeiten. Wenn Sie möchten, können Sie stattdessen Change Manager mit nur einigen Ihrer Organisationseinheiten verwenden.

> **Important**
>
> Bevor Sie mit diesem Verfahren beginnen, empfehlen wir dringend, die Schritte zu lesen, um die von Ihnen vorgenommenen Konfigurationsoptionen und die Berechtigungen zu verstehen, die Sie erteilen. Planen Sie insbesondere die benutzerdefinierten Auftragsfunktionen, die Sie erstellen, und die Berechtigungen, die Sie jeder Auftragsfunktion zuweisen. Dadurch wird sichergestellt, dass, wenn Sie später die von Ihnen erstellten Auftragsfunktionsrichtlinien an einzelne Benutzer, Benutzergruppen oder IAM-Rollen anhängen, ihnen nur die Berechtigungen erteilt werden, die Sie für diese beabsichtigen.
> Es hat sich bewährt, zunächst das delegierte Administratorkonto mit dem Anmeldenamen eines Administrators einzurichten. AWS-Konto Konfigurieren Sie dann Auftragsfunktionen

und deren Berechtigungen, nachdem Sie Änderungsvorlagen erstellt und die Runbooks identifiziert haben, die jedes einzelne verwendet.

Um Change Manager für die Verwendung mit einer Organisation einzurichten, führen Sie die folgende Aufgabe im Quick Setup-Bereich der Systems Manager Konsole aus.

Sie wiederholen diese Aufgabe für jede Auftragsfunktion, die Sie für Ihre Organisation erstellen möchten. Jede Auftragsfunktion, die Sie erstellen, kann Berechtigungen für einen anderen Satz von Organisationseinheiten haben.

So richten Sie eine Organisation für Change Manager im Organizations-Verwaltungskonto ein

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Change Manager-Registerkarte Create (Erstellen) aus.
4. Geben Sie für Delegiertes Administratorkonto die ID des AWS-Konto ein, das Sie zum Verwalten von Änderungsvorlagen, Änderungsanforderungen und Runbook-Workflows in Change Manager verwenden möchten.

   Wenn Sie zuvor ein delegiertes Administratorkonto für Systems Manager angegeben haben, wird seine ID bereits in diesem Feld gemeldet.

   > **Important**
   >
   > Das delegierte Administratorkonto muss das einzige Mitglied der Organisationseinheit (OU) sein, der es in Organizations zugewiesen ist.
   > Wenn das delegierte Administratorkonto, das Sie registrieren, später von dieser Rolle abgemeldet wird, entfernt das System seine Berechtigungen für die gleichzeitige Verwaltung von Systems Manager-Vorgängen. Denken Sie daran, dass es notwendig sein wird, dass Sie zum Quick Setup zurückkehren, ein anderes delegiertes Administratorkonto festlegen, und alle Auftragsfunktionen und -Berechtigungen erneut angeben.
   > Wenn Sie den Change Manager in einer Organisation verwenden, empfehlen wir, Änderungen immer über das delegierte Administratorkonto vorzunehmen. Obwohl Sie Änderungen von anderen Konten in der Organisation vornehmen, werden diese

Änderungen nicht im delegierten Administratorkonto gemeldet oder können nicht angezeigt werden.

5. Im Bereich Berechtigungen zum Anfordern und Vornehmen von Änderungen gehen Sie wie folgt vor.

> **Note**
>
> Jede von Ihnen erstellte Bereitstellungskonfiguration stellt die Berechtigungsrichtlinie für nur eine Auftragsfunktion bereit. Sie können zum Quick Setup zurückkehren, um weitere Auftragsfunktionen zu erstellen, wenn Sie Änderungsvorlagen zur Verwendung in Ihren Vorgängen erstellt haben.

So erstellen Sie eine Administratorrolle - Für eine Administratorauftragsfunktion, die IAM-Berechtigungen für alle AWS -Aktionen hat, gehen Sie wie folgt vor.

> **Important**
>
> Das Erteilen von vollständigen Administratorberechtigungen sollte sparsam und nur dann erfolgen, wenn für die Rollen der vollständige Zugriff auf Systems Manager erforderlich ist. Wichtige Informationen zu Sicherheitsüberlegungen für den Zugriff auf Systems Manager finden Sie unter Identity and Access Management für AWS Systems Manager und Bewährte Methoden für die Sicherheit für Systems Manager.

1. Für Auftragsfunktion geben Sie einen Namen zur Identifizierung dieser Rolle und ihrer Berechtigungen ein, z. B. **`My AWS Admin`**.
2. Für die Option Rolle und Berechtigungen wählen Sie Administratorberechtigungen.

So erstellen Sie andere Auftragsfunktionen - Gehen Sie wie folgt vor, um eine nicht-administrative Rolle zu erstellen:

1. Geben Sie für Auftragsfunktion einen Namen ein, um diese Rolle zu identifizieren und ihre Berechtigungen vorzuschlagen. Der von Ihnen gewählte Name sollte den Bereich der Runbooks repräsentieren, für die Sie Berechtigungen erteilen werden, z. B. `DBAdmin` oder `S3Admin`.

2. Für die Option Rolle und Berechtigungen wählen Sie Benutzerdefinierte Berechtigungen.
3. Geben Sie im Editor Berechtigungsrichtlinie die IAM-Berechtigungen im JSON-Format ein, die dieser Auftragsfunktion gewährt werden sollen.

> **Tip**
>
> Es wird empfohlen, dass Sie den IAM-Richtlinien-Editor verwenden, um Ihre Richtlinie zu erstellen und dann den Richtlinien-JSON-Code in das Feld Berechtigungsrichtlinie kopieren.

Beispielrichtlinie: DynamoDB-Datenbankverwaltung

Sie könnten zum Beispiel mit Richtlinieninhalten beginnen, die Berechtigungen für die Arbeit mit den Systems Manager-Dokumenten (SSM-Dokumenten) vorsehen, auf die die Auftragsfunktion Zugriff benötigt. Hier ist ein Beispiel für einen Richtlinieninhalt, der Zugriff auf alle AWS verwalteten Automation-Runbooks gewährt, die sich auf DynamoDB-Datenbanken beziehen, sowie auf zwei Änderungsvorlagen AWS-Konto `123456789012`, die im Beispiel in der Region USA Ost (Ohio) erstellt wurden (). `us-east-2`

Die Richtlinie enthält auch die Berechtigung für die StartChangeRequestExecution Operation, die für die Erstellung eines Änderungsantrags in Change Calendar erforderlich ist.

> **Note**
>
> Dieses Beispiel ist nicht umfassend. Für die Arbeit mit anderen AWS Ressourcen wie Datenbanken und Knoten sind möglicherweise zusätzliche Berechtigungen erforderlich.

```
{
    "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:123456789012:document/MyFirstDBChangeTemplate",
                "arn:aws:ssm:region:123456789012:document/MySecondDBChangeTemplate"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ssm:ListDocuments",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:StartChangeRequestExecution",
            "Resource": "arn:aws:ssm:region:123456789012:automation-definition/*:*"
        }
    ]
}
```

Weitere Informationen zu IAM-Richtlinien finden Sie unter Zugriffsverwaltung für AWS - Ressourcen und Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch

6. Im Bereich Targets wählen Sie aus, ob Sie der gesamten Organisation oder nur einigen Organisationseinheiten Berechtigungen für die Auftragsfunktion gewähren möchten, die Sie erstellen.

   Fahren Sie mit Schritt 9 fort, wenn Sie Ganze Organisation wählen.

Fahren Sie mit Schritt 8 fort, wenn Sie Benutzerdefiniert wählen.

7. Wählen Sie im Bereich Ziel-OUs die Kontrollkästchen der Organisationseinheiten, die mit Change Manager verwendet werden sollen.
8. Wählen Sie Erstellen.

Nachdem das System die Einrichtung von Change Manager für Ihre Organisation abgeschlossen hat, wird eine Zusammenfassung Ihrer Bereitstellungen angezeigt. Diese zusammenfassenden Informationen enthalten den Namen der Rolle, die für die von Ihnen konfigurierte Jobfunktion erstellt wurde. z. B. `AWS-QuickSetup-SSMChangeMgr-DBAdminInvocationRole`.

> **Note**
>
> Quick Setupverwendet AWS CloudFormation StackSets , um Ihre Konfigurationen bereitzustellen. Sie können auch Informationen zu einer abgeschlossenen Bereitstellungskonfiguration in der AWS CloudFormation -Konsole einsehen. Weitere Informationen zu StackSets finden Sie unter Arbeiten mit AWS CloudFormation StackSets im AWS CloudFormation Benutzerhandbuch.

Im nächsten Schritt konfigurieren Sie zusätzliche Change Manager-Optionen. Sie können diese Aufgabe entweder in Ihrem delegierten Administratorkonto oder in einem beliebigen Konto in einer Organisationseinheit ausführen, das Sie für die Verwendung mit Change Manager zugelassen haben. Sie konfigurieren Optionen, wie z. B. die Auswahl einer Option für die Verwaltung der Benutzeridentität, die Festlegung, welche Benutzer Änderungsvorlagen und -Anfragen prüfen und genehmigen oder ablehnen können, und die Auswahl der Optionen zu bewährten Methoden, die für Ihr Unternehmen zulässig sein sollen. Weitere Informationen finden Sie unter Konfigurieren von Change Manager-Optionen und bewährten Methoden.

Konfigurieren von Change Manager-Optionen und bewährten Methoden

Die Aufgaben in diesem Abschnitt müssen unabhängig davon ausgeführt werdenChange Manager, ob Sie eine Funktion von AWS Systems Manager, unternehmensweit oder in einer einzelnen Organisation verwenden AWS-Konto.

Wenn Sie Change Manager für eine Organisation verwenden, können Sie die folgenden Aufgaben entweder in Ihrem delegierten Administratorkonto oder in einem beliebigen Konto in einer

Organisationseinheit durchführen, das Sie für die Verwendung mit Change Manager zugelassen haben.

Themen

- Aufgabe 1: Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern
- Aufgabe 2: Konfigurieren von Change Manager Change-Freeze-Ereignisgenehmigern und bewährten Methoden
- Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen

Aufgabe 1: Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern

Führen Sie die Aufgabe in diesem Verfahren beim ersten Zugriff auf Change Manager aus. Sie können diese Konfigurationseinstellungen später aktualisieren, indem Sie zu Change Manager zurückkehren und Bearbeiten in der Registerkarte Einstellungen auswählen.

Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern

1. Melden Sie sich bei der an AWS Management Console.

   Wenn Sie Change Manager für eine Organisation verwenden, melden Sie sich mit Ihren Anmeldeinformationen für Ihr delegiertes Administratorkonto an. Das Benutzerkonto, das Sie verwenden, muss über die erforderlichen AWS Identity and Access Management (IAM)-Berechtigungen zum Vornehmen von Aktualisierungen an Ihren Change Manager-Einstellungen verfügen.
2. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Change Manager aus.
4. Führen Sie auf der Startseite des Dienstes je nach den verfügbaren Optionen einen der folgenden Schritte aus:

   - Wenn Sie Change Manager mit verwenden AWS Organizations , wählen Sie Delegiertes Konto einrichten.
   - Wenn Sie Change Manager mit einem einzigen verwenden AWS-Konto, wählen Sie Einrichten Change Manager aus.

   –oder–

Klicken Sie auf Beispieländerungsanforderung erstellen, Überspringen und wählen Sie dann die Registerkarte Einstellungen.

5. Wählen Sie für Verwaltung der Benutzeridentität eine der folgenden Optionen.

   - AWS Identity and Access Management (IAM) — Identifizieren Sie die Benutzer, die Anfragen stellen und genehmigen und andere Aktionen ausführen, Change Manager indem Sie Ihre vorhandenen Benutzer, Gruppen und Rollen verwenden.
   - AWS IAM Identity Center (IAM Identity Center) — Erlauben Sie IAM Identity Center, Identitäten zu erstellen und zu verwalten, oder stellen Sie eine Verbindung zu Ihrer vorhandenen Identitätsquelle her, um die Benutzer zu identifizieren, die Aktionen in ausführen. Change Manager

6. Geben Sie im Abschnitt Template reviewer notification (Benachrichtigung für Vorlagenprüfer) die Amazon Simple Notification Service (Amazon SNS)-Themen an, die verwendet werden sollen, um Vorlagenprüfer darüber zu informieren, dass eine neue Änderungsvorlage oder Änderungsvorlagenversion zur Überprüfung bereit ist. Stellen Sie sicher, dass das von Ihnen ausgewählte Amazon SNS-Thema so konfiguriert ist, dass Benachrichtigungen an Ihre Vorlagenprüfer gesendet werden.

   Informationen zum Erstellen und Konfigurieren von Amazon SNS-Themen für Änderungsvorlagenprüferbenachrichtigungen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

   1. Wählen Sie eine der folgenden Optionen aus, um das Amazon SNS-Thema für die Benachrichtigung der Vorlagenprüfer anzugeben:

      - Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.
      - Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen AWS-Konto. (Diese Option ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)

      > **Note**
      >
      > Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet

werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

2. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

7. Wählen Sie im Abschnitt Änderungsvorlagenprüfer die Benutzer in Ihrer Organisation oder Ihrem Konto aus, um neue Änderungsvorlagen zu überprüfen oder Vorlagenversionen zu ändern, bevor sie in Ihren Vorgängen verwendet werden können.

   Änderungsvorlagenprüfer sind dafür verantwortlich, die Eignung und Sicherheit von Vorlagen zu überprüfen, die andere Benutzer zur Verwendung in Change Manager-Runbook-Workflows eingereicht haben.

   Wählen Sie die Änderungsvorlagenprüfer folgendermaßen aus:

   1. Wählen Sie Hinzufügen aus.
   2. Aktivieren Sie das Kontrollkästchen neben dem Namen aller Benutzer, Gruppen oder IAM-Rollen, die Sie als Änderungsvorlagenprüfer zuweisen möchten.
   3. Wählen Sie Add approvers (Hinzufügen von Genehmigern).

8. Wählen Sie Absenden aus.

Nachdem Sie diese erste Einrichtung abgeschlossen haben, konfigurieren Sie zusätzliche Change Manager-Einstellungen und bewährte Methoden, indem Sie die Schritte unterAufgabe 2: Konfigurieren von Change Manager Change-Freeze-Ereignisgenehmigern und bewährten Methoden befolgen.

Aufgabe 2: Konfigurieren von Change Manager Change-Freeze-Ereignisgenehmigern und bewährten Methoden

Nachdem Sie die Schritte unter Aufgabe 1: Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern abgeschlossen haben, können Sie zusätzliche Änderungsanforderungsprüfer während Change–Freeze-Ereignissen bestimmen und angeben, welche verfügbaren bewährten Methoden Sie für Ihre Change Manager-Vorgänge zulassen wollen.

Ein Ereignis zum Einfrieren von Änderungen bedeutet, dass Einschränkungen im aktuellen Änderungskalender gelten (der Kalenderstatus AWS Systems Manager Change Calendar ist`CLOSED`). In diesen Fällen müssen zusätzlich zu den regulären Genehmigern für

Änderungsanforderungen oder wenn die Änderungsanforderung mit einer Vorlage erstellt wurde, die automatische Genehmigungen zulässt, die Genehmiger des Änderungsstopps die Genehmigung für die Ausführung dieser Änderungsanforderung erteilen. Wenn dies nicht der Fall ist, wird die Änderung erst verarbeitet, wenn der Kalenderstatus wieder `OPEN` ist.

Konfigurieren von Change Manager Change-Freeze-Ereignisgenehmigern und bewährten Methoden

1. Wählen Sie im Navigationsbereich Change Manager aus.
2. Wählen Sie die Registerkarte Einstellungen und anschließend Bearbeiten.
3. Wählen Sie im Abschnitt Genehmiger für Change-Freeze-Ereignisse die Benutzer in Ihrer Organisation oder Ihrem Konto aus, die Änderungen genehmigen können, die ausgeführt werden, selbst wenn der verwendete Kalender in Change Calendar derzeit GESCHLOSSEN ist.

   > **Note**
   >
   > Um Change-Freeze-Überprüfungen zu erlauben, müssen Sie das Kontrollkästchen für die Option Änderungskalender auf eingeschränkte Änderungsereignisse prüfen in Bewährte Methoden aktivieren.

   Wählen Sie Genehmiger für Change–Freeze-Ereignisse aus, indem Sie die folgenden Schritte ausführen:

   1. Wählen Sie Hinzufügen aus.
   2. Aktivieren Sie das Kontrollkästchen neben dem Namen aller Benutzer, Gruppen oder IAM-Rollen, die Sie als Genehmiger für Change-Freeze-Ereignisse zuweisen möchten.
   3. Wählen Sie Add approvers (Hinzufügen von Genehmigern).
4. Aktivieren Sie im Abschnitt Bewährte Methoden unten auf der Seite die bewährten Methoden, die Sie für jede der folgenden Optionen erzwingen möchten.

   - Option:Änderungskalender auf eingeschränkte Änderungsereignisse prüfen

     Um anzugeben, dass Change Manager einen Kalender in Change Calendar überprüft, um sicherzustellen, dass Änderungen nicht durch geplante Ereignisse blockiert werden, wählen Sie zunächst das Kontrollkästchen Enabled und wählen Sie dann den Kalender aus, um auf eingeschränkte Ereignisse der Änderungskalender-Liste zu prüfen.

     Mehr über Change Calendar erfahren Sie unter AWS Systems Manager Change Calendar.

- Option: SNS-Thema für Genehmiger für geschlossene Ereignisse

  1. Wählen Sie eine der folgenden Optionen aus, um das Amazon Simple Notification Service (Amazon SNS)-Thema in Ihrem Konto anzugeben, das für das Senden von Benachrichtigungen an Genehmiger während der Change-Freeze-Ereignisse verwendet werden soll. (Beachten Sie, dass Sie Genehmiger auch im Abschnitt Genehmiger für Change-Freeze-Ereignisse über Bewährte Methoden angeben müssen.)

     - Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.
     - Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen AWS-Konto. (Diese Option ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)

     > **Note**
     >
     > Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

  2. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

- Option: Überwachungen für alle Vorlagen erforderlich

  Wenn Sie sicherstellen möchten, dass alle Vorlagen für Ihre Organisation oder Ihr Konto einen CloudWatch Amazon-Alarm zur Überwachung Ihres Änderungsvorgangs angeben, aktivieren Sie das Kontrollkästchen Aktiviert.

- Option: Überprüfung und Genehmigung der Vorlage vor der Verwendung erforderlich

  Um sicherzustellen, dass keine Änderungsanforderungen erstellt und keine Runbook-Workflows ausgeführt werden, ohne auf einer Vorlage basieren zu müssen, die überprüft und genehmigt wurde, aktivieren Sie das Kontrollkästchen Enabled.

5. Klicken Sie auf Speichern.

Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen

Sie können Change Manager, eine Funktion von AWS Systems Manager, konfigurieren, wenn Sie Benachrichtigungen zu einem Amazon Simple Notification Service (Amazon SNS)-Thema für Ereignisse im Zusammenhang mit Änderungsanforderungen und Änderungsvorlagen senden möchten. Führen Sie die folgenden Aufgaben aus, um Benachrichtigungen für die Change Manager-Ereignisse zu erhalten, denen Sie ein Thema hinzufügen.

Themen

- Aufgabe 1: Erstellen und Abonnieren eines Amazon SNS-Themas
- Aufgabe 2: Aktualisieren der Amazon SNS-Zugriffsrichtlinie
- Aufgabe 3: (Optional) Aktualisieren der AWS Key Management Service-Zugriffsrichtlinie

Aufgabe 1: Erstellen und Abonnieren eines Amazon SNS-Themas

Zunächst müssen Sie ein Amazon SNS-Thema erstellen und abonnieren. Weitere Informationen finden Sie unter Erstellen eines Amazon-SNS-Themas und Abonnieren eines Amazon-SNS-Themas im Entwicklerhandbuch zu Amazon Simple Notification Service.

> **Note**
>
> Um Benachrichtigungen zu erhalten, müssen Sie den Amazon-Ressourcennamen (ARN) eines Amazon SNS-Themas angeben, das sich in derselben AWS-Region und demselben AWS-Konto wie das delegierte Administratorkonto befindet.

Aufgabe 2: Aktualisieren der Amazon SNS-Zugriffsrichtlinie

Gehen Sie wie folgt vor, um die Amazon SNS-Zugriffsrichtlinie zu aktualisieren, damit Systems Manager Change Manager-Benachrichtigungen für das Amazon SNS-Thema veröffentlichen kann, das Sie in Aufgabe 1 erstellt haben. Ohne Fertigstellung dieser Aufgabe hat Change Manager keine Berechtigung zum Senden von Benachrichtigungen für die Ereignisse, für die Sie das Thema hinzufügen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home.
2. Wählen Sie im Navigationsbereich Topics (Themen) aus.

3. Wählen Sie das Thema aus, das Sie in Aufgabe 1 erstellt haben und klicken Sie dann auf Edit (Bearbeiten).
4. Erweitern Sie Access policy (Zugriffsrichtlinie).
5. Aktualisieren und fügen Sie den folgenden `Sid`-Block der vorhandenen Richtlinie hinzu und ersetzen Sie jeden *`Platzhalter für Benutzereingabe`* mit Ihren eigenen Informationen.

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

   Geben Sie diesen Block nach dem vorhandenen `Sid`-Block ein und ersetzen Sie *`region`*, *`account-id`* und *`topic-name`* durch die geeigneten Werte für das von Ihnen erstellte Thema.
6. Wählen Sie Save Changes.

Das System sendet jetzt Benachrichtigungen an das Amazon SNS-Thema, wenn der Ereignistyp auftritt, den Sie dem Thema hinzufügen.

> **Important**
>
> Wenn Sie das Amazon SNS-Thema mit einem serverseitigen AWS Key Management Service (AWS KMS) Verschlüsselungsschlüssel konfiguriert haben, müssen Sie Aufgabe 3 ausführen.

Aufgabe 3: (Optional) Aktualisieren der AWS Key Management Service-Zugriffsrichtlinie

Wenn Sie die serverseitige AWS Key Management Service (AWS KMS)-Verschlüsselung für Ihr Amazon SNS-Thema aktiviert haben, müssen Sie auch die Zugriffsrichtlinie des AWS KMS key aktualisieren, den Sie bei der Konfiguration des Themas ausgewählt haben. Gehen Sie wie folgt vor, um die Zugriffsrichtlinie zu aktualisieren, damit Systems Manager Change Manager-Genehmigungsbenachrichtigungen für das Amazon SNS-Thema veröffentlichen kann, das Sie in Aufgabe 1 erstellt haben.

1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
3. Wählen Sie die ID des Kundenmasterschlüssels aus, den Sie bei der Erstellung des Themas ausgewählt haben.
4. Wählen Sie im Abschnitt Key policy (Schlüsselrichtlinie) die Option Switch to policy view (Zur Richtlinienansicht wechseln) aus.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Geben Sie den folgenden `Sid`-Block nach einem der vorhandenen `Sid`-Blöcke in die vorhandene Richtlinie ein. Ersetzen Sie jedes *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

```
{
    "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. Geben Sie nun den folgenden `Sid`-Block nach einem der vorhandenen `Sid`-Blöcke in die Ressourcenrichtlinie ein, um zu verhindern, dass das Problem des dienstübergreifenden verwirrten Stellvertreters auftritt.

   Dieser Block verwendet die globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount`, um die Berechtigungen einzuschränken, die Systems Manager der Ressource einem anderen Dienst erteilt.

   Ersetzen Sie jedes *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

```
{
  "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:region:account-id:topic-name",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:ssm:region:account-id:*"
        },
        "StringEquals": {
          "aws:SourceAccount": "account-id"
        }
      }
    }
  ]
}
```

8. Wählen Sie Save Changes.

Konfigurieren von Rollen und Berechtigungen für Change Manager

Standardmäßig hat Change Manager keine Berechtigung zur Ausführung von Aktionen auf Ihre Ressourcen. Sie müssen den Zugriff mithilfe einer AWS Identity and Access Management (IAM-) Servicerolle gewähren oder eine Rolle übernehmen. Diese Rolle ermöglicht es Change Manager, die in einer genehmigten Änderungsanforderung angegebenen Runbook-Workflows in Ihrem Namen sicher auszuführen. Die Rolle gewährt AWS Security Token Service (AWS STS) AssumeRoleVertrauen fürChange Manager.

Durch die Bereitstellung dieser Berechtigungen für eine Rolle, um im Namen von Benutzern in einer Organisation zu handeln, muss Benutzern dieses Array von Berechtigungen nicht selbst gewährt werden. Die durch die Berechtigungen zulässigen Aktionen sind nur auf genehmigte Vorgänge beschränkt.

Wenn Benutzer in Ihrem Konto oder Ihrer Organisation eine Änderungsanforderung erstellen, können sie diese Übernahmerolle auswählen, um die Änderungsvorgänge auszuführen.

Sie können eine neue Übernahmerolle für Change Manager erstellen oder eine vorhandene Rolle mit den erforderlichen Berechtigungen aktualisieren.

Wenn Sie eine Servicerolle für Change Manager erstellen müssen, führen Sie die folgenden Schritte aus.

Aufgaben

- Aufgabe 1: Erstellen einer Übernahmerollenrichtlinie für Change Manager
- Aufgabe 2: Erstellen einer Übernahmerolle für Change Manager
- Aufgabe 3: Anfügen der iam:PassRole-Richtlinie an andere Rollen
- Aufgabe 4: Hinzufügen von Inline-Richtlinien zu einer übernommenen Rolle, um andere aufzurufen AWS-Services
- Aufgabe 5: Konfigurieren des Benutzerzugriffs auf Change Manager

Aufgabe 1: Erstellen einer Übernahmerollenrichtlinie für Change Manager

Verwenden Sie das folgende Verfahren, um die Richtlinie zu erstellen, die Sie an Ihre Change Manager-Übernahmerolle anfügen.

Erstellen einer Übernahmerollenrichtlinie für Change Manager

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

2. Wählen Sie im Navigationsbereich Policies und dann Create Policy.
3. Wählen Sie auf der Seite Create policy (Richtlinie erstellen) die Registerkarte JSON aus und ersetzen Sie den Standardinhalt durch folgenden, den Sie in den folgenden Schritten für Ihre eigenen Change Manager-Vorgänge ändern.

> **Note**
>
> Wenn Sie eine Richtlinie für ein einzelnes AWS-Konto Konto und nicht für eine Organisation mit mehreren Konten erstellen AWS-Regionen, können Sie den ersten Anweisungsblock weglassen. Die `iam:PassRole`-Berechtigung ist nicht erforderlich, wenn ein einziges Konto Change Manager verwendet.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::delegated-admin-account-id: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:region:account-id:automation-definition/template-name:
$DEFAULT",
                "arn:aws:ssm:region::document/template-name"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "ssm:ListOpsItemEvents",
                "ssm:GetOpsItem",
                "ssm:ListDocuments",
                "ssm:DescribeOpsItems"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Aktualisieren Sie den `Resource`-Wert für die `iam:PassRole`-Aktion, um die ARNs aller für Ihre Organisation definierten Auftragsfunktionen einzuschließen, denen Sie Berechtigungen zum Initiieren von Runbook-Workflows erteilen möchten.
5. Ersetzen Sie die Platzhalter *`region`*, *`account-id`*, *`template-name`*, *`delegated-admin-account-id`* und *`job-function`* mit Werten für Ihre Change Manager-Vorgänge.
6. Ändern Sie für die zweite `Resource`-Anweisung die Liste so, dass sie alle Änderungsvorlagen enthält, für die Sie Berechtigungen erteilen möchten. Alternativ können Sie `"Resource": "*"` angeben, um Berechtigungen für alle Änderungsvorlagen in Ihrer Organisation zu erteilen.
7. Wählen Sie Weiter: Markierungen.
8. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern.
9. Wählen Sie Weiter: Prüfen aus.
10. Geben Sie auf der Seite Review policy (Richtlinie überprüfen) im Feld Name einen Namen ein, wie z. B **`MyChangeManagerAssumeRole`**, und geben Sie anschließend eine optionale Beschreibung ein.
11. Klicken Sie auf Create policy (Richtlinie erstellen) und fahren Sie mit Aufgabe 2: Erstellen einer Übernahmerolle für Change Manager fort.

Aufgabe 2: Erstellen einer Übernahmerolle für Change Manager

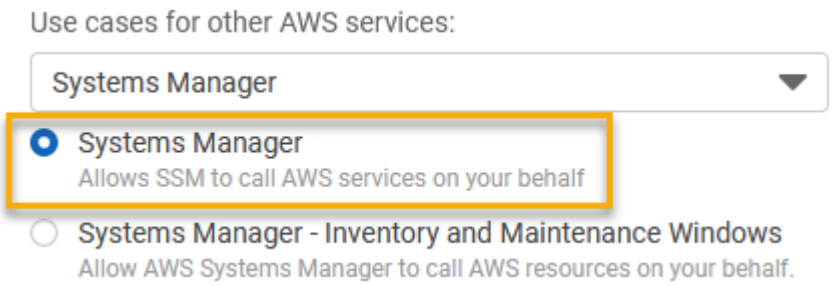
Führen Sie die folgenden Schritte zum Erstellen einer Change Manager-Übernahmerolle, ein Art von Servicerolle, für Change Manager aus.

Erstellen einer Übernahmerolle für Change Manager

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

2. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
3. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die folgenden Optionen:

   1. Wählen Sie unter Trusted entity type (Typ der vertrauenswürdigen Entität) die Option AWS - Service
   2. Für Anwendungsfälle für andere AWS-Services wählen Sie Systems Manager
   3. Wählen Sie Systems Manager, wie im folgenden Image gezeigt.



4. Wählen Sie Weiter aus.
5. Suchen Sie auf der Seite Attached permissions policy (Richtlinie für angehängte Berechtigungen) nach der Übernahmerollenrichtlinie, die Sie in Aufgabe 1: Erstellen einer Übernahmerollenrichtlinie für Change Manager erstellt haben, wie beispielsweise **`MyChangeManagerAssumeRole`**.
6. Aktivieren Sie das Kontrollkästchen neben dem Namen der Übernahmerollenrichtlinie und wählen Sie anschließend Next: Tags (Weiter: Tags) aus.
7. Geben Sie unter Role name (Rollenname) einen Namen für Ihr neues Instance-Profil ein, wie z. B. **`MyChangeManagerAssumeRole`**.
8. (Optional) Aktualisieren Sie für Description (Beschreibung) die Beschreibung für diese Instance-Rolle.
9. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, zu verfolgen oder zu steuern.
10. Wählen Sie Weiter: Prüfen aus.
11. (Optional) Fügen Sie für Tags ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, nachzuverfolgen oder zu steuern, und wählen Sie dann Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Roles (Rollen) zurück.
12. Wählen Sie Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Roles (Rollen) zurück.

13. Wählen Sie auf der Seite Roles (Rollen) die gerade erstellte Rolle aus, um die Seite Summary (Übersicht) zu öffnen.

Aufgabe 3: Anfügen der **`iam:PassRole`**-Richtlinie an andere Rollen

Gehen Sie wie nachfolgend beschrieben vor, um die `iam:PassRole`-Richtlinie an ein IAM-Instance-Profil oder eine IAM-Servicerolle anzuhängen. (Der Systems-Manager-Dienst verwendet IAM-Instance-Profile für die Kommunikation mit EC2-Instances. Für Nicht-EC2-verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung wird stattdessen eine IAM-Servicerolle verwendet.)

Durch Anfügen der `iam:PassRole`-Richtlinie, kann der Change Manager-Service Übernahmerollenberechtigungen anderen Services oder Systems-Manager-Funktionen übergeben, wenn Runbook-Workflows ausgeführt werden.

Fügen Sie die **`iam:PassRole`**-Richtlinie an ein IAM-Instance-Profil oder eine Servicerolle wie folgt an

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Suchen Sie nach der Change Manager-Übernahmerolle, die Sie erstellt haben, wie z. B **`MyChangeManagerAssumeRole`**, und wählen Sie seinen Namen aus.
4. Wählen Sie auf der Seite Summary (Zusammenfassung) für die gerade erstellte Rolle die Registerkarte Permissions (Berechtigungen) aus.
5. Wählen Sie Add permissions, Create inline policy (Berechtigungen hinzufügen, eingebundene Richtlinie erstellen).
6. Wählen Sie auf der Seite Create policy die Registerkarte Visual editor aus.
7. Wählen Sie Service (Service) und anschließend die Option IAM aus.
8. Geben Sie im Textfeld Aktionen filtern die PassRoleOption ein**`PassRole`**, und wählen Sie sie aus.
9. Erweitern Sie Resources (Ressourcen). Stellen Sie sicher, dass Specific ausgewählt ist und wählen Sie dann Add ARN aus.
10. Geben Sie im Feld Specify ARN for role (ARN für Rolle angeben) den ARN der IAM-Instance-Profilrolle oder der IAM-Servicerolle ein, an die Sie Übernahmerollenberechtigungen übergeben möchten. Das System füllt die Felder Account (Konto) und Role name with path (Rollenname mit Pfad) automatisch aus.
11. Wählen Sie Hinzufügen aus.

12. Wählen Sie Richtlinie prüfen.
13. Geben Sie für Name einen Namen ein, um diese Richtlinie zu identifizieren und wählen Sie dann Create poliy (Richtlinie erstellen) aus.

Weitere Informationen

- Konfigurieren Sie die für Systems Manager erforderlichen Instanzberechtigungen
- Erstellen Sie die für Systems Manager in Hybrid- und Multicloud-Umgebungen erforderliche IAM-Servicerolle

Aufgabe 4: Hinzufügen von Inline-Richtlinien zu einer übernommenen Rolle, um andere aufzurufen AWS-Services

Wenn eine Änderungsanforderung andere AWS-Services mithilfe der Rolle „Change ManagerÜbernehmen“ aufruft, muss die Rolle „Übernehmen“ so konfiguriert werden, dass sie berechtigt ist, diese Dienste aufzurufen. Diese Anforderung gilt für alle AWS Automations-Runbooks (AWS-*-Runbooks), die möglicherweise in einer Änderungsanforderung verwendet werden, wie z. B. die Runbooks`AWS-ConfigureS3BucketLogging`, und`AWS-CreateDynamoDBBackup`. `AWS-RestartEC2Instance` Diese Anforderung gilt auch für alle von Ihnen erstellten benutzerdefinierten Runbooks, die andere mithilfe AWS-Services von Aktionen aufrufen, die andere Dienste aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:CreateStack`- oder `aws:copyImage`-Aktionen verwenden, dann müssen Sie die Servicerolle mit der Berechtigung zum Aufrufen solcher Services konfigurieren. Sie können Berechtigungen für andere AWS-Services aktivieren, indem Sie der IAM-Rolle eine eingebundene Richtlinie hinzufügen.

So fügen Sie einer angenommenen Rolle eine eingebunden Richtlinie hinzu, um andere AWS-Services (IAM-Konsole) aufzurufen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie in der Liste den Namen der Übernahmerolle aus, die Sie aktualisieren möchten, z. B `MyChangeManagerAssumeRole`.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie Add permissions, Create inline policy (Berechtigungen hinzufügen, eingebundene Richtlinie erstellen).

6. Wählen Sie den Tab JSON.
7. Geben Sie ein JSON-Richtliniendokument für das Dokument ein AWS-Services , das Sie aufrufen möchten. Nachfolgend sind zwei Beispiele für JSON-Richtliniendokumente aufgeführt.

   Amazon-S3-**`PutObject`** und **`GetObject`**-Beispiel

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

   Amazon EC2-**`CreateSnapshot`** und **`DescribeSnapShots`**-Beispiel

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:DescribeSnapshots",
         "Resource":"*"
      }
   ]
}
```

   Details zur IAM-Richtliniensprache und finden Sie in der IAM JSON Policy Reference im IAM-Benutzerhandbuch.

8. Wählen Sie, wenn Sie fertig sind, Review policy (Richtlinie überprüfen) aus. Die Richtlinienvalidierung meldet mögliche Syntaxfehler.
9. Für Name geben Sie einen Namen zur Identifizierung der Richtlinie ein, die Sie erstellen. Überprüfen Sie unter Summary die Richtlinienzusammenfassung, um die Berechtigungen einzusehen, die von Ihrer Richtlinie gewährt werden. Wählen Sie dann Create policy aus, um Ihre Eingaben zu speichern.
10. Nachdem Sie eine Inline-Richtlinie erstellt haben, wird sie automatisch in Ihre Rolle eingebettet.

Aufgabe 5: Konfigurieren des Benutzerzugriffs auf Change Manager

Wenn Ihrem Benutzer, Ihrer Gruppe oder Rolle Administratorrechte zugewiesen sind, haben Sie Zugriff auf Change Manager. Wenn Sie nicht über Administratorberechtigungen verfügen, muss ein Administrator die von `AmazonSSMFullAccess` verwaltete Richtlinie oder eine Richtlinie, die vergleichbare Berechtigungen bereitstellt, Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zuweisen.

Konfigurieren Sie mit den folgenden Schritten einen Benutzer zur Verwendung von Change Manager. Der ausgewählte Benutzer verfügt über die Berechtigung zum Konfigurieren und Ausführen von Change Manager.

Abhängig von der Identitätsanwendung, die Sie in Ihrer Organisation verwenden, können Sie eine der drei verfügbaren Optionen zum Konfigurieren des Benutzerzugriffs auswählen. Weisen Sie beim Konfigurieren des Benutzerzugriffs Folgendes zu oder fügen Sie Folgendes hinzu:

1. Weisen Sie die `AmazonSSMFullAccess`-Richtlinie oder eine vergleichbare Richtlinie zu, die Zugriff auf Systems Manager gewährt.
2. Weisen Sie die `iam:PassRole`-Richtlinie zu.
3. Fügen Sie den ARN für die von Change Manager übernommene Rolle hinzu, die Sie am Ende von Aufgabe 2: Erstellen einer Übernahmerolle für Change Manager kopiert haben.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter Erstellen einer Rolle für einen externen Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch.

- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter Erstellen einer Rolle für einen IAM-Benutzer im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Sie haben die Konfiguration der erforderlichen Rollen für Change Manager abgeschlossen. Sie können jetzt die Change Manager-Übernahmerolle-ARN in Ihren Change Manager-Vorgängen verwenden.

Steuern des Zugriffs auf Runbook-Workflows für automatische Genehmigung

In jeder Änderungsvorlage, die für Ihre Organisation oder Ihr Konto erstellt wurde, können Sie angeben, ob Änderungsanforderungen, die mit dieser Vorlage erstellt wurden, als automatisch genehmigte Änderungsanforderungen ausgeführt werden können. Dies bedeutet, dass sie automatisch ohne Überprüfungsschritt ausgeführt werden (mit Ausnahme von Change–Freeze-Ereignissen).

Möglicherweise möchten Sie jedoch verhindern, dass bestimmte Benutzer, Gruppen oder AWS Identity and Access Management-(IAM)-Rollen automatisch genehmigte Änderungsanforderungen ausführen, selbst wenn eine Änderungsvorlage dies zulässt. Sie können dies durch die Verwendung des `ssm:AutoApprove`-Bedingungsschlüssel für den `StartChangeRequestExecution`-Vorgang in einer IAM-Richtlinie tun, die der Benutzer-, Gruppen- oder IAM-Rolle zugewiesen ist.

Sie können die folgende Richtlinie als Inline-Richtlinie hinzufügen, wobei die Bedingung als `false` angegeben wird, um zu verhindern, dass Benutzer automatisch genehmigungsfähige Änderungsanforderungen ausführen.

```
{
    "Version": "2012-10-17",
    "Statement": [
            {
            "Effect": "Allow",
            "Action": "ssm:StartChangeRequestExecution",
```

```
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "ssm:AutoApprove": "false"
                }
            }
        }
    ]
}
```

Informationen zum Festlegen von Inline-Richtlinien finden Sie unter Inline-Richtlinien und Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch.

Weitere Informationen über Bedingungsschlüssel für Systems Manager finden Sie unter Condition keys for Systems Manager (Bedingungsschlüssel für Systems Manager).

Arbeiten mit Change Manager

Mit Change Manager, eine Funktion von AWS Systems Manager, können Benutzer in Ihrer gesamten Organisation oder in einem einzigen AWS-Konto änderungsbezogene Aufgaben ausführen, für die ihnen die erforderlichen Berechtigungen erteilt wurden. Change Manager-Aufgaben umfassen u. a. folgende:

- Änderungsvorlagen erstellen, prüfen und genehmigen oder ablehnen.

  Eine Änderungsvorlage ist eine Sammlung von Konfigurationseinstellungen in Change Manager, die beispielsweise erforderliche Genehmigungen, verfügbare Runbooks und Benachrichtigungsoptionen für Änderungsanforderungen definiert.
- Änderungsanforderungen erstellen, prüfen und genehmigen oder ablehnen.

  Ein Änderungsantrag ist eine Anforderung in Change Manager, um ein Automation-Runbook auszuführen, das eine oder mehrere Ressourcen in Ihren AWS- oder On-Premises-Umgebungen aktualisiert. Ein Änderungsantrag wird mit einer Änderungsvorlage erstellt.
- Geben Sie an, welche Benutzer in Ihrer Organisation oder Ihrem Konto zu Prüfern für Änderungsvorlagen und Änderungsanforderungen gemacht werden können.
- Bearbeiten Sie Konfigurationseinstellungen, z. B. die Verwaltung von Benutzeridentitäten in Change Manager und welche der verfügbaren bewährten Methoden-Optionen in Ihrem Change Manager erzwungen werden. Weitere Informationen zum Konfigurieren dieser Einstellungen finden Sie unter Konfigurieren von Change Manager-Optionen und bewährten Methoden.

Themen

- Arbeiten mit Änderungsvorlagen
- Verwenden von Änderungsanforderungen
- Überprüfen von Details, Aufgaben und Zeitplänen für Änderungsanforderungen (Konsole)
- Aggregierte Anzahl von Änderungsaufträgen anzeigen (Befehlszeile)

Arbeiten mit Änderungsvorlagen

Eine Änderungsvorlage ist eine Sammlung von Konfigurationseinstellungen in Change Manager, die beispielsweise erforderliche Genehmigungen, verfügbare Runbooks und Benachrichtigungsoptionen für Änderungsanforderungen definiert.

> **Note**
>
> AWS bietet eine Beispiel-Änderungsvorlage namens Hello World, die Sie zum Ausprobieren von Change Manager verwenden können, eine Funktion von AWS Systems Manager. Sie erstellen jedoch Ihre eigenen Änderungsvorlagen, um die Änderungen zu definieren, die Sie an den Ressourcen in Ihrer Organisation oder Ihrem Konto zulassen möchten.

Die Änderungen, die bei der Ausführung eines Runbook-Workflows vorgenommen werden, basieren auf dem Inhalt eines Automation-Runbooks. In jede von Ihnen erstellte Änderungsvorlage können Sie ein oder mehrere Automation-Runbooks aufnehmen, aus denen der Benutzer, der eine Änderungsanforderung stellt, auswählen kann, um sie während der Aktualisierung auszuführen. Sie können auch Änderungsvorlagen erstellen, mit denen Anforderer ein beliebiges Automation-Runbook für den Änderungsantrag auswählen können.

Um eine Änderungsvorlage zu erstellen, können Sie die Builder-Option in der Konsolenseite Vorlage erstellen verwenden, um eine Änderungsvorlage zu erstellen. Alternativ können Sie mit der Editor-Option JSON- oder YAML-Inhalte mit der gewünschten Konfiguration für Ihren Runbook-Workflow manuell erstellen. Sie können auch ein Befehlszeilentool verwenden, um eine Änderungsvorlage zu erstellen, wobei JSON-Inhalt für die Änderungsvorlage in einer externen Datei gespeichert ist.

Themen

- Testen Sie die Vorlage für AWS verwaltete Hello World Änderungen
- Erstellen von Änderungsvorlagen

- Überprüfen und Genehmigen oder Ablehnen von Änderungsvorlagen
- Löschen von Änderungsvorlagen

Testen Sie die Vorlage für AWS verwaltete **`Hello World`** Änderungen

Sie können die Beispielvorlage für Änderungen verwenden`AWS-HelloWorldChangeTemplate`, in der das Automation-Runbook zum Beispiel verwendet wird`AWS-HelloWorld`, um den Überprüfungs- und Genehmigungsprozess zu testen, nachdem Sie die Einrichtung Change Manager abgeschlossen haben. Dabei handelt es sich um eine Funktion von AWS Systems Manager. Diese Vorlage dient zum Testen oder Überprüfen der konfigurierten Berechtigungen, Genehmigungszuweisungen und des Genehmigungsprozesses. Die Genehmigung zur Verwendung dieser Änderungsvorlage in Ihrer Organisation oder Ihrem Konto wurde bereits von AWS bereitgestellt. Jeder Änderungsantrag, der auf dieser Änderungsvorlage basiert, muss jedoch weiterhin von Prüfern in Ihrer Organisation oder Ihrem Konto genehmigt werden.

Anstatt Änderungen an einer Ressource vorzunehmen, besteht das Ergebnis des mit dieser Vorlage verknüpften Runbook-Workflows darin, eine Meldung in der Ausgabe eines Automatisierungsschritts zu drucken.

Bevor Sie beginnen

Überprüfen Sie zu Beginn, ob Sie die folgenden Aufgaben ausgeführt haben:

- Wenn Sie Änderungen in einer Organisation verwalten AWS Organizations möchten, führen Sie die unter beschriebenen Aufgaben zur Einrichtung der Organisation durch. Einrichten von Change Manager für eine Organisation (Management-Konto)
- Konfigurieren Sie Change Manager für Ihr delegiertes Administratorkonto oder ein einzelnes Konto, wie unter Konfigurieren von Change Manager-Optionen und bewährten Methoden beschrieben.

> **Note**
>
> Wenn Sie in Ihren Change Manager-Einstellungen die Bewährte–Methoden-Option Überwachen für alle Vorlagen erforderlich aktiviert haben, schalten Sie sie vorübergehend aus, während Sie die Änderungsvorlage Hello World testen.

Um die AWS verwaltete Hello World-Änderungsvorlage auszuprobieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie Create request (Erstellen einer Anfrage).
4. Wählen Sie die Änderungsvorlage mit dem Namen `AWS-HelloWorldChangeTemplate` und wählen Sie danach Weiter.
5. Geben Sie für Name einen Namen für die Änderungsanforderung ein, mit der ihre Funktion leicht zu erkennen ist, z. B. **`MyChangeRequestTest`**.
6. Weitere Informationen zu den weiteren Schritten zum Erstellen der Änderungsanforderung finden Sie unter Erstellen von Änderungsanforderungen.

Nächste Schritte

Weitere Informationen zum Genehmigen von Änderungsanforderungen finden Sie unter Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen.

Um den Status und die Ergebnisse Ihres Änderungsantrags anzuzeigen, wählen Sie den Namen Ihres Änderungsantrags auf der Registerkarte Anforderungen in Change Manager.

Erstellen von Änderungsvorlagen

Eine Änderungsvorlage ist eine Sammlung von Konfigurationseinstellungen in Change Manager, die beispielsweise erforderliche Genehmigungen, verfügbare Runbooks und Benachrichtigungsoptionen für Änderungsanforderungen definiert.

Sie können Änderungsvorlagen für Ihre Vorgänge in Change Manager, einer Funktion von AWS Systems Manager, mithilfe der Konsole, die Builder- und Editor-Optionen enthält, oder mit Befehlszeilentools erstellen.

Themen

- Über Genehmigungen in Ihren Änderungsvorlagen
- Erstellen von Änderungsvorlagen mit Builder
- Erstellen von Änderungsvorlagen mit dem Editor
- Erstellen von Änderungsvorlagen mit Befehlszeilenwerkzeugen

Über Genehmigungen in Ihren Änderungsvorlagen

Für jede von Ihnen erstellte Änderungsvorlage können Sie bis zu fünf Genehmigungsebenen für daraus erstellte Änderungsanfragen angeben. Für jede dieser Ebenen können Sie bis zu fünf potenzielle Genehmiger benennen. Ein Genehmiger ist nicht auf einen einzelnen Benutzer beschränkt. Sie können auch eine IAM-Gruppe oder IAM-Rolle als einzelne Genehmiger angeben. Für IAM-Gruppen und IAM-Rollen können ein oder mehrere Benutzer, die zu der Gruppe oder Rolle gehören, Genehmigungen für den Erhalt der Gesamtzahl der Genehmigungen erteilen, die für eine Änderungsanforderung erforderlich sind. Sie können auch mehr Genehmiger angeben, als Ihre Änderungsvorlage erfordert.

Change Manager unterstützt zwei Hauptansätze für Genehmigungen: Genehmigungen pro Ebene und Genehmigungen pro Zeile. In manchen Situationen ist auch eine Kombination der beiden Typen möglich. Wir empfehlen, in Ihren Change Manager-Operationen nur Genehmigungen pro Ebene zu verwenden.

Per-level approvals

Empfohlen. Ab dem 23. Januar 2023 unterstützt Change Manager Genehmigungen pro Ebene. In diesem Modell geben Sie zunächst für jede Genehmigungsebene in Ihrer Änderungsvorlage an, wie viele Genehmigungen für diese Ebene erforderlich sind. Anschließend legen Sie mindestens so viele Genehmiger für die Ebene fest und können weitere Genehmiger angeben. Allerdings muss nur die von Ihnen festgelegte Anzahl von Genehmigern pro Ebene die Änderungsanfrage genehmigen. Sie können zum Beispiel fünf Genehmiger angeben, aber nur drei Genehmigungen verlangen.

Beispiele für diesen Genehmigungstyp in Konsolenansicht und JSON finden Sie unter the section called “Beispiel für eine Genehmigungskonfiguration pro Ebene”.

Per-line approvals

Unterstützt aus Gründen der Abwärtskompatibilität. Die ursprüngliche Version von Change Manager hat nur Genehmigungen pro Zeile unterstützt. In diesem Modell wird jeder für eine Genehmigungsebene angegebene Genehmiger als Genehmigungszeile dargestellt. Jeder Genehmiger musste eine Änderungsanfrage genehmigen, damit es auf dieser Ebene genehmigt werden konnte. Vor dem 23. Januar 2023 war dies das einzige unterstützte Modell für Genehmigungen. Änderungsvorlagen, die vor diesem Datum erstellt wurden, unterstützen weiterhin Genehmigungen pro Zeile, aber wir empfehlen, stattdessen Genehmigungen pro Ebene zu verwenden.

Beispiele für diesen Genehmigungstyp in Konsolenansicht und JSON finden Sie unter the section called “Beispiel für eine Genehmigungskonfiguration pro Zeile”.

Combined per-line and per-level approvals

Nicht empfohlen. In der Konsole unterstützt die Registerkarte Builder nicht mehr das Hinzufügen von Genehmigungen pro Zeile. In einigen Fällen kann es jedoch vorkommen, dass Sie in einer Änderungsvorlage sowohl Genehmigungen pro Zeile als auch pro Ebene erhalten. Dies kann vorkommen, wenn Sie eine Änderungsvorlage aktualisieren, die vor dem 23. Januar 2023 erstellt wurde, oder wenn Sie eine Änderungsvorlage erstellen oder aktualisieren, indem Sie ihren YAML-Inhalt manuell bearbeiten,

Beispiele für diesen Genehmigungstyp in Konsolenansicht und JSON finden Sie unter the section called “Beispiel für eine kombinierte Genehmigungskonfiguration pro Ebene und pro Zeile”.

Important

Es ist zwar möglich, eine Änderungsvorlage zu erstellen, die Genehmigungen pro Zeile und pro Ebene kombiniert, diese Konfiguration ist jedoch nicht empfohlen oder erforderlich. Die Genehmigungsart, die mehr Genehmigungen erfordert (Genehmigungen pro Zeile oder pro Ebene), hat Vorrang. Beispiele:

- Wenn eine Änderungsvorlage drei Genehmigungen pro Ebene, aber fünf Genehmigungen pro Zeile angibt, sind fünf Genehmigungen erforderlich.
- Wenn eine Änderungsvorlage vier Genehmigungen pro Ebene, aber zwei Genehmigungen pro Zeile vorsieht, sind vier Genehmigungen erforderlich.

Sie können eine Ebene erstellen, die sowohl Genehmigungen pro Zeile als auch pro Ebene enthält, indem Sie den YAML- oder JSON-Inhalt manuell bearbeiten. Anschließend werden auf der Registerkarte Builder Steuerelemente zum Festlegen der erforderlichen Anzahl von Genehmigungen sowohl für die Ebene als auch für einzelne Zeilen angezeigt. Neue Ebenen, die Sie mithilfe der Konsole hinzufügen, unterstützen jedoch weiterhin nur Genehmigungskonfigurationen pro Ebene.

Benachrichtigungen und Ablehnungen von Änderungsanfragen

Amazon-SNS-Benachrichtigungen

Wenn eine Änderungsanfrage mit Ihrer Änderungsvorlage erstellt wird, werden Benachrichtigungen an Abonnenten des Amazon Simple Notification Service (Amazon SNS)-Themas gesendet, das für Genehmigungsbenachrichtigungen auf dieser Ebene vorgesehen ist. Sie können das Benachrichtigungsthema in der Änderungsvorlage angeben oder dem Benutzer, der die Änderungsanfrage erstellt, erlauben, eines anzugeben.

Nachdem die Mindestanzahl erforderlicher Genehmigungen auf einer Ebene empfangen wurde, werden Benachrichtigungen an Genehmiger gesendet, die das Amazon-SNS-Thema für die nächste Ebene abonniert haben, und so weiter.

> **Important**
>
> Stellen Sie sicher, dass die von Ihnen gemeinsam benannten IAM-Rollen, -Gruppen und -Benutzer über ausreichend Genehmigungen verfügen, um die von Ihnen angegebene Anzahl von Genehmigungen zu erfüllen. Wenn Sie beispielsweise nur eine einzelne IAM-Gruppe mit drei Benutzern als Genehmiger festlegen, können Sie nicht festlegen, dass auf dieser Ebene fünf Genehmigungen obligatorisch sind, sondern nur drei oder weniger.

Ablehnungen von Änderungsanfragen

Unabhängig davon, wie viele Genehmigungsebenen und Genehmiger Sie angeben, ist nur eine Ablehnung einer Änderungsanfrage erforderlich, um zu verhindern, dass der Runbook-Workflow für diese Anfrage ausgeführt wird.

Beispiele für Change Manager-Genehmigungsarten

Die folgenden Beispiele veranschaulichen die Konsolenansicht und den JSON-Inhalt für die drei Arten von Genehmigungstypen in Change Manager.

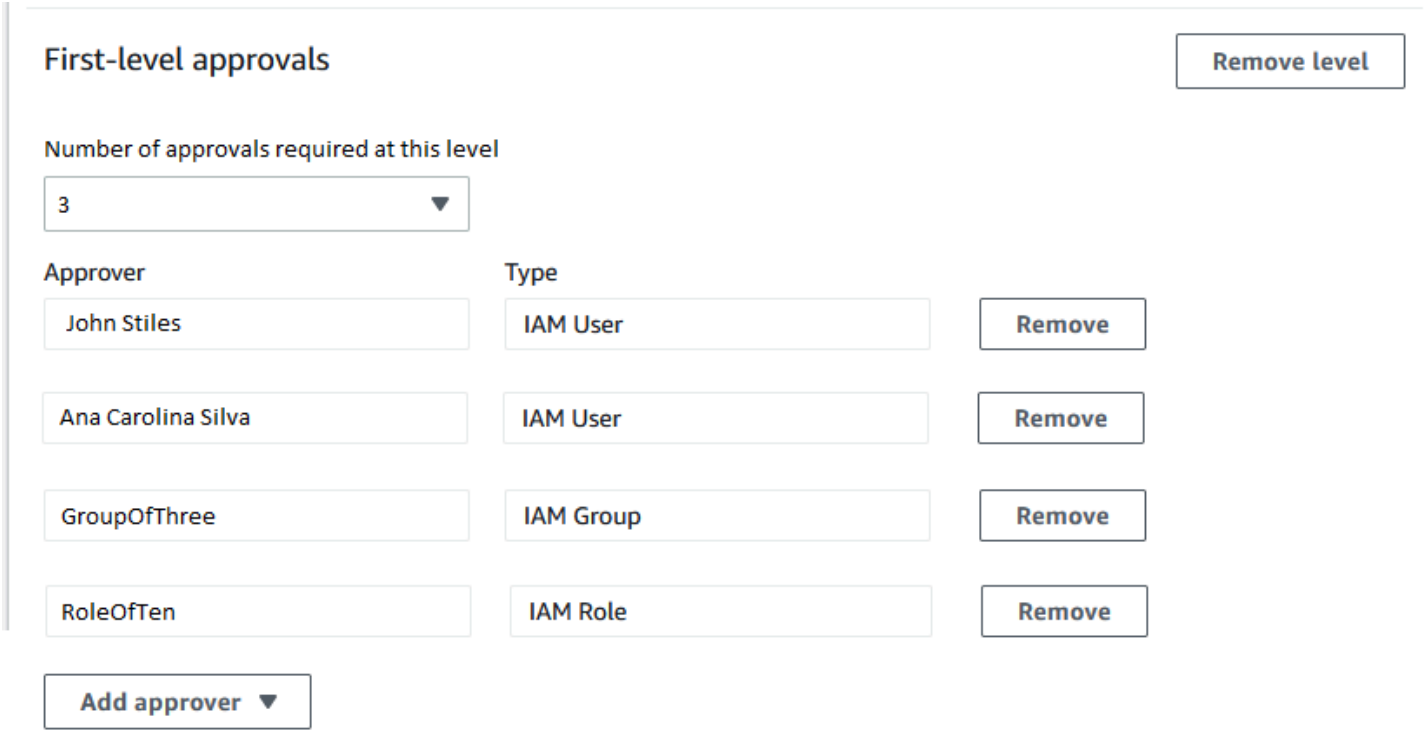
Themen

- Beispiel für eine Genehmigungskonfiguration pro Ebene
- Beispiel für eine Genehmigungskonfiguration pro Zeile
- Beispiel für eine kombinierte Genehmigungskonfiguration pro Ebene und pro Zeile

Beispiel für eine Genehmigungskonfiguration pro Ebene

Bei der im folgenden Image gezeigten Einrichtung der Genehmigungsebene pro Ebene sind drei Genehmigungen erforderlich. Diese Genehmigungen können aus einer beliebigen Kombination von IAM-Benutzern, Gruppen und Rollen stammen, die als Genehmiger angegeben sind. Zu den angegebenen Genehmigern gehören zwei IAM-Benutzer (John Stiles und Ana Carolina Silva), eine Benutzergruppe mit drei Mitgliedern (`GroupOfThree`) und eine Benutzerrolle, die zehn Benutzer repräsentiert (`RoleOfTen`).

Wenn alle drei Benutzer in der `GroupOfThree`-Gruppe die Änderungsanfrage genehmigen, wird sie für diese Ebene genehmigt. Es ist nicht erforderlich, eine Genehmigung von jedem Benutzer, Gruppe oder Rolle zu erhalten. Die Mindestanzahl an Genehmigungen kann von einer beliebigen Kombination festgelegter Genehmiger stammen. Wir empfehlen, für Ihre Change Manager-Operationen nur Genehmigungen pro Ebene zu verwenden.



Das folgende Beispiel veranschaulicht einen Teil des YAML-Codes für diese Konfiguration.

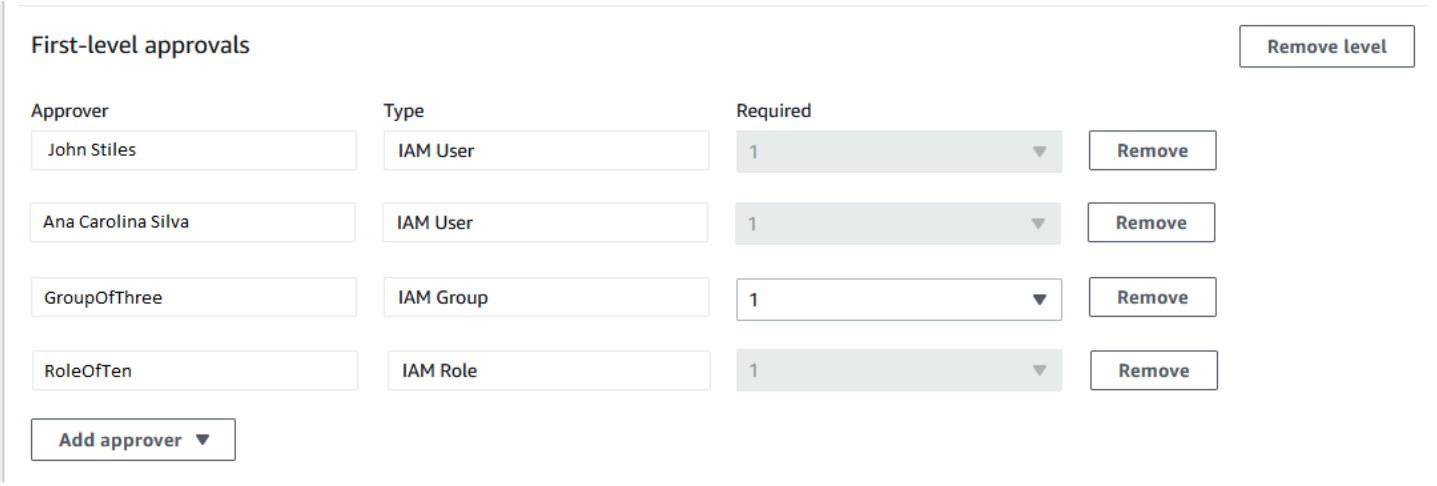
> **Note**
>
> Diese Version des YAML-Codes enthält eine zusätzliche Eingabe, `MinRequiredApprovals` (mit einem großen Anfangsbuchstaben `M`). Der Wert für diese Eingabe gibt an, wie viele Genehmigungen von allen verfügbaren Prüfern erforderlich sind. Beachten Sie

auch, dass der Wert `minRequiredApprovals` (in Kleinbuchstaben `m`) für jeden Genehmiger in der `Approvers`-Liste `0` (Null) ist. Dies zeigt an, dass der Genehmiger zu den Gesamtgenehmigungen beitragen kann, aber nicht dazu verpflichtet ist.

```
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
```

Beispiel für eine Genehmigungskonfiguration pro Zeile

In der Konfiguration der Genehmigungsebene, die im folgenden Image dargestellt ist, werden vier Genehmiger angegeben. Dazu gehören zwei IAM-Benutzer (John Stiles und Ana Carolina Silva), eine Benutzergruppe mit drei Mitgliedern (`GroupOfThree`) und eine Benutzerrolle, die zehn Benutzer repräsentiert (`RoleOfTen`). Aus Gründen der Abwärtskompatibilität werden Genehmigungen pro Zeile unterstützt, jedoch nicht empfohlen.



Damit die Änderungsanfrage in dieser Genehmigungskonfiguration pro Zeile genehmigt werden kann, muss sie von allen genehmigenden Zeilen genehmigt werden:: John Stiles, Ana Carolina Silva, einem Mitglied der `GroupOfThree`-Gruppe und einem Mitglied der `RoleOfTen`-Rolle.

Das folgende Beispiel veranschaulicht einen Teil des YAML-Codes für diese Konfiguration.

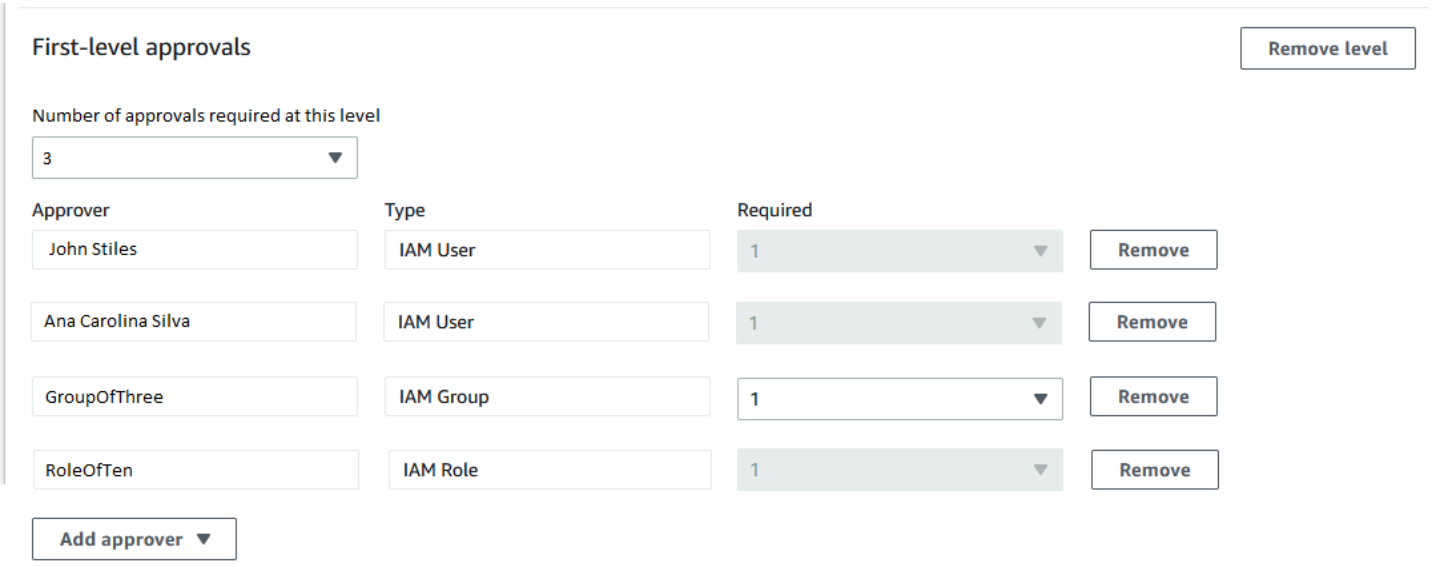
> **Note**
>
> Beachten Sie, dass der Wert für jeden `minRequiredApprovals`-Genehmiger 1 beträgt. Dies bedeutet, dass von jedem Genehmiger eine Genehmigung erforderlich ist.

```
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
```

Beispiel für eine kombinierte Genehmigungskonfiguration pro Ebene und pro Zeile

Im folgenden Image werden bei der kombinierten Genehmigungskonfiguration pro Ebene und pro Zeile drei Genehmigungen für die Ebene angegeben, aber vier Genehmigungen für die Genehmigungen der einzelnen Positionen. Welcher Genehmigungstyp mehr Genehmigungen erfordert, hat Vorrang vor dem anderen, sodass für diese Konfiguration vier Genehmigungen erforderlich sind. Eine kombinierte Genehmigung pro Ebene und pro Linie wird nicht empfohlen.




```
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
```

Themen

- Erstellen von Änderungsvorlagen mit Builder
- Erstellen von Änderungsvorlagen mit dem Editor
- Erstellen von Änderungsvorlagen mit Befehlszeilenwerkzeugen

Erstellen von Änderungsvorlagen mit Builder

Mit dem Builder für Änderungsvorlagen in Change Manager, eine Funktion von AWS Systems Manager, können Sie den in Ihrer Änderungsvorlage definierten Runbook-Workflow konfigurieren, ohne JSON- oder YAML-Syntax verwenden zu müssen. Nachdem Sie Ihre Optionen festgelegt haben, konvertiert das System Ihre Eingabe in das YAML-Format, das Systems Manager zum Ausführen von Runbook-Workflows verwenden kann.

So erstellen Sie eine Änderungsvorlage mit Builder

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.

3. Wählen Sie Create template (Vorlage erstellen) aus.
4. Geben Sie für Name einen Namen für die Vorlage ein, mit der ihre Funktion leicht zu erkennen ist, z. B. **`UpdateEC2LinuxAMI`**.
5. Gehen Sie im Abschnitt Details zur Änderungsvorlage wie folgt vor:

   - Geben Sie für Beschreibung eine kurze Erklärung ein, wie und wann die von Ihnen erstellte Änderungsvorlage verwendet werden soll.

     Mit dieser Beschreibung können Benutzer, die Änderungsanforderungen erstellen, feststellen, ob sie die richtige Änderungsvorlage verwenden. Es hilft denjenigen, die Änderungsanforderungen überprüfen, zu verstehen, ob die Anforderung genehmigt werden soll.
   - Geben Sie für Änderungsvorlagentyp an, ob Sie eine Standard- oder eine Notfalländerungsvorlage erstellen.

     Eine Vorlage für Notfalländerungen wird für Situationen verwendet, in denen eine Änderung auch dann vorgenommen werden muss, wenn die Änderungen ansonsten durch ein Ereignis im verwendeten Kalender blockiert werden AWS Systems Manager Change Calendar. Änderungsanforderungen, die aus einer Notfalländerungsvorlage erstellt wurden, müssen immer noch von den dafür vorgesehenen Genehmigern genehmigt werden, aber die angeforderten Änderungen können auch dann ausgeführt werden, wenn der Kalender gesperrt ist.
   - Geben Sie für Runbook-Optionen die Runbooks an, aus denen Benutzer beim Erstellen einer Änderungsanforderung auswählen können. Sie können ein einzelnes Runbook oder mehrere Runbooks hinzufügen. Alternativ können Sie Anforderern erlauben, anzugeben, welches Runbook verwendet werden soll. In jedem dieser Fälle kann nur ein Runbook in der Änderungsanforderung aufgenommen werden.
   - Wählen Sie für Runbook die Namen der Runbooks und die Versionen dieser Runbooks aus, aus denen Benutzer für ihre Änderungsanforderungen auswählen können. Unabhängig davon, wie viele Runbooks Sie der Änderungsvorlage hinzufügen, kann pro Änderungsanforderung nur eines ausgewählt werden.

     Sie geben kein Runbook an, wenn Sie Jedes Runbook kann verwendet werden vorher bereits gewählt haben.

> **Tip**
>
> Wählen Sie ein Runbook und eine Runbook-Version aus und wählen Sie dann View (Anzeigen), um den Inhalt des Runbooks in der Oberfläche von Systems Manager Documents zu prüfen.

6. Geben Sie im Abschnitt Vorlageninformationen mit Markdown Informationen für Benutzer ein, die Änderungsanforderungen von dieser Änderungsvorlage erstellen. Wir haben eine Reihe von Fragen bereitgestellt, die Sie für Benutzer, die Änderungsanforderungen erstellen, einfügen können, oder Sie können stattdessen andere Informationen und Fragen hinzufügen.

   > **Note**
   >
   > Markdown ist eine Markup-Sprache, die es Ihnen ermöglicht, Dokumente und einzelne Schritte innerhalb des Dokuments mit Beschreibungen im Wiki-Stil zu versehen. Weitere Informationen zur Verwendung von Markdown finden Sie unter Verwenden von Markdown in AWS.

   Wir empfehlen, Benutzern Fragen zur Beantwortung ihrer Änderungsanforderungen zur Verfügung zu stellen, damit Genehmiger entscheiden können, ob sie jede Änderungsanforderung erteilen möchten oder nicht, z. B. das Auflisten aller manuellen Schritte, die für die Ausführung als Teil der Änderung erforderlich sind, und ein Rollback-Plan.

   > **Tip**
   >
   > Wechseln Sie zwischen Vorschau ausblenden und Vorschau anzeigen, um zu sehen, wie der Inhalt während der Erstellung aussieht.

7. Im Abschnitt Change request approvals (Genehmigungen für Änderungsanträge) gehen Sie wie folgt vor:

   - (Optional) Wenn Sie zulassen möchten, dass Änderungsanforderungen, die aus dieser Änderungsvorlage erstellt wurden, automatisch ausgeführt werden, ohne von Genehmigern geprüft zu werden (mit Ausnahme von Change–Freeze-Ereignissen), wählen Sie Aktivieren der automatischen Genehmigung (Enable auto-approval).

> **Note**
>
> Durch Aktivieren von automatischen Genehmigungen in einer Änderungsvorlage erhalten Benutzer die Option zur Umgehung von Überprüfern. Sie können weiterhin auswählen, ob Prüfer beim Erstellen einer Änderungsanforderung angegeben werden sollen. Daher müssen Sie in der Änderungsvorlage weiterhin Prüferoptionen angeben.

> **Important**
>
> Wenn Sie die automatische Genehmigung für eine Änderungsvorlage aktivieren, können Benutzer Änderungsanforderungen mithilfe dieser Vorlage übermitteln, die vor der Ausführung nicht von Prüfern überprüft werden müssen (mit Ausnahme von Change-Freeze-Genehmigern). Wenn Sie einen bestimmten Benutzer, eine Gruppe oder IAM-Rolle daran hindern möchten, automatische Genehmigungsanforderungen zu senden, können Sie eine Bedingung in einer IAM-Richtlinie zu diesem Zweck verwenden. Weitere Informationen finden Sie unter Steuern des Zugriffs auf Runbook-Workflows für automatische Genehmigung.

- Wählen Sie für Anzahl der auf dieser Ebene erforderlichen Genehmigungen die Anzahl der Genehmigungen aus, die aus dieser Änderungsvorlage erstellte Änderungsanfragen für diese Ebene erhalten müssen.
- Um obligatorische Genehmiger der ersten Ebene hinzuzufügen, wählen Sie Genehmiger hinzufügen und wählen Sie eine der folgenden Optionen:
  - In der Vorlage angegebene Genehmiger - Wählen Sie einen oder mehrere Benutzer, Gruppen oder AWS Identity and Access Management -(IAM)-Rollen Ihres Kontos aus, um Änderungsanforderungen zu genehmigen, die mit dieser Änderungsvorlage erstellt wurden. Alle Änderungsanforderungen, die mit dieser Vorlage erstellt werden, müssen von jedem von Ihnen angegebenen Genehmiger geprüft und genehmigt werden.
  - Request specified approvers (Angegebene Genehmiger anfordern) – Der Benutzer, der die Änderungsanforderung stellt, gibt Prüfer zum Zeitpunkt der Anforderung an und kann aus einer Liste von Benutzern in Ihrem Konto wählen.

    Die Nummer, die Sie im Feld Erforderlich eingeben, legt fest, wie viele Prüfer von einer Änderungsanforderung angegeben werden müssen, die diese Änderungsvorlage verwendet.

> **Important**
>
> Vor dem 23. Januar 2023 konnten auf der Registerkarte Builder nur Genehmigungen pro Zeile angegeben werden. Neue Änderungsvorlagen und neue Ebenen, die Sie mithilfe der Registerkarte Builder zu vorhandenen Änderungsvorlagen hinzufügen, unterstützen nur Genehmigungen pro Ebene. Wir empfehlen, in Ihren Change Manager-Operationen nur Genehmigungen pro Ebene zu verwenden.
> Weitere Informationen finden Sie unter Über Genehmigungen in Ihren Änderungsvorlagen.

- Gehen Sie für SNS-Thema zur Benachrichtigung von Genehmiger wie folgt vor:

  1. Wählen Sie eine der folgenden Optionen, um das Amazon Simple Notification Service (Amazon SNS)-Thema in Ihrem Konto anzugeben, das für das Senden von Benachrichtigungen an die Genehmiger verwendet werden soll, wenn eine Änderungsanforderung zur Überprüfung bereit ist:

     - Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.
     - Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen AWS-Konto. (Diese Option ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)
     - SNS-Thema angeben, wenn die Änderungsanforderung erstellt wird - Der Benutzer, der eine Änderungsanforderung erstellt, kann das Amazon SNS-Thema angeben, das für Benachrichtigungen verwendet werden soll.

     > **Note**
     >
     > Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

  2. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

8. (Optional) Um eine zusätzliche Ebene von Genehmigern hinzuzufügen, wählen Sie Add approval level (Genehmigungsebene hinzufügen) und wählen Sie zwischen vorlagenspezifischen Genehmigern und angeforderten Genehmigern für diese Ebene. Wählen Sie dann ein SNS-Thema aus, um diese Genehmiger zu benachrichtigen.

   Nachdem alle Genehmigungen von Genehmiger der ersten Ebene eingegangen sind, werden Genehmiger der zweiten Ebene benachrichtigt usw.

   Sie können maximal fünf Genehmigungsebenen in jeder Vorlage hinzufügen. So könnten Sie beispielsweise für die erste Stufe die Genehmigung von Benutzern in technischen Rollen und für die zweite Stufe die Genehmigung des Managers verlangen.

9. Geben Sie im Abschnitt Überwachung für den zu überwachenden CloudWatch Alarm den Namen eines CloudWatch Amazon-Alarms im aktuellen Konto ein, um den Fortschritt der Runbook-Workflows zu überwachen, die auf dieser Vorlage basieren.

   > **Tip**
   >
   > Um einen neuen Alarm zu erstellen oder die Einstellungen eines Alarms, den Sie angeben möchten, zu überprüfen, wählen Sie Die CloudWatch Amazon-Konsole öffnen. Informationen zum Arbeiten mit CloudWatch Alarmen finden Sie unter Verwenden von CloudWatch Alarmen im CloudWatch Amazon-Benutzerhandbuch.

10. Führen Sie im Abschnitt Notifications (Benachrichtigungen) folgende Schritte aus:

    1. Wählen Sie eine der folgenden Optionen aus, um das Amazon SNS-Thema in Ihrem Konto anzugeben, das zum Senden von Benachrichtigungen über Änderungsanforderungen verwendet werden soll, die mit dieser Änderungsvorlage erstellt werden:

       - Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.
       - Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen AWS-Konto. (Diese Option ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)

> **Note**
>
> Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

2. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

11. (Optional) Wenden Sie im Abschnitt Tags ein oder mehrere Tag-Schlüssel-Name/Wert-Paare auf die Änderungsvorlage an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise eine Änderungsvorlage mit Tags versehen, um den Änderungstyp und die Umgebung, in der sie ausgeführt wird, zu identifizieren. In diesem Fall könnten Sie z.B. die folgenden Schlüsselname-Wert-Paare angeben:

    - `Key=TaskType,Value=InstanceRepair`
    - `Key=Environment,Value=Production`

    Weitere Informationen über das Taggen von System Manager-Ressourcen finden Sie unter Markieren von Systems Manager-Ressourcen.

12. Klicken Sie auf Save and preview (speichern und Vorschau ansehen).

13. Überprüfen Sie die Details der Änderungsvorlage, die Sie gerade erstellen.

    Wenn Sie die Änderungsvorlage ändern möchten, bevor Sie sie zur Überprüfung einreichen, wählen Sie Actions (Aktionen).

    Wenn Sie mit dem Inhalt der Änderungsvorlage zufrieden sind, klicken Sie auf Submit for review (Zur Überprüfung einreichen). Die Benutzer in Ihrer Organisation oder Ihrem Konto, die als Vorlagenprüfer in der Registerkarte Einstellungen in Change Manager angegeben sind, werden benachrichtigt, dass eine neue Änderungsvorlage zur Überprüfung aussteht.

Wenn ein Amazon SNS-Thema für Änderungsvorlagen angegeben wurde, werden Benachrichtigungen gesendet, wenn die Änderungsvorlage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen zu dieser Änderungsvorlage erhalten, können Sie zu Change Manager später zurückkehren, um ihren Status zu überprüfen.

Erstellen von Änderungsvorlagen mit dem Editor

Gehen Sie wie in diesem Thema beschrieben vor, um eine ÄnderungsvorlageChange Manager, eine Fähigkeit von, zu konfigurieren AWS Systems Manager, indem Sie JSON oder YAML eingeben, anstatt die Konsolensteuerelemente zu verwenden.

Erstellen einer Änderungsvorlage mit dem Editor

1. Wählen Sie im Navigationsbereich Change Manager aus.
2. Wählen Sie Create template (Vorlage erstellen) aus.
3. Geben Sie für Name einen Namen für die Vorlage ein, mit der ihre Funktion leicht zu erkennen ist, z. B. **`RestartEC2LinuxInstance`**.
4. Wählen Sie über Change template details (Vorlagendetails ändern) Editor.
5. Wählen Sie im Abschnitt Document Editor (Dokumenteneditor) die Option Edit (Bearbeiten) und geben Sie dann den JSON- oder YAML-Inhalt für Ihre Änderungsvorlage ein.

   Im Folgenden wird ein Beispiel gezeigt.

> **Note**
>
> Der Parameter `minRequiredApprovals` wird verwendet, um anzugeben, wie viele Prüfer auf einer bestimmten Ebene eine Änderungsanforderung genehmigen müssen, die mit dieser Vorlage erstellt wird.
> Dieses Beispiel zeigt zwei Genehmigungsebenen. Sie können bis zu fünf Genehmigungsebenen angeben, aber nur eine Ebene ist erforderlich.
> In der ersten Ebene muss der spezifische Benutzer „John-Doe“ jeden Änderungsantrag genehmigen. Danach müssen drei beliebige Mitglieder der IAM-Rolle `Admin` die Änderungsanforderung genehmigen.
> Weitere Informationen zum Genehmigen von Änderungsvorlagen finden Sie unter Über Genehmigungen in Ihren Änderungsvorlagen.

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. Klicken Sie auf Save and preview (speichern und Vorschau ansehen).
7. Überprüfen Sie die Details der Änderungsvorlage, die Sie gerade erstellen.

   Wenn Sie die Änderungsvorlage ändern möchten, bevor Sie sie zur Überprüfung einreichen, wählen Sie Actions (Aktionen).

   Wenn Sie mit dem Inhalt der Änderungsvorlage zufrieden sind, klicken Sie auf Submit for review (Zur Überprüfung einreichen). Die Benutzer in Ihrer Organisation oder Ihrem Konto, die als Vorlagenprüfer in der Registerkarte Einstellungen in Change Manager angegeben sind, werden benachrichtigt, dass eine neue Änderungsvorlage zur Überprüfung aussteht.

Wenn ein Amazon Simple Notification Service (Amazon SNS)-Thema für Änderungsvorlagen angegeben wurde, werden Benachrichtigungen gesendet, wenn die Änderungsvorlage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen zu dieser Änderungsvorlage erhalten, können Sie zu Change Manager später zurückkehren, um ihren Status zu überprüfen.

Erstellen von Änderungsvorlagen mit Befehlszeilenwerkzeugen

Die folgenden Verfahren beschreiben, wie Sie die AWS Command Line Interface (AWS CLI) (unter Linux oder Windows) verwenden oder AWS Tools for Windows PowerShell eine Änderungsanforderung inChange Manager, einer Fähigkeit von, erstellen AWS Systems Manager. macOS

Erstellen einer Änderungsvorlage:

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS Tools for PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

2. Erstellen Sie eine JSON-Datei auf Ihrem lokalen Computer und geben Sie Ihr beispielsweise einen Namen wie `MyChangeTemplate.json`. Kopieren Sie anschließend den folgenden Inhalt in die Datei:

   > **Note**
   >
   > Änderungsvorlagen verwenden eine Version von Schema 0.3, die nicht die gleiche Unterstützung wie für Automation-Runbooks enthält.

   Im Folgenden wird ein Beispiel gezeigt.

   > **Note**
   >
   > Der Parameter `minRequiredApprovals` wird verwendet, um anzugeben, wie viele Prüfer auf einer bestimmten Ebene eine Änderungsanforderung genehmigen müssen, die mit dieser Vorlage erstellt wird.

Dieses Beispiel zeigt zwei Genehmigungsebenen. Sie können bis zu fünf Genehmigungsebenen angeben, aber nur eine Ebene ist erforderlich.
In der ersten Ebene muss der spezifische Benutzer „John-Doe“ jeden Änderungsantrag genehmigen. Danach müssen drei beliebige Mitglieder der IAM-Rolle `Admin` die Änderungsanforderung genehmigen.
Weitere Informationen zum Genehmigen von Änderungsvorlagen finden Sie unter Über Genehmigungen in Ihren Änderungsvorlagen.

```
{
   "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. Führen Sie den folgenden Befehl aus, um die Änderungsvorlage zu erstellen.

   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"
   ```

   Weitere Informationen zu anderen Optionen, die Sie angeben können, finden Sie unter create-document.

   Das System gibt unter anderem folgende Informationen zurück

```
{
   "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":[

      ]
```

```
    }
}
```

Die Benutzer in Ihrer Organisation oder Ihrem Konto, die als Vorlagenprüfer in der Registerkarte Einstellungen in Change Manager angegeben sind, werden benachrichtigt, dass eine neue Änderungsvorlage zur Überprüfung aussteht.

Wenn ein Amazon Simple Notification Service (Amazon SNS)-Thema für Änderungsvorlagen angegeben wurde, werden Benachrichtigungen gesendet, wenn die Änderungsvorlage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen zu dieser Änderungsvorlage erhalten, können Sie zu Change Manager später zurückkehren, um ihren Status zu überprüfen.

Überprüfen und Genehmigen oder Ablehnen von Änderungsvorlagen

Wenn Sie inChange Manager, mit einer Funktion von, als Prüfer für Änderungsvorlagen angegeben sind AWS Systems Manager, werden Sie benachrichtigt, wenn eine neue Änderungsvorlage oder eine neue Version einer Änderungsvorlage auf Ihre Überprüfung wartet. Ein Amazon Simple Notification Service (Amazon SNS)-Thema sendet die Benachrichtigungen.

> **Note**
>
> Diese Funktionalität hängt davon ab, ob Ihr Konto so konfiguriert wurde, dass ein Amazon SNS-Thema verwendet wird, um Benachrichtigungen zur Überprüfung von Änderungsvorlagen zu senden. Informationen zum Festlegen eines Themas für die Benachrichtigung eines Vorlagenprüfers finden Sie unter Aufgabe 1: Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern.

Um die Änderungsvorlage zu überprüfen, folgen Sie dem Link in Ihrer Benachrichtigung, melden Sie sich bei der AWS Management Console an und folgen Sie den Schritten in diesem Verfahren.

Überprüfen und Genehmigen oder Ablehnen einer Änderungsvorlage

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie im Abschnitt Change templates (Änderungsvorlagen) unten auf der Registerkarte Overview (Übersicht) die Nummer in Pending review (ausstehende Prüfung).

4. Suchen Sie in der Liste Change templates (Änderungsvorlagen) den Namen der zu überprüfenden Änderungsvorlage und wählen Sie ihn aus.
5. Überprüfen Sie auf der Übersichtsseite den vorgeschlagenen Inhalt der Änderungsvorlage und führen Sie einen der folgenden Schritte aus:

   - Um die Änderungsvorlage zu genehmigen, wodurch sie in Änderungsanforderungen verwendet werden kann, wählen Sie Approve (Genehmigen).
   - Um die Änderungsvorlage abzulehnen, wodurch ihre Verwendung in Änderungsanforderungen verhindert wird, wählen Sie Reject (Ablehnen).

Löschen von Änderungsvorlagen

In diesem Thema wird beschrieben, wie Sie Vorlagen löschen, die Sie in Change Manager, eine Funktion von Systems Manager, erstellt haben. Wenn Sie Change Manager für eine Organisation verwenden, wird dieses Verfahren in Ihrem delegierten Administratorkonto durchgeführt.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie die Registerkarte Templates (Vorlagen) aus.
4. Wählen Sie den Namen der zu löschenden Vorlage aus.
5. Wählen Sie Actions, Delete template (Aktionen, Vorlage löschen).
6. Geben Sie in das Bestätigungsfeld das Wort **DELETE** ein und wählen Sie dann Delete (Löschen).

Verwenden von Änderungsanforderungen

Ein Änderungsantrag ist eine Anforderung in Change Manager, um ein Automation-Runbook auszuführen, das eine oder mehrere Ressourcen in Ihren AWS- oder On-Premises-Umgebungen aktualisiert. Ein Änderungsantrag wird mit einer Änderungsvorlage erstellt.

Wenn Sie eine Änderungsanforderung in Change Manager verwenden, eine Funktion von AWS Systems Manager, muss ein oder müssen mehrere Genehmiger in Ihrer Organisation oder Ihrem Konto die Anforderung überprüfen und genehmigen. Ohne die erforderlichen Genehmigungen kann der Runbook-Workflow, der die angeforderten Änderungen vornimmt, nicht ausgeführt werden.

Themen

- Erstellen von Änderungsanforderungen
- Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen

Erstellen von Änderungsanforderungen

Wenn Sie einen Änderungsantrag inChange Manager, mit einer Funktion von, erstellen AWS Systems Manager, führt die von Ihnen gewählte Änderungsvorlage in der Regel Folgendes aus:

- Bestimmt Genehmiger für die Änderungsanforderung oder gibt an, wie viele Genehmigungen erforderlich sind
- Gibt das Amazon Simple Notification Service (Amazon SNS)-Thema an, das verwendet werden soll, um Genehmiger über Ihre Änderungsanforderung zu benachrichtigen
- Spezifiziert einen CloudWatch Amazon-Alarm zur Überwachung des Runbook-Workflows für die Änderungsanforderung
- Identifiziert, aus welchen Automation-Runbooks Sie wählen können, um die angeforderte Änderung vorzunehmen

In einigen Fällen kann eine Änderungsvorlage konfiguriert werden, sodass Sie Ihr eigenes Automation-Runbook angeben und angeben, wer die Anforderung überprüfen und genehmigen soll.

> **Important**
>
> Wenn Sie den Change Manager in einer Organisation verwenden, empfehlen wir, Änderungen immer über das delegierte Administratorkonto vorzunehmen. Obwohl Sie Änderungen von anderen Konten in der Organisation vornehmen, werden diese Änderungen nicht im delegierten Administratorkonto gemeldet oder können nicht angezeigt werden.

Themen

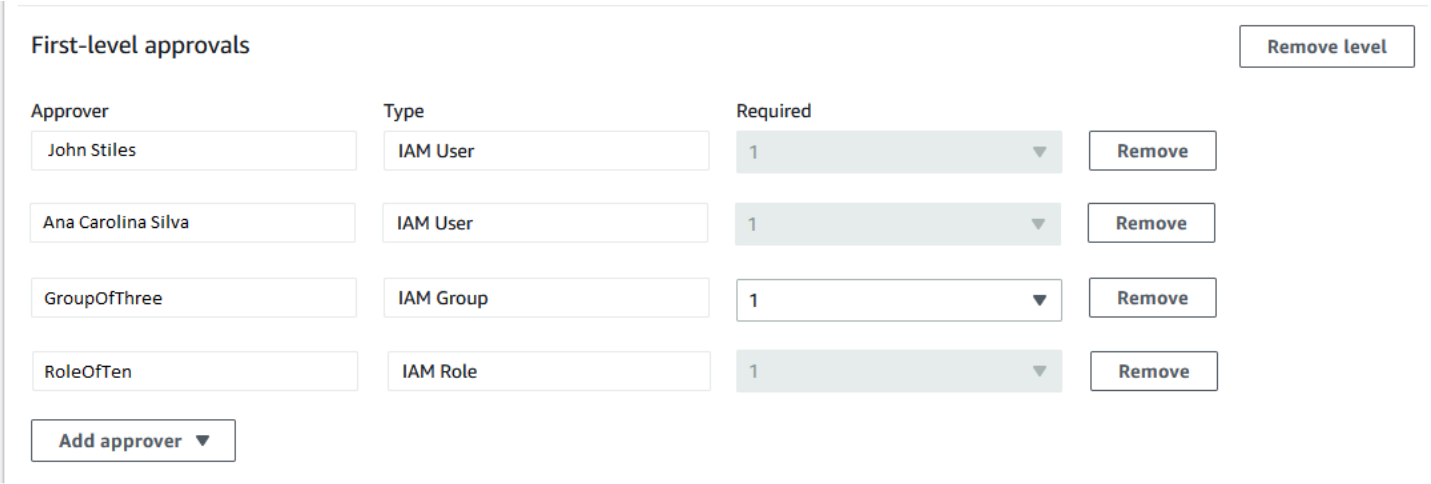
- Über die Genehmigung von Änderungsanfragen
- Erstellen von Änderungsanforderungen (Konsole)
- Erstellen von Änderungsanforderungen (AWS CLI)

Über die Genehmigung von Änderungsanfragen

Abhängig von den in einer Änderungsvorlage festgelegten Anforderungen können Änderungsanfragen, die Sie auf dieser Grundlage erstellen, Genehmigungen von bis zu fünf Ebenen erfordern, bevor der Runbook-Workflow für die Anfrage ausgeführt werden kann. Für jede dieser Ebenen kann der Ersteller der Vorlage bis zu fünf potenzielle Genehmiger angeben. Ein Genehmiger ist nicht auf einen einzelnen Benutzer beschränkt. Ein Genehmiger in diesem Sinne kann auch eine IAM-Gruppe oder IAM-Rolle sein. Für IAM-Gruppen und IAM-Rollen können ein oder mehrere Benutzer, die zu der Gruppe oder Rolle gehören, Genehmigungen für den Erhalt der Gesamtzahl der Genehmigungen erteilen, die für eine Änderungsanforderung erforderlich sind. Ersteller von Vorlagen können auch mehr Genehmiger angeben, als die Änderungsvorlage erfordert.

Ursprüngliche Genehmigungs-Workflows und aktualisierte und/oder Genehmigungen

Bei Verwendung von Änderungsvorlagen, die vor dem 23. Januar 2023 erstellt wurden, muss eine Genehmigung von jedem angegebenen Genehmiger eingeholt werden, damit die Änderungsanfrage auf dieser Ebene genehmigt werden kann. Beispielsweise sind in der Einrichtung der Genehmigungsebene, die im folgenden Image angezeigt werden, vier Genehmiger angegeben. Zu den angegebenen Genehmigern gehören zwei Benutzer (John Stiles und Ana Carolina Silva), eine Benutzergruppe mit drei Mitgliedern (GroupOfThree) und eine Benutzerrolle, die zehn Benutzern () entspricht. RoleOfTen

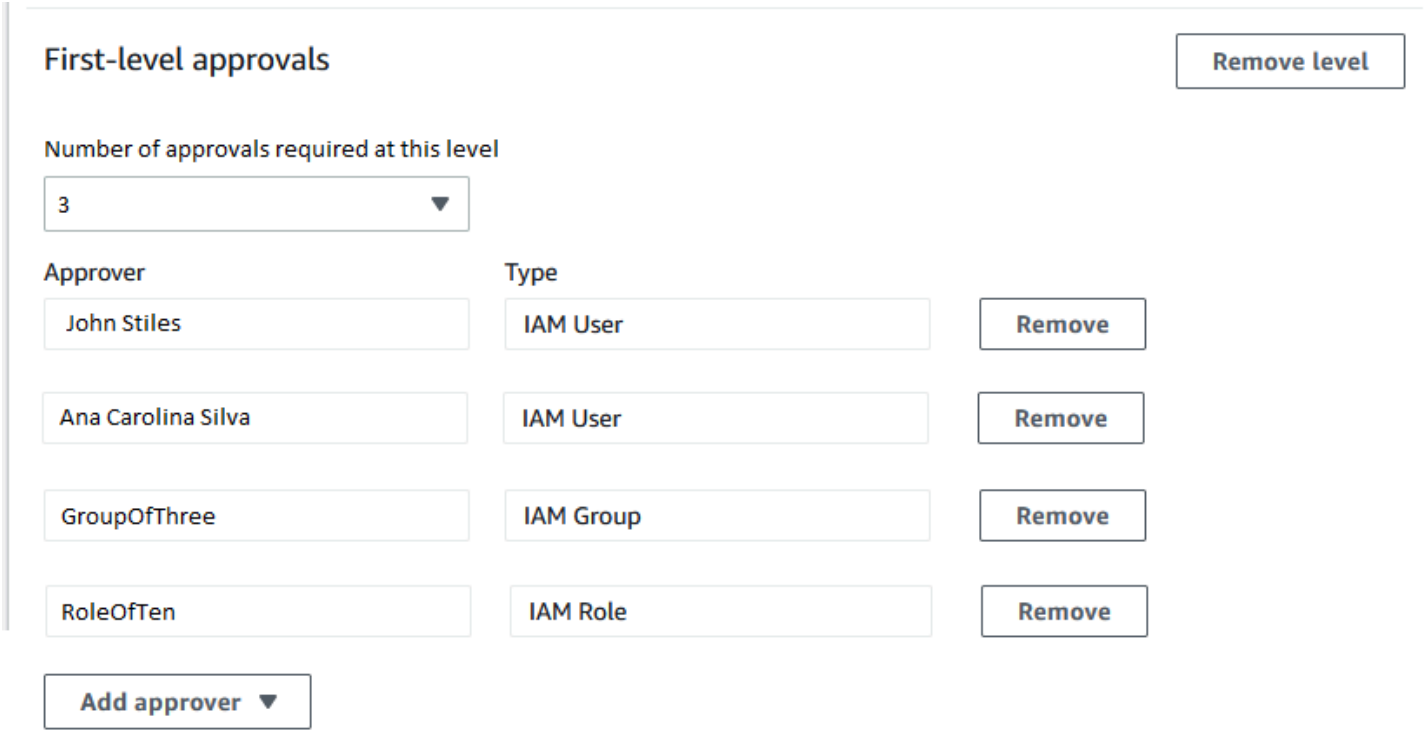


Damit die Änderungsanfrage auf dieser Ebene genehmigt werden kann, muss sie von John Stiles, Ana Carolina Silva, einem Mitglied der `GroupOfThree`-Gruppe und einem Mitglied der `RoleOfTen`-Rolle genehmigt werden.

Unter Verwendung von Änderungsvorlagen, die am oder nach dem 23. Januar 2023 erstellt wurden, können Vorlagenersteller für jede Genehmigungsebene eine Gesamtzahl der erforderlichen

Genehmigungen angeben. Diese Genehmigungen können aus einer beliebigen Kombination von Benutzern, Gruppen und Rollen stammen, die als Genehmiger angegeben sind. Eine Änderungsvorlage könnte nur eine Genehmigung für eine Ebene erfordern, aber beispielsweise zwei einzelne Benutzer, zwei Gruppen und eine Rolle als potenzielle Genehmiger angeben.

Beispielsweise sind in der Einrichtung der Genehmigungsebene, die im folgenden Image angezeigt werden, pro Ebene drei Genehmigungen erforderlich. Zu den angegebenen Genehmigern gehören zwei Benutzer (John Stiles und Ana Carolina Silva), eine Benutzergruppe mit drei Mitgliedern (`GroupOfThree`) und eine Benutzerrolle, die zehn Benutzer repräsentiert (`RoleOfTen`).



Wenn alle drei Benutzer in der `GroupOfThree`-Gruppe die Änderungsanfrage genehmigen, wird er für diese Ebene genehmigt. Es ist nicht erforderlich, eine Genehmigung von jedem Benutzer, Gruppe oder Rolle zu erhalten. Die Mindestanzahl an Genehmigungen kann von einer beliebigen Kombination potenzieller Genehmiger stammen.

Wenn Ihre Änderungsanfrage erstellt wird, werden Benachrichtigungen an die Abonnenten des Amazon-SNS-Themas gesendet, das für Genehmigungsbenachrichtigungen auf dieser Ebene angegeben wurde. Der Ersteller der Änderungsvorlage hat möglicherweise das zu verwendende Benachrichtigungsthema angegeben oder Ihnen erlaubt, eines anzugeben.

Nachdem die Mindestanzahl erforderlicher Genehmigungen auf einer Ebene eingegangen ist, werden Benachrichtigungen an Genehmiger gesendet, die das Amazon-SNS-Thema für die nächste Ebene abonniert haben, und so weiter.

Unabhängig davon, wie viele Genehmigungsebenen und Genehmiger angegeben sind, ist nur eine Ablehnung einer Änderungsanfrage erforderlich, um zu verhindern, dass der Runbook-Workflow für diese Anforderung ausgeführt wird.

Erstellen von Änderungsanforderungen (Konsole)

Im Folgenden wird beschrieben, wie Sie eine Änderungsanforderung mit Hilfe der Systems Manager-Konsole erstellen.

So erstellen Sie eine Änderungsanforderung (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie Create request (Erstellen einer Anfrage).
4. Suchen Sie nach einer Änderungsvorlage, die Sie für diese Änderungsanforderung verwenden möchten, und wählen Sie sie aus.
5. Wählen Sie Weiter aus.
6. Geben Sie für Name einen Namen für die Änderungsanforderung ein, mit der ihre Funktion leicht zu erkennen ist, z. B. **`UpdateEC2LinuxAMI-us-east-2`**.
7. Wählen Sie für Runbook das Runbook aus, das Sie für die gewünschte Änderung verwenden möchten.

   > **Note**
   >
   > Wenn die Option zum Auswählen eines Runbooks nicht verfügbar ist, hat der Autor der Änderungsvorlage angegeben, welches Runbook verwendet werden muss.

8. Verwenden Sie für Change request information (Informationen zu Änderungsanforderung) Markdown, um zusätzliche Informationen zur Änderungsanforderung bereitzustellen, damit Prüfer entscheiden können, ob sie die Änderungsanforderung genehmigen oder ablehnen möchten. Der Autor der Vorlage, die Sie verwenden, hat möglicherweise Anweisungen oder Fragen zur Beantwortung bereitgestellt.

> **Note**
>
> Markdown ist eine Markup-Sprache, die es Ihnen ermöglicht, Dokumente und einzelne Schritte innerhalb des Dokuments mit Beschreibungen im Wiki-Stil zu versehen. Weitere Informationen zur Verwendung von Markdown finden Sie unter Verwenden von Markdown in AWS.

9. Wählen Sie im Abschnitt Workflow start time (Workflow-Startzeit) eine der folgenden Optionen:

   - Run the operation at a scheduled time (Ausführen des Vorgangs zu einem geplanten Zeitpunkt) - Geben Sie für Requested start time (Gewünschte Startzeit) das Datum und die Uhrzeit ein, die Sie für die Ausführung des Runbook-Workflows für diesen Auftrag vorschlagen. Geben Sie bei Estimated end time (Geschätzte Endzeit) das Datum und die Uhrzeit ein, zu der der Runbook-Workflow voraussichtlich abgeschlossen sein wird. (Diese Zeit ist nur eine Schätzung, die Sie Prüfern zur Verfügung stellen.)

     > **Tip**
     >
     > Wählen Sie View Change Calendar (Änderungskalender anzeigen), um zu prüfen, ob für die von Ihnen angegebene Zeit Sperrungen vorliegen.

   - Führen Sie den Vorgang so schnell wie möglich nach der Genehmigung aus. - Wenn die Änderungsanforderung genehmigt wird, wird der Runbook-Workflow ausgeführt, sobald ein Zeitraum nicht eingeschränkt ist, in dem Änderungen vorgenommen werden können.

10. Im Abschnitt Change request approvals (Genehmigungen für Änderungsanträge) gehen Sie wie folgt vor:

    1. Wenn Approval type (Genehmigungstyp)-Optionen angezeigt werden, wählen Sie eine der folgenden Optionen:

       - Automatic approval (Automatische Genehmigung) - Die ausgewählte Änderungsvorlage ist so konfiguriert, dass Änderungsanforderungen automatisch ausgeführt werden können - ohne Prüfung durch Genehmiger. Fahren Sie fort mit Schritt 11.

         > **Note**
         >
         > Die in den IAM-Richtlinien angegebenen Berechtigungen, die Ihre Verwendung von Systems Manager regeln, dürfen Sie nicht daran hindern, Änderungsanforderungen

zur automatischen Genehmigung zu übermitteln, damit sie automatisch ausgeführt werden können.

- Specify approvers (Angeben von Genehmigern) - Sie müssen einen oder mehrere Benutzer, Gruppen oder IAM-Rollen hinzufügen, um diese Änderungsanforderung zu überprüfen und zu genehmigen.

> **Note**
>
> Sie können Überprüfer auch dann angeben, wenn die in den IAM-Richtlinien, die Ihre Verwendung von Systems Manager regeln, festgelegten Berechtigungen die Ausführung von Änderungsanforderungen mit automatischer Genehmigung erlauben.

2. Wählen Sie Genehmiger hinzufügen und wählen Sie dann einen oder mehrere Benutzer, Gruppen oder AWS Identity and Access Management (IAM-) Rollen aus der Liste der verfügbaren Prüfer aus.

> **Note**
>
> Möglicherweise sind bereits ein oder mehrere Genehmiger angegeben. Dies bedeutet, dass obligatorische Genehmiger bereits in der von Ihnen ausgewählten Änderungsvorlage angegeben sind. Diese Genehmiger können nicht aus der Anforderung entfernt werden. Wenn die Schaltfläche Genehmiger hinzufügen nicht verfügbar ist, lässt die von Ihnen ausgewählte Vorlage nicht zu, dass zusätzliche Prüfer zu Anfragen hinzugefügt werden.

Weitere Informationen zum Genehmigen von Änderungsanfragen finden Sie unter Über die Genehmigung von Änderungsanfragen.

3. Wählen Sie unter SNS topic to notify approvers (SNS-Thema zur Benachrichtigung von Genehmigern) eine der folgenden Optionen, um das Amazon SNS-Thema in Ihrem Konto anzugeben, das zum Senden von Benachrichtigungen an die Genehmigenden verwendet werden soll, die Sie zu dieser Änderungsanforderung hinzufügen.

> **Note**
>
> Wenn die Option zum Angeben eines Amazon SNS-Themas nicht verfügbar ist, gibt die von Ihnen ausgewählte Änderungsvorlage bereits das zu verwendende Amazon SNS-Thema an.

- Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.
- Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen Konto. (Diese Option ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)

> **Note**
>
> Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

4. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

11. Wählen Sie Weiter aus.
12. Wählen Sie für IAM role (IAM-Rolle) eine IAM-Rolle in Ihrem aktuellen Konto aus, die über die erforderlichen Berechtigungen zur Ausführung der Runbooks verfügt, die für diese Änderungsanforderung angegeben sind.

    Diese Rolle wird auch als Dienstrolle bezeichnet oder Übernahmerolle für Automation. Weitere Informationen über diese Rolle finden Sie unter Einrichten der Automatisierung.
13. Wählen Sie im Abschnitt Deployment location (Standort der Bereitstellung) eine der folgenden Optionen:

> **Note**
>
> Wenn Sie AWS-Konto nur Change Manager mit einer einzigen Person und nicht mit einer Organisation AWS Organizations, in der Sie eingerichtet sind, verwenden, müssen Sie keinen Bereitstellungsort angeben.

- Apply change to this account (Änderung auf dieses Konto anwenden)— Der Runbook-Workflow wird nur im aktuellen Konto ausgeführt. Für eine Organisation bedeutet dies das delegierte Administratorkonto.
- Apply change to multiple organizational units (OUs) (Anwenden von Änderungen auf mehrere Organisationseinheiten) - Gehen Sie wie folgt vor:
  1. Für Accounts and organizational units (OUs) (Konten und Organisationseinheiten) geben Sie die ID eines Mitgliedskontos in Ihrer Organisation im Format **123456789012** oder die ID einer Organisationseinheit im Format **o-o96EXAMPLE** ein.
  2. (Optional) Geben Sie für Execution role name (Name der Ausführungsrolle) den Namen der IAM-Rolle im Zielkonto oder Organisationseinheit ein, die über die erforderlichen Berechtigungen zum Ausführen der Runbooks verfügt, die für diese Änderungsanforderung angegeben sind. Alle Konten in einer von Ihnen angegebenen Organisationseinheit sollten für diese Rolle denselben Namen verwenden.
  3. (Optional) Wählen Sie Add another target location (Weiteres Ziel hinzufügen) für jedes zusätzliche Konto oder jede Organisationseinheit, die Sie angeben möchten, und wiederholen Sie die Schritte a und b.
  4. Wählen Sie für Target die Region aus AWS-Region, in der die Änderung vorgenommen werden soll, z. B. `Ohio (us-east-2)` für die Region USA Ost (Ohio).
  5. Erweitern Sie den Reiter Rate control (Ratenregelung).

     Geben Sie für Concurrency (Gleichzeitigkeit) eine Zahl ein, und wählen Sie dann aus der Liste aus, ob dies die Anzahl oder den Prozentsatz der Konten darstellt, in denen der Runbook-Workflow gleichzeitig ausgeführt werden kann.

     Geben Sie für Error threshold (Schwellenwert-Fehler) eine Zahl ein und wählen Sie dann aus der Liste aus, ob dies die Anzahl oder den Prozentsatz der Konten darstellt, bei denen der Runbook-Workflow fehlschlagen kann, bevor der Vorgang beendet wird.

14. Gehen Sie im Abschnitt Deployment targets (Bereitstellungsziele) wie folgt vor:

1. Wählen Sie eine der folgenden Optionen aus:
   - Single resource (Einzelne Ressource) - Die Änderung soll nur für eine Ressource vorgenommen werden. Zum Beispiel, ein einzelner Knoten oder ein einzelnes Amazon Machine Image (AMI), je nach der in den Runbooks für diesen Änderungsauftrag definierten Operation.
   - Multiple resources (Mehrere Ressourcen) - Wählen Sie für Parameter die verfügbaren Parameter aus den Runbooks für diesen Änderungsauftrag aus. Diese Auswahl spiegelt den Typ der Ressource wider, die aktualisiert wird.

     Wenn das Runbook für diese Änderungsanforderung beispielsweise `AWS-RetartEC2Instance` ist, können Sie `InstanceId` wählen und dann festlegen, welche Instances aktualisiert werden, indem Sie eine der folgenden Optionen auswählen:
     - Specify tags (Tags angeben) - Geben Sie ein Schlüssel-Wert-Paar ein, mit dem alle zu aktualisierenden Ressourcen getaggt werden.
     - Choose a resource group (Eine Ressourcengruppe auswählen) - Wählen Sie den Namen der Ressourcengruppe aus, zu der alle zu aktualisierenden Ressourcen gehören.
     - Specify parameter values (Parameterwerte angeben) - Identifizieren Sie die zu aktualisierenden Ressourcen im Abschnitt Runbook parameters (Runbook-Parameter).
     - Target all instances (Alle Instances anvisieren) – Nehmen Sie die Änderung für alle verwalteten Knoten an den Zielorten vor.
2. Wenn Sie Multiple resources (Mehrere Ressourcen) wählen, erweitern Sie Rate control (Ratenregelung).

   Geben Sie für Concurrency (Gleichzeitigkeit) eine Zahl ein und wählen Sie dann aus der Liste aus, ob dies die Anzahl oder den Prozentsatz der Ziele darstellt, die der Runbook-Workflow gleichzeitig aktualisieren kann.

   Geben Sie für Error threshold (Schwellenwert-Fehler) eine Zahl ein und wählen Sie dann aus der Liste aus, ob dies die Anzahl oder den Prozentsatz der Ziele darstellt, bei denen die Aktualisierung fehlschlagen kann, bevor der Vorgang beendet wird.

15. Wenn Sie Specify parameter values (Parameterwerte angeben) gewählt haben, um mehrere Ressourcen im vorherigen Schritt zu aktualisieren: Geben Sie im Abschnitt Runbook parameters (Runbook-Parameter die Werte für die erforderlichen Eingabeparameter an. Die Parameterwerte, die Sie angeben müssen, basieren auf dem Inhalt der Automation-Runbooks, die der ausgewählten Änderungsvorlage zugeordnet sind.

Wenn die Änderungsvorlage beispielsweise das `AWS-RetartEC2Instance` Runbook verwendet, müssen Sie eine oder mehrere Instanz-IDs für den InstanceIdParameter eingeben. Alternativ können Sie Show interactive instance picker (Interaktive Instance-Auswahl anzeigen) wählen und nachher die verfügbaren Instance einzeln auswählen.

16. Wählen Sie Weiter aus.
17. Überprüfen Sie auf der Seite Review and submit (Überprüfen und Einreichen) die Ressourcen und Optionen, die Sie für diesen Änderungsantrag angegeben haben.

    Wählen Sie die Schaltfläche Bearbeiten für jeden Abschnitt, an dem Sie Änderungen vornehmen möchten.

    Wenn Sie mit den Details zur Änderungsanforderung zufrieden sind, klicken Sie auf Submit for approval (Zur Genehmigung einreichen).

Wenn in der Änderungsvorlage, die Sie für die Anfrage ausgewählt haben, ein Amazon SNS-Thema angegeben wurde, werden Benachrichtigungen gesendet, wenn die Anfrage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen für die Anfrage erhalten, können Sie zu Change Manager zurückkehren, um den Status Ihrer Anfrage zu überprüfen.

Erstellen von Änderungsanforderungen (AWS CLI)

Sie können mithilfe von AWS Command Line Interface (AWS CLI) eine Änderungsanforderung erstellen, indem Sie Optionen und Parameter für die Änderungsanforderung in einer JSON-Datei angeben und die `--cli-input-json` Option verwenden, um sie in Ihren Befehl aufzunehmen.

So erstellen Sie eine Änderungsanforderung (AWS CLI)

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS Tools for PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.
2. Erstellen Sie eine JSON-Datei auf Ihrem lokalen Computer und geben Sie Ihr beispielsweise einen Namen wie `MyChangeRequest.json`. Fügen Sie der Datei anschließend den folgenden Inhalt ein:

   `placeholders (Platzhalter)` mit Werten für Ihre Änderungsanforderung ersetzen.

> **Note**
>
> Dieser Beispiel-JSON-Code erstellt eine Änderungsanforderung mithilfe der `AWS-HelloWorldChangeTemplate`-Änderungsvorlage und dem `AWS-HelloWorld`-Runbook. Damit Sie dieses Beispiel für Ihre eigenen Änderungsanfragen anpassen können, finden Sie unter StartChangeRequestExecution in der AWS Systems Manager - API-Referenz Informationen zu allen verfügbaren Parametern
> Weitere Informationen zum Genehmigen von Änderungsanfragen finden Sie unter Über die Genehmigung von Änderungsanfragen.

```
{
    "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. Führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, den folgenden Befehl aus.

```
aws ssm start-change-request-execution --cli-input-json file://MyChangeRequest.json
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "AutomationExecutionId": "b3c1357a-5756-4839-8617-2d2a4EXAMPLE"
}
```

Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen

Wenn Sie als Prüfer für einen Änderungsantrag in angegeben sindChange Manager, eine Funktion von AWS Systems Manager, werden Sie über ein Amazon Simple Notification Service (Amazon SNS) -Thema benachrichtigt, wenn ein neuer Änderungsantrag auf Ihre Überprüfung wartet.

> **Note**
>
> Diese Funktion hängt davon ab, ob ein Amazon SNS in der Änderungsvorlage für das Senden von Überprüfungsbenachrichtigungen angegeben wurde. Weitere Informationen

finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

Um den Änderungsantrag zu überprüfen, können Sie dem Link in Ihrer Benachrichtigung folgen oder sich AWS Management Console direkt bei der anmelden und die Schritte in diesem Verfahren befolgen.

> **Note**
>
> Wenn ein Amazon SNS-Thema für Prüfer in einer Änderungsvorlage zugewiesen ist, werden Benachrichtigungen an die Abonnenten des Themas gesendet, wenn sich die Änderungsanforderung ändert.
> Weitere Informationen zum Genehmigen von Änderungsanfragen finden Sie unter Über die Genehmigung von Änderungsanfragen.

Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen (Konsole)

Die folgenden Verfahren beschreiben, wie Sie die Systems-Manager-Konsole verwenden, um Änderungsanforderungen zu überprüfen und zu genehmigen oder abzulehnen.

Überprüfen und Genehmigen oder Ablehnen eines einzigen Änderungsantrags

1. Öffnen Sie den Link in der E-Mail-Benachrichtigung, die Sie erhalten haben, und melden Sie sich bei der an AWS Management Console, wodurch Sie zur Änderungsanfrage weitergeleitet werden, die Sie überprüfen können.
2. Überprüfen Sie auf der Übersichtsseite den vorgeschlagenen Inhalt der Änderungsanforderung.

   Um die Änderungsanforderung zu genehmigen, wählen Sie Approve (Genehmigen). Geben Sie im Dialogfeld alle Kommentare ein, die Sie für diese Genehmigung hinzufügen möchten, und wählen Sie dann Approve (Genehmigen). Der durch diese Anforderung dargestellte Runbook-Workflow beginnt entweder nach der Planung oder sobald Änderungen nicht durch Einschränkungen blockiert sind.

   –oder–

Um die Änderungsanforderung abzulehnen, wählen Sie Reject (Ablehnen). Geben Sie im Dialogfeld alle Kommentare ein, die Sie für diese Ablehnung hinzufügen möchten, und wählen Sie dann Reject (Ablehnen).

Überprüfen und Genehmigen oder Ablehnen von Änderungsanträgen auf einmal

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie die Registerkarte Approvals (Genehmigungen) aus.
4. (Optional) Überprüfen Sie die Details der Anfragen, für die Ihre Genehmigung aussteht, indem Sie den Namen jeder Anfrage auswählen und dann zur Registerkarte Approvals (Genehmigungen) zurückkehren.
5. Aktivieren Sie das Kontrollkästchen jeder Änderungsanforderung, die Sie genehmigen möchten.

   –oder–

   Aktivieren Sie das Kontrollkästchen jeder Änderungsanforderung, die Sie ablehnen möchten.
6. Geben Sie im Dialogfeld alle Kommentare ein, die Sie für diese Genehmigung oder Ablehnung hinzufügen möchten.
7. Je nachdem, ob Sie die ausgewählten Änderungsanträge genehmigen oder ablehnen, wählen Sie Approve (Genehmigen) oder Reject (Ablehnen) aus.

Überprüfen und Genehmigen oder Ablehnen einer Änderungsanforderung (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die AWS Command Line Interface (AWS CLI) (unter Linux, oder Windows) verwendenmacOS, um eine Änderungsanforderung zu überprüfen und zu genehmigen oder abzulehnen.

Überprüfen und Genehmigen oder Ablehnen eines Änderungsantrags

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Erstellen Sie auf Ihrem lokalen Computer eine JSON-Datei, die die Parameter für Ihren AWS CLI Aufruf angibt.

```
{
  "OpsItemFilters":
  [
    {
      "Key": "OpsItemType",
      "Values": ["/aws/changerequest"],
      "Operator": "Equal"
    }
  ],
  "MaxResults": number
}
```

Sie können die Ergebnisse für einen bestimmten Genehmiger filtern, indem Sie den Amazon Resource Name (ARN) des Genehmigers in der JSON-Datei angeben. Ein Beispiel.

```
{
  "OpsItemFilters":
  [
    {
      "Key": "OpsItemType",
      "Values": ["/aws/changerequest"],
      "Operator": "Equal"
    },
    {
      "Key": "ChangeRequestByApproverArn",
      "Values": ["arn:aws:iam::account-id:user/user-name"],
      "Operator": "Equal"
    }
  ],
  "MaxResults": number
}
```

3. Führen Sie den folgenden Befehl aus, um die maximale Anzahl von Änderungsanforderungen anzuzeigen, die Sie in der JSON-Datei angegeben haben.

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. Führen Sie den folgenden Befehl aus, um eine Änderungsanforderung zu genehmigen oder abzulehnen.

   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"
   ```

   Wenn in der Änderungsvorlage, die Sie für die Anfrage ausgewählt haben, ein Amazon SNS-Thema angegeben wurde, werden Benachrichtigungen gesendet, wenn die Anfrage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen für die Anfrage erhalten, können Sie zu Change Manager zurückkehren, um den Status Ihrer Anfrage zu überprüfen. Informationen zu anderen Optionen bei Verwendung dieses Befehls finden Sie unter send-automation-signal im AWS Systems Manager -Abschnitt der AWS CLI -Befehlsreferenz.

Überprüfen von Details, Aufgaben und Zeitplänen für Änderungsanforderungen (Konsole)

Im Dashboard von Change Manager, eine Funktion von AWS Systems Manager, können Sie Informationen zu einem Änderungsauftrag anzeigen, einschließlich der Aufträge, für die bereits Änderungen bearbeitet wurden. Diese Details enthalten einen Link zur Automatisierungs-Operation, mit der die Runbooks ausgeführt werden, die die Änderung vornehmen. Eine Automation-

Ausführungs-ID wird generiert, wenn die Anforderung erstellt wird. Der Prozess wird jedoch erst ausgeführt, wenn alle Genehmigungen erteilt wurden und keine Einschränkungen vorhanden sind, um die Änderung zu blockieren.

Überprüfung von Details, Aufgaben und Zeitplänen für Änderungsanträge

1. Wählen Sie im Navigationsbereich Change Manager aus.
2. Wählen Sie die Registerkarte Requests (Anforderungen).
3. Suchen Sie im Abschnitt Change requests (Änderungsanforderungen) nach der Änderungsanforderung, die Sie überprüfen möchten.

   Sie können die Option Create date range (Datumsbereich erstellen) verwenden, um die Ergebnisse auf einen bestimmten Zeitraum zu beschränken.

   Sie können Anforderungen nach folgenden Eigenschaften filtern:

   - `Status`
   - `Request ID`
   - `Approver`
   - `Requester`

   Führen Sie beispielsweise die folgenden Schritte aus, um Details zu allen Änderungsanforderungen anzuzeigen, die in den letzten 24 Stunden erfolgreich abgeschlossen wurden:

   1. Wählen Sie für Create date range (Datumsbereich erstellen) 1d.
   2. Wählen Sie im Suchfeld Status, aus CompletedWithSuccess.
   3. Wählen Sie in den Ergebnissen den Namen der erfolgreich abgeschlossenen Änderungsanforderung aus, für die die Ergebnisse überprüft werden sollen.
4. Zeigen Sie Informationen zur Änderungsanforderung auf den folgenden Registerkarten an:

   - Request details (Details anfordern) - Zeigt grundlegende Details zur Änderungsanforderung an, einschließlich des Anforderers, der Änderungsvorlage und der Automation-Runbooks, die für die Änderung ausgewählt wurden. Sie können auch einem Link zu den Details des Automatisierungsvorgangs folgen und Informationen zu allen in der Anfrage angegebenen Runbook-Parametern, den dem Änderungsantrag zugewiesenen CloudWatch Amazon-

Alarmen sowie zu den Genehmigungen und Kommentaren, die für die Anfrage bereitgestellt wurden, einsehen.

- Task (Aufgabe) - Zeigt Informationen über die Aufgabe in der Änderung an, einschließlich des Aufgabenstatus für abgeschlossene Änderungsanforderungen, der Zielressourcen, der Schritte in den zugeordneten Automation-Runbooks sowie Details zum Parallelitäts- und Fehlerschwellenwert.
- Timeline (Zeitplan)- Zeigt eine Zusammenfassung aller Ereignisse an, die mit dem Änderungsauftrag verknüpft sind, nach Datum und Uhrzeit. Die Zusammenfassung gibt an, wann die Änderungsanforderung erstellt wurde, welche Aktionen von den zugewiesenen Genehmigern durchgeführt wurden, wann die genehmigten Änderungsanforderungen ausgeführt werden sollen, wie der Workflow des Runbooks aussieht und welche Statusänderungen für den gesamten Änderungsprozess und die einzelnen Schritte im Runbook vorgenommen wurden.
- Associated events (Zugeordnete Ereignisse) – Zeigen Sie überprüfbare Details zu Änderungsanfragen an, die in AWS CloudTrail Lake aufgezeichnet wurden. Zu den Details gehören die ausgeführten API-Aktionen, die für diese Aktionen enthaltenen Anforderungsparameter, das Benutzerkonto, das die Aktion ausgeführt hat, die während des Prozesses aktualisierten Ressourcen und mehr.

  Wenn Sie die CloudTrail Lake-Ereignisverfolgung aktivieren, erstellt CloudTrail Lake einen Ereignisdatenspeicher für Ereignisse im Zusammenhang mit Ihren Änderungsanforderungen. Die Ereignisdetails sind für das Konto oder die Organisation verfügbar, für die die Änderungsanfrage ausgeführt wurde. Sie können die CloudTrail Lake-Ereignisverfolgung von jeder Änderungsanfrage in Ihrem Konto oder Ihrer Organisation aus aktivieren. Informationen zur Aktivierung der CloudTrail Lake-Integration und zum Erstellen eines Ereignisdatenspeichers finden Sie unterÜberwachung der Ereignisse Ihrer Änderungsanfragen.

  > **Note**
  >
  > Die Nutzung von CloudTrail Lake ist kostenpflichtig. Weitere Details finden Sie unter AWS CloudTrail -Preise.

Aggregierte Anzahl von Änderungsaufträgen anzeigen (Befehlszeile)

Anzeigen der aggregierten Anzahl von Änderungsanforderungen in Change Manager, eine Funktion von AWS Systems Manager, mithilfe des GetOpsSummary API-Vorgangs. Dieser API-Vorgang kann Zählungen für einen einzelnen AWS-Konto in einer einzigen AWS-Region oder für mehrere Konten und Regionen ausgeben.

> **Note**
>
> Wenn Sie aggregierte Anzahl von Änderungsaufträgen für mehrere AWS-Konten und mehrere AWS-Regionen anzeigen möchten, müssen Sie eine Ressourcendaten-Synchronisierung einrichten und konfigurieren. Weitere Informationen finden Sie unter Konfigurieren von Resource Data Sync für Inventory.

Im Folgenden wird beschrieben, wie Sie die AWS Command Line Interface (AWS CLI) (unter Linux, macOS, oder Windows) verwenden, um die aggregierte Anzahl der Änderungsanforderungen anzuzeigen.

Anzeigen der aggregierten Anzahl von Änderungsanforderungen

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie einen der folgenden Befehle aus.

   Single account and Region (Einzelnes Konto und Region)

   Dieser Befehl gibt eine Anzahl aller Änderungsanforderungen für den AWS-Konto und die AWS-Region an, für die Ihre AWS CLI-Sitzung konfiguriert ist.

   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
```

Das Aufruf gibt unter anderem folgende Informationen zurück.

```
{
    "Entities": [
        {
            "Data": {
                "AWS:OpsItem": {
                    "Content": [
                        {
                            "Count": "38",
                            "Status": "Open"
                        }
                    ]
                }
            }
        }
    ]
}
```

Multiple accounts and/or Regions (Mehrere Konten und/oder Regionen)

Dieser Befehl gibt eine Anzahl aller Änderungsanforderungen für die AWS-Konten und AWS-Regionen aus, die in der Ressourcendatensynchronisation angegeben sind.

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
```

Das Aufruf gibt unter anderem folgende Informationen zurück.

```
{
    "Entities": [
        {
            "Data": {
                "AWS:OpsItem": {
                    "Content": [
                        {
                            "Count": "43",
                            "Status": "Open"
                        },
                        {
                            "Count": "2",
                            "Status": "Resolved"
                        }
                    ]
                }
            }
        }
    ]
}
```

Multiple accounts and a specific Region (Mehrere Konten und eine spezifische Region

Dieser Befehl gibt eine Anzahl aller Änderungsanforderungen für die AWS-Konten aus, die in der Ressourcendatensynchronisation angegeben sind. Er gibt jedoch nur Daten aus der Region aus, die im Befehl angegeben ist.

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
```

Multiple accounts and Regions with output grouped by Region (Mehrere Konten und Regionen mit Ausgabe gruppiert nach Region)

Dieser Befehl gibt eine Anzahl aller Änderungsanforderungen für die AWS-Konten und AWS-Regionen aus, die in der Ressourcendatensynchronisation angegeben sind. Die Ausgabe zeigt die Zählinformationen pro Region an.

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"}]}]'
```

Das Aufruf gibt unter anderem folgende Informationen zurück.

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

Multiple accounts and Regions with output grouped by accounts and Regions (Mehrere Konten und Regionen mit Ausgabe gruppiert nach Konten und Regionen)

Dieser Befehl gibt eine Anzahl aller Änderungsanforderungen für die AWS-Konten und AWS-Regionen aus, die in der Ressourcendatensynchronisation angegeben sind. Die Ausgabe gruppiert die Zählinformationen nach Konten und Regionen.

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"}]}]}]
```

Das Aufruf gibt unter anderem folgende Informationen zurück.

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

Prüfen und Protokollieren von Change Manager-Aktivitäten

Sie können Aktivitäten in Change Manager, eine Funktion von AWS Systems Manager, mithilfe von Amazon CloudWatch und AWS CloudTrail-Alarmen prüfen.

Weitere Informationen zu den Prüfungs- und Protokollierungsoptionen für Systems Manager finden Sie unter Überwachung AWS Systems Manager.

Prüfen von Change Manager-Aktivität mithilfe von Amazon CloudWatch-Alarmen.

Sie können einer Änderungsvorlage einen CloudWatch-Alarm konfigurieren und zuweisen. Wenn im Alarm definierte Bedingungen erfüllt sind, werden die für den Alarm angegebenen Aktionen ausgeführt. In der Alarmkonfiguration können Sie ein Amazon Simple Notification Service (Amazon SNS)-Thema angeben, um zu benachrichtigen, wenn eine Alarmbedingung erfüllt ist.

Weitere Informationen zum Erstellen einer Change Manager-Vorlage finden Sie unter Arbeiten mit Änderungsvorlagen.

Informationen zum Erstellen von CloudWatch-Alarmen finden Sie unter Verwendung von CloudWatch-Alarmen im Benutzerhandbuch zu Amazon CloudWatch.

Change Manager-Aktivität mithilfe von CloudTrail prüfen

CloudTrail erfasst API-Aufrufe, die über die Systems Manager-Konsole erfolgt sind, dieAWS Command Line Interface (AWS CLI) und das Systems Manager SDK. Sie können die Informationen in der CloudTrail-Konsole oder in einem Amazon Simple Storage Service (Amazon S3)-Bucket anzeigen, in dem sie gespeichert sind. Für alle CloudTrail-Protokolle in Ihrem Konto wird nur ein Bucket benötigt.

Protokolle von Change Manager-Aktionen zeigen die Erstellung von Änderungsvorlagendokumenten, Genehmigungen und Ablehnungen von Änderungsvorlagen und Änderungsanforderungen, von Automation-Runbooks generierte Aktivitäten und vieles mehr. Weitere Informationen zum Anzeigen und Verwenden von CloudTrail-Protokollen von Systems Manager-Aktivitäten finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.

Fehlerbehebung für Change Manager

Verwenden Sie die folgenden Informationen bei der Behebung von Problemen mit Change Manager, eine Funktion von AWS Systems Manager.

Themen

- Fehler „Group {GUID} not found“ (Gruppe {GUID} nicht gefunden) bei Änderungsanforderungsgenehmigungen bei Verwendung von Active Directory (Gruppen)

Fehler „Group ***{GUID}*** not found“ (Gruppe {GUID} nicht gefunden) bei Änderungsanforderungsgenehmigungen bei Verwendung von Active Directory (Gruppen)

Problem: Wenn AWS IAM Identity Center (IAM Identity Center) für die Benutzeridentitätsverwaltung verwendet wird, erhält ein Mitglied einer Active-Directory-Gruppe, dem Genehmigungsberechtigungen in Change Manager erteilt wurden, den Fehler „nicht autorisiert“ oder „Gruppe nicht gefunden“.

- Lösung: Wenn Sie Active-Directory-Gruppen in IAM Identity Center für den Zugriff auf AWS Management Console auswählen, plant das System eine regelmäßige Synchronisierung, die Informationen aus diesen Active-Directory-Gruppen in IAM Identity Center kopiert. Dieser Vorgang muss abgeschlossen werden, bevor Benutzer, die über die Active Directory-Gruppenmitgliedschaft autorisiert sind, eine Anforderung erfolgreich genehmigen können. Weitere Informationen finden Sie unter Mit Ihrem Microsoft-AD-Verzeichnis verbinden im AWS IAM Identity Center-Benutzerhandbuch.

AWS Systems Manager-Automatisierung

Automation, eine Funktion von AWS Systems Manager, vereinfacht häufige Wartungs-, Bereitstellungs- und Fehlerbehebungsaufgaben für AWS-Services wie Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS), Amazon Redshift und Amazon Simple Storage Service (Amazon S3) und viele mehr. Um mit der Automatisierung zu beginnen, öffnen Sie die Systems-Manager-Konsole. Klicken Sie im Navigationsbereich auf Automation.

Automation hilft Ihnen, automatisierte Lösungen für die Bereitstellung, Konfiguration und Verwaltung von AWS-Ressourcen im großen Umfang zu entwickeln. Mit Automation haben Sie eine detaillierte Kontrolle über die Nebenläufigkeit der Automatisierungen. So können Sie zum Beispiel angeben, wie viele Ressourcen gleichzeitig verarbeitet werden sollen und wie viele Fehler auftreten können, bevor eine Automatisierung gestoppt wird.

Um die ersten Schritte mit Automation zu erleichtern, entwickelt und pflegt AWS mehrere vordefinierte Runbooks. Je nach Anwendungsfall können Sie diese vordefinierten Runbooks nutzen, die verschiedene Aufgaben ausführen, oder eigene benutzerdefinierte Runbooks erstellen, die Ihren Anforderungen besser entsprechen. Um den Fortschritt und den Status der Automatisierungen zu überwachen, können Sie die Systems-Manager-Automation-Konsole oder Ihr bevorzugtes Befehlszeilen-Tool nutzen. Automation lässt sich auch in Amazon integrieren EventBridge , um Ihnen zu helfen, ereignisgesteuerte Architekturen in großem Umfang zu erstellen.

Wie kann meine Organisation von Automation profitieren?

Automation bietet die folgenden Vorteile:

- Unterstützung der Skripterstellung in Runbook-Inhalten

Mit der `aws:executeScript` Aktion können Sie benutzerdefinierte Python- und - PowerShell Funktionen direkt aus Ihren Runbooks ausführen. Das bietet Ihnen eine größere Flexibilität beim Erstellen eigener Runbooks, da Sie verschiedene Aufgaben ausführen können, die andere Automation-Aktionen nicht unterstützen. Zudem haben Sie eine bessere Kontrolle über die Logik des Runbooks. Ein Beispiel dafür, wie diese Aktion genutzt werden kann und wie sie zur Verbesserung einer bestehenden automatisierten Lösung beitragen kann, finden Sie unter Erstellen von Automation-Runbooks.

- Ausführen von Automatisierungen in mehreren AWS-Konten und AWS-Regionen von einer zentralen Stelle aus

  Administratoren können über die Systems-Manager-Konsole Automatisierungen für Ressourcen in mehreren Konten und Regionen ausführen.

- Verbesserte Betriebssicherheit

  Administratoren verfügen über eine zentrale Stelle zum Erteilen und Widerrufen des Zugriffs auf Runbooks. Mithilfe von IAM-Richtlinien AWS Identity and Access Management allein können Sie steuern, welche einzelnen Benutzer oder Gruppen in Ihrer Organisation Automation verwenden und auf welche Runbooks sie zugreifen dürfen.

- Automatisieren von häufigen IT-Aufgaben

  Die Automatisierung häufiger Aufgaben kann dazu beitragen, die betriebliche Effizienz zu verbessern, organisatorische Standards durchzusetzen und Bedienfehler zu reduzieren. Mit dem Runbook `AWS-UpdateCloudFormationStackWithApproval` können Sie beispielsweise Ressourcen aktualisieren, die mithilfe einer AWS CloudFormation-Vorlage bereitgestellt wurden. Die Aktualisierung wendet eine neue Vorlage an. Sie können Automation so konfigurieren, dass es eine Genehmigung von einem oder mehreren -Benutzer anfordert, bevor die Aktualisierung beginnt.

- Sichere Ausführung störender Aufgaben auf einmal

  Automation umfasst Funktionen wie etwa Ratensteuerelemente, mit deren Hilfe Sie die Bereitstellung einer Automatisierung in der gesamten Flotte durch Angabe eines Nebenläufigkeits- und eines Fehlerschwellenwerts steuern können. Weitere Informationen zum Arbeiten mit Ratensteuerelementen finden Sie unter Ausführen von Automatisierungen im großen Maßstab.

- Optimieren komplexer Aufgaben

Automation bietet vordefinierte Runbooks, die komplexe und zeitaufwendige Aufgaben wie das Erstellen von goldenen Amazon Machine Images (AMIs) optimieren. Mit den Runbooks `AWS-UpdateLinuxAmi` und `AWS-UpdateWindowsAmi` können Sie zum Beispiel goldene AMIs aus einem Quell-AMI erstellen. Mithilfe dieser Runbooks können Sie benutzerdefinierte Skripte ausführen, bevor und nachdem Updates angewendet werden. Zudem können Sie bestimmte Softwarepakete in die Installation einbeziehen oder daraus ausschließen. Beispiele für die Ausführung dieser Runbooks finden Sie unter Tutorials.

- Definieren von Einschränkungen für Eingaben

  In benutzerdefinierten Runbooks können Einschränkungen definiert werden, um die Werte einzugrenzen, die Automation für einen bestimmten Eingabeparameter akzeptiert. `allowedPattern` zum Beispiel akzeptiert für einen Eingabeparameter nur Werte, die dem von Ihnen definierten regulären Ausdruck entsprechen. Wenn Sie `allowedValues` für einen Eingabeparameter angeben, werden nur die Werte akzeptiert, die Sie im Runbook angegeben haben.

- Protokollieren der Ausgabe von Automatisierungsaktionen in Amazon CloudWatch Logs

  Zur Erfüllung betriebs- oder sicherheitsbezogener Anforderungen in Ihrer Organisation müssen Sie möglicherweise eine Aufzeichnung der während eines Runbooks ausgeführten Skripte bereitstellen. Mit - CloudWatch Protokollen können Sie Protokolldateien aus verschiedenen überwachen, speichern und darauf zugreifenAWS-Services. Sie können die Ausgabe der `aws:executeScript` Aktion an eine CloudWatch Logs-Protokollgruppe zum Debuggen und zur Fehlerbehebung senden. Protokolldaten können mit oder ohne AWS KMS-Verschlüsselung über Ihren KMS-Schlüssel an Ihre Protokollgruppe gesendet werden. Weitere Informationen finden Sie unter Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs.

- Amazon- EventBridge Integration

  Automatisierung wird als Zieltyp in Amazon EventBridge-Regeln unterstützt. Das bedeutet, dass Sie Runbooks mithilfe von Ereignissen auslösen können. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager.

- Gemeinsame Nutzung von bewährten Methoden für Organisationen

  Sie können bewährte Methoden u. a. für das Ressourcenmanagement und Betriebsaufgaben in Runbooks definieren, die Sie in mehreren Konten und Regionen nutzen.

Wer sollte Automation nutzen?

- AWS-Kunden, die ihre betriebliche Effizienz im großen Maßstab verbessern, Fehler im Zusammenhang mit manuellen Eingriffen reduzieren und die Zeit bis zur Lösung häufig auftretender Probleme verkürzen möchten.
- Infrastrukturexperten, die Bereitstellungs- und Konfigurationsaufgaben automatisieren möchten.
- Administratoren, die häufig auftretende Probleme zuverlässig lösen, die Effizienz bei der Fehlerbehebung verbessern und die Anzahl sich wiederholender Vorgänge reduzieren möchten.
- Benutzer, die eine Aufgabe automatisieren möchten, die sie normalerweise manuell ausführen.

Was ist eine Automatisierung?

Eine Automatisierung besteht aus allen Aufgaben, die in einem Runbook definiert sind und vom Automation-Service ausgeführt werden. Automation nutzt die folgenden Komponenten zur Ausführung von Automatisierungen.

Konzept	Details
Automation-Runbook	Ein Systems-Manager-Automatisierungs-Runbook definiert die Automatisierung (die Aktionen, die Systems Manager auf Ihren verwalteten Knoten und AWS-Ressourcen durchführt). Automatisierung umfasst mehrere vordefinierte Runbooks, die Sie verwenden können, um allgemeine Aufgaben wie das Neustarten einer oder mehrerer Amazon-EC 2-Instances oder das Erstellen eines Amazon Machine Image (AMI) auszuführen. Sie können auch eigene Runbooks erstellen. Die Runbooks liegen im YAML- oder JSON-Format vor und enthalten die von Ihnen angegebenen Schritte und Parameter. Die Schritte werden nacheinander ausgeführt. Weitere Informationen finden Sie unter Erstellen Ihrer eigenen Runbooks.

Konzept	Details
	Runbooks sind Systems Manager-Dokumente vom Typ `Automation` , im Gegensatz zu `Command`, `Policy`, `Session`-Dokumenten. Runbooks unterstützen die Schemaversion 0.3. Befehlsdokumente mit Schema-Version 1.2, 2.0 oder 2.2. Richtliniendokumente verwenden die Schemaversion 2.0 oder höher.
Automation-Aktion	Die in einem Runbook definierte Automatisierung umfasst einen oder mehrere Schritte. Jeder Schritt ist einer bestimmten Aktion zugeordnet. Die Aktion bestimmt die Eingaben, das Verhalten und die Ausgaben des Schritts. Die Schritte sind im `mainSteps` -Bereich Ihres Runbooks definiert. Die Automatisierung unterstützt 20 verschiedene Aktionstypen. Weitere Informationen hierzu finden Sie unter Systems Manager Automation Aktionen-Referenz.

Konzept	Details
Automation-Kontingente	Jeder AWS-Konto kann 100 Automatisierungen gleichzeitig ausführen. Dazu gehören untergeordnete Automatisierungen (Automatisierungen, die durch eine andere Automatisierung gestartet werden) und Automatisierungen der Ratenregelung. Wenn Sie versuchen, mehr Automatisierungen auszuführen, fügt Systems Manager die zusätzlichen Automatisierungen zu einer Warteschlange hinzu und zeigt den Status „Pending“ an. Dieses Kontingent kann mithilfe von adaptiver Nebenläufigkeit angepasst werden. Weitere Informationen finden Sie unterZulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst. Informationen zur Ausführung von Automatisierungen finden Sie unter Ausführen von Automatisierungen.
Automatisierungs-Warteschlange	Wenn Sie versuchen, mehr Automatisierungen als das gleichzeitige Automatisierungslimit auszuführen, werden nachfolgende Automatisierungen zu einer Warteschlange hinzugefügt. Jedes AWS-Konto kann 5 000 Automatisierungen in die Warteschlange stellen. Sobald eine Automatisierung abgeschlossen ist (oder einen Terminalstatus erreicht), beginnt die erste Automatisierung in der Warteschlange.

Konzept	Details
Kontingent für Automatisierungen der Ratenregelung	Jeder AWS-Konto kann 25 Automatisierungen der Ratenregelung gleichzeitig ausführen. Wenn Sie versuchen, mehr Automatisierungen der Ratenregelung als das gleichzeitige Limit durchzuführen, fügt Systems Manager der Warteschlange die nachfolgenden Automatisierungen der Ratenregelung hinzu und zeigt den Status „Pending“ . Weitere Informationen über die Ratenregelung-Automatisierungen finden Sie unter Ausführen von Automatisierungen im großen Maßstab.
Kontingent für Automatisierungs-Warteschlange der Ratenregelung	Wenn Sie versuchen, mehr Automatisierungen als das Limit für gleichzeitige Automatisierungen der Ratenregelung auszuführen, werden nachfolgende Automatisierungen zu einer Warteschlange hinzugefügt. Jeder AWS-Konto kann 1 000 Automatisierungen der Ratenregelung in die Warteschlange stellen. Sobald eine Automatisierung abgeschlossen ist (oder einen Terminalstatus erreicht), beginnt die erste Automatisierung in der Warteschlange.

Themen

- Einrichten der Automatisierung
- Ausführen von Automatisierungen
- Planung von Automatisierungen
- Systems Manager Automation Aktionen-Referenz
- Erstellen Ihrer eigenen Runbooks
- Referenz zu Systems Manager Automation
- Tutorials

- Grundlegendes zu Automatisierungsstatus
- Fehlerbehebung für Systems Manager Automation.

Einrichten der Automatisierung

Um Automation, eine Funktion von, einzurichten AWS Systems Manager, müssen Sie den Benutzerzugriff auf den Automation-Service überprüfen und Rollen situationsabhängig konfigurieren, damit der Service Aktionen für Ihre Ressourcen ausführen kann. Außerdem empfiehlt es sich, in den Automation-Einstellungen den adaptiven Nebenläufigkeitsmodus zu aktivieren. Die adaptive Nebenläufigkeit passt Ihr Automatisierungskontingent automatisch an Ihre Anforderungen an. Weitere Informationen finden Sie unter Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst.

Um den ordnungsgemäßen Zugriff auf AWS Systems Manager Automation sicherzustellen, überprüfen Sie die folgenden Anforderungen an Benutzer- und Servicerollen.

Überprüfen des Benutzerzugriffs für Runbooks

Stellen Sie sicher, dass Sie berechtigt sind, Runbooks zu verwenden. Wenn Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle Administratorrechte zugewiesen sind, haben Sie Zugriff auf Systems Manager Automation. Wenn Sie nicht über Administratorrechte verfügen, muss ein Administrator Ihnen die Berechtigung gewähren, indem er die von `AmazonSSMFullAccess` verwaltete Richtlinie oder eine Richtlinie, die vergleichbare Berechtigungen bereitstellt, Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zuweist.

> **Important**
>
> Die IAM-Richtlinie `AmazonSSMFullAccess` erteilt Berechtigungen für Systems Manager Aktionen. Einige Runbooks erfordern jedoch Berechtigungen für andere Services, z. B. das Runbook `AWS-ReleaseElasticIP`, das IAM-Berechtigungen für `ec2:ReleaseAddress` erfordert. Daher müssen Sie die in einem Runbook ausgeführten Aktionen überprüfen, um sicherzustellen, dass Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle die erforderlichen Berechtigungen zum Ausführen der im Runbook enthaltenen Aktionen zugewiesen sind.

Konfigurieren eines Service-Rollenzugriffs (Rolle übernehmen) für Automatisierungen

Automation kann im Kontext einer Service-Rolle initiiert werden (oder Übernahmerolle). Auf diese Weise kann der Service Aktionen in Ihrem Namen ausführen. Wenn Sie keine Übernahmerolle angeben, verwendet Automation den Kontext des Benutzers, der die Automatisierung aufgerufen hat.

In den folgenden Situationen müssen Sie jedoch eine Servicerolle für Automation angeben:

- Wenn Sie die Zugriffsberechtigungen eines Benutzers für eine Ressource einschränken, aber dem Benutzer die Ausführung einer Automatisierung gestatten möchten, der höhere Berechtigungen erfordert. In diesem Szenario können Sie eine Servicerolle mit höheren Berechtigungen erstellen und dem Benutzer das Ausführen der Automatisierung gestatten.
- Wenn Sie eine Systems Manager State Manager-Zuordnung zum Ausführen eines Runbooks erstellen.
- Wenn Sie Vorgänge haben, die voraussichtlich länger als 12 Stunden ausgeführt werden.
- Wenn Sie ein Runbook ausführen, das nicht Amazon gehört und die `aws:executeScript` Aktion verwendet, um eine AWS API-Operation aufzurufen oder auf eine AWS Ressource zu reagieren. Weitere Informationen finden Sie unter Berechtigungen für die Verwendung von Runbooks.

Wenn Sie eine Servicerolle für Automation erstellen müssen, können Sie eine der folgenden Methoden anwenden.

Themen

- Methode 1: Verwenden von AWS CloudFormation zum Konfigurieren einer Servicerolle für Automation
- Methode 2: Konfigurieren von Automation-Rollen mit IAM
- Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst
- Implementieren von Änderungskontrollen für Automatisierung

Methode 1: Verwenden von AWS CloudFormation zum Konfigurieren einer Servicerolle für Automation

Sie können eine Servicerolle für Automation, eine Funktion von AWS Systems Manager, aus einer AWS CloudFormation-Vorlage erstellen. Nachdem Sie die Servicerolle erstellt haben, können Sie die Servicerolle in Runbooks mit dem Parameter `AutomationAssumeRole` angeben.

Erstellen der Servicerolle mit AWS CloudFormation

Erstellen Sie mit den folgenden Schritten die erforderliche AWS Identity and Access Management-(IAM)-Rolle für Systems Manager Automation mithilfe von AWS CloudFormation.

Erstellen der erforderlichen IAM-Rolle

1. Laden Sie die `AWS-SystemsManager-AutomationServiceRole.zip`-Datei herunter und entpacken Sie diese. Dieser Ordner enthält die `AWS-SystemsManager-AutomationServiceRole.yaml` AWS CloudFormation-Vorlagendatei.
2. Öffnen Sie die AWS CloudFormation-Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie Create Stack aus.
4. Wählen Sie im Abschnitt Specify template (Vorlage angeben) die Option Upload a template file (Vorlagendatei hochladen) aus.
5. Wählen Sie Browse (Durchsuchen) und dann die AWS CloudFormation-Vorlagendatei `AWS-SystemsManager-AutomationServiceRole.yaml` aus.
6. Wählen Sie Next (Weiter).
7. Geben Sie auf der Seite Specify Stack details (Stack-Details angeben) im Feld Stack name (Stack-Name) einen Namen ein.
8. Auf der Seite Configure stack options (Stack-Optionen konfigurieren) müssen Sie keine Auswahl treffen. Wählen Sie Next (Weiter).
9. Scrollen Sie auf der Seite Review (Prüfen) nach unten und wählen Sie die Option I acknowledge that AWS CloudFormation might create IAM resources (Ich bin mir bewusst, dass CFN IAM-Ressourcen erstellen kann).
10. Wählen Sie Erstellen aus.

CloudFormation zeigt etwa drei Minuten den Status CREATE_IN_PROGRESS an. Der Status wird in CREATE_COMPLETE geändert, sobald der Stack erstellt wurde und die Rollen verwendet werden können.

> **Important**
>
> Wenn Sie einen automatisierten Workflow ausführen, der andere Services mithilfe einer AWS Identity and Access Management-(IAM)-Servicerolle aufruft, muss die Servicerolle mit der Berechtigung zum Aufrufen dieser Services konfiguriert sein. Diese

Anforderung gilt für alle AWS Automation-Runbooks (`AWS-*`-Runbooks), wie zum Beispiel `AWS-ConfigureS3BucketLogging`, `AWS-CreateDynamoDBBackup` und `AWS-RestartEC2Instance`-Runbooks, um nur einige zu nennen. Diese Anforderung gilt auch für alle von Ihnen erstellten benutzerdefinierten Automation-Runbooks, die andere AWS-Services mithilfe von Aktionen aufrufen, die andere Services aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:createStack`- oder `aws:copyImage`-Aktionen verwenden, konfigurieren Sie die Dienstrolle mit der Berechtigung zum Aufrufen solcher Services. Sie können anderen AWS-Services Berechtigungen erteilen, indem Sie der Rolle eine eingebundene IAM-Richtlinie hinzufügen. Weitere Informationen finden Sie unter (Optional) Fügen Sie eine Inline-Automatisierungsrichtlinie oder eine vom Kunden verwaltete Richtlinie hinzu, um andere aufzurufen AWS-Services.

Kopieren von Rolleninformationen für Automation

Kopieren Sie mit den folgenden Schritten Informationen über die Automation-Servicerolle aus der AWS CloudFormation-Konsole. Sie müssen diese Rollen beim Verwenden eines Runbooks festlegen.

> **Note**
>
> Sie müssen keine Rolleninformationen mit diesen Schritten kopieren, wenn Sie die Runbooks `AWS-UpdateLinuxAmi` oder `AWS-UpdateWindowsAmi` ausführen. In diesen Runbook sind die erforderlichen Rollen bereits als Standardwerte festgelegt. Die Rollen in diesen Runbooks verwenden von IAM verwaltete Richtlinien.

Kopieren der Rollennamen

1. Öffnen Sie die AWS CloudFormation-Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie den Stack name (Stack-Name) der Automation aus, den Sie im vorherigen Verfahren erstellt haben.
3. Wählen Sie die Registerkarte Resources (Ressourcen) aus.
4. Wählen Sie den Link Physical ID (Physische ID) für AutomationServiceRole. Beim Öffnen der IAM-Konsole wird eine Zusammenfassung der Servicerolle für die Automatisierung angezeigt.
5. Kopieren Sie den Amazon-Ressourcennamen (ARN) neben Role ARN (Rollen-ARN). Der ARN ist ähnlich wie der folgende: `arn:aws:iam::12345678:role/AutomationServiceRole`

6. Kopieren Sie den ARN zur späteren Verwendung in eine Textdatei.

Sie haben die Konfiguration der Automation-Servicerolle abgeschlossen. Sie können jetzt den ARN der Automation-Servicerolle in Ihren Runbooks verwenden.

Methode 2: Konfigurieren von Automation-Rollen mit IAM

Wenn Sie eine Servicerolle für Automation mit einer Fähigkeit von erstellen müssen AWS Systems Manager, führen Sie die folgenden Aufgaben aus. Weitere Informationen darüber, wann eine Servicerolle für Automation erforderlich ist, finden Sie unter Einrichten der Automatisierung.

Aufgaben

- Aufgabe 1: Erstellen einer Servicerolle für Automation
- Aufgabe 2: Hängen Sie die iam: PassRole -Richtlinie an Ihre Automation-Rolle an

Aufgabe 1: Erstellen einer Servicerolle für Automation

Führen Sie die folgenden Schritte zum Erstellen einer Service-Rolle (oder Übernahmerolle) für Systems Manager Automation.

> **Note**
>
> Sie können diese Rolle auch in Runbooks, wie dem `AWS-CreateManagedLinuxInstance`-Runbook, verwenden. Wenn Sie diese Rolle oder den Amazon-Ressourcennamen (ARN) einer AWS Identity and Access Management (IAM) -Rolle in Runbooks verwenden, kann Automation Aktionen in Ihrer Umgebung ausführen, z. B. neue Instances starten und Aktionen in Ihrem Namen ausführen.

Erstellen einer IAM-Rolle und Gestatten der Automatisierung

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
3. Wählen Sie unter Select type of trusted entity (Typ der vertrauenswürdigen Entität auswählen) die Option AWS -Service aus.
4. Wählen Sie im Abschnitt Choose a use case (Anwendungsfall auswählen) die Option Systems Manager und wählen Sie dann Next: Permissions (Weiter: Berechtigungen).

5. Suchen Sie auf der Seite „Angehängte Berechtigungsrichtlinie“ nach der AutomationRoleAmazonSSM-Richtlinie, wählen Sie sie aus und klicken Sie dann auf Weiter: Überprüfen.
6. Geben Sie auf der Seite Review im Feld Role name einen Namen und anschließend eine Beschreibung ein.
7. Wählen Sie Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Roles (Rollen) zurück.
8. Wählen Sie auf der Seite Roles (Rollen) die gerade erstellte Rolle aus, um die Seite Summary (Übersicht) zu öffnen. Notieren Sie sich den Role Name (Rollenname) und Role ARN (Rollen-ARN). Sie geben den Rollen-ARN an, wenn Sie im nächsten Verfahren die iam: PassRole - Richtlinie an Ihr IAM-Konto anhängen. Sie können den Rollennamen und den ARN in Runbooks festlegen.

> **Note**
>
> Die `AmazonSSMAutomationRole` Richtlinie weist die Automatisierungs-Rollenberechtigung einer Teilmenge von AWS Lambda Funktionen in Ihrem Konto zu. Diese Funktionen beginnen mit „Automation“ (Automatisierung). Wenn Sie die Automatisierung mit Lambda-Funktionen verwenden möchten, muss der Lambda-ARN das folgende Format verwenden: `"arn:aws:lambda:*:*:function:Automation*"`
> Wenn Sie über bestehende Lambda-Funktionen verfügen, deren ARNs dieses Format nicht verwenden, müssen Sie Ihrer Automatisierungsrolle auch eine zusätzliche Lambda-Richtlinie hinzufügen, z. B. die Richtlinie. AWSLambdaRole Die zusätzliche Richtlinie oder Rolle muss umfassendere Zugriffsberechtigungen für Lambda-Funktionen im AWS-Konto bieten.

Nachdem Sie Ihre Servicerolle erstellt haben, sollten Sie die Vertrauensrichtlinie bearbeiten, um das serviceübergreifende Confused-Deputy-Problem zu vermeiden. Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine Entität, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine Entität mit größeren Rechten zwingen kann, die Aktion auszuführen. In der AWS Tat kann ein dienstübergreifender Identitätswechsel zum Problem des verwirrten Stellvertreters führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der Anruf-Dienst kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, AWS bietet Tools, mit denen

Sie Ihre Daten für alle Dienste mit Dienstprinzipalen schützen können, denen Zugriff auf Ressourcen in Ihrem Konto gewährt wurde.

Wir empfehlen die Verwendung der globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen, die Automation einem anderen Service erteilt, auf eine bestimmte Ressource zu beschränken. Wenn der `aws:SourceArn`-Wert nicht die Konto-ID enthält, z. B. den ARN eines Amazon-S3-Buckets, müssen Sie beide globalen Bedingungskontext-Schlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden. Der Wert von `aws:SourceArn` muss für Automatisierungsausführungen der ARN sein. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Bedingungskontext-Schlüssel `aws:SourceArn` mit Platzhaltern (*) für die unbekannten Teile des ARN. z. B. `arn:aws:ssm:*:123456789012:automation-execution/*`.

Das folgende Beispiel zeigt, wie Sie die `aws:SourceArn` und `aws:SourceAccount` globale Bedingungskontext-Schlüssel für Automatisierung verwenden können, um das Confused-Deputy-Problem zu verhindern.

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

So ändern Sie die Vertrauensrichtinie einer Rolle

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie in der Rollenliste in Ihrem Konto den Namen der Automation-Servicerolle aus.
4. Klicken Sie auf der Registerkarte Trust Relationships (Vertrauensbeziehungen) auf Edit Trust Relationship (Vertrauensbeziehungen bearbeiten).
5. Bearbeiten Sie die Vertrauensrichtlinie mit den globalen Bedingungskontext-Schlüsseln `aws:SourceArn` und `aws:SourceAccount` für Automation, um das Confused-Deputy-Problem zu verhindern.
6. Wählen Sie Update Trust Policy (Vertrauensrichtlinie aktualisieren) aus, um die Änderungen zu speichern.

(Optional) Fügen Sie eine Inline-Automatisierungsrichtlinie oder eine vom Kunden verwaltete Richtlinie hinzu, um andere aufzurufen AWS-Services

Wenn Sie eine Automatisierung ausführen, die andere Dienste AWS-Services mithilfe einer IAM-Servicerolle aufruft, muss die Servicerolle so konfiguriert sein, dass sie berechtigt ist, diese Dienste aufzurufen. Diese Anforderung gilt für alle AWS Automatisierungs-Runbooks (`AWS-*`Runbooks) wie, und `AWS-RestartEC2Instance` Runbooks `AWS-ConfigureS3BucketLoggingAWS-CreateDynamoDBBackup`, um nur einige zu nennen. Diese Anforderung gilt auch für alle von Ihnen erstellten benutzerdefinierten Runbooks, die andere AWS-Services aufrufen, indem sie Aktionen verwenden, die andere Services aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:CreateStack`- oder `aws:copyImage`-Aktionen verwenden, dann müssen Sie die Servicerolle mit der Berechtigung zum Aufrufen solcher Services konfigurieren. Sie können anderen Benutzern Berechtigungen erteilen, AWS-Services indem Sie der Rolle eine IAM-Inline-Richtlinie oder eine vom Kunden verwaltete Richtlinie hinzufügen.

So betten Sie eine eingebundene Richtlinie für eine Servicerolle ein (IAM-Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie in der Liste den Namen der Rolle aus, die Sie bearbeiten möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie in der Dropdown-Liste Berechtigungen hinzufügen die Option Richtlinien anhängen oder Inline-Richtlinie erstellen.
6. Wenn Sie die Option Richtlinien anhängen wählen, aktivieren Sie das Kontrollkästchen neben der Richtlinie, die Sie hinzufügen möchten, und wählen Sie Berechtigungen hinzufügen.
7. Wenn Sie Inline-Richtlinie erstellen wählen, wählen Sie die Registerkarte JSON.
8. Geben Sie ein JSON-Richtliniendokument für das Dokument ein AWS-Services , das Sie aufrufen möchten. Nachfolgend sind zwei Beispiele für JSON-Richtliniendokumente aufgeführt.

   Amazon S3 PutObject und GetObject Beispiel

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

   Amazon EC2 CreateSnapshot und Beispiel DescribeSnapShots

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

```
            "Action":"ec2:CreateSnapshot",
            "Resource":"*"
        },
        {
            "Effect":"Allow",
            "Action":"ec2:DescribeSnapshots",
            "Resource":"*"
        }
    ]
}
```

Details zur IAM-Richtliniensprache und finden Sie in der IAM JSON Policy Reference im IAM-Benutzerhandbuch.

9. Wählen Sie, wenn Sie fertig sind, Review policy (Richtlinie überprüfen) aus. Die Richtlinienvalidierung meldet mögliche Syntaxfehler.
10. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) im Feld Name (Name) einen Namen für die zu erstellende Richtlinie ein. Überprüfen Sie unter Summary die Richtlinienzusammenfassung, um die Berechtigungen einzusehen, die von Ihrer Richtlinie gewährt werden. Wählen Sie dann Create policy aus, um Ihre Eingaben zu speichern.
11. Nachdem Sie eine Inline-Richtlinie erstellt haben, wird sie automatisch in Ihre Rolle eingebettet.

Aufgabe 2: Hängen Sie die iam: PassRole -Richtlinie an Ihre Automation-Rolle an

Fügen Sie mit den folgenden Schritten die Richtlinie `iam:PassRole` Ihrer Automation-Servicerolle hinzu. Dies erlaubt dem Automation-Service, die Rolle anderen Services oder Systems Manager-Funktionen zu übergeben, wenn Automatisierungen ausgeführt werden.

So hängen Sie die iam: PassRole -Richtlinie an Ihre Automatisierungsrolle an

1. Wählen Sie auf der Seite Summary für die gerade erstellte Rolle die Registerkarte Permissions.
2. Wählen Sie Inline-Richtlinie hinzufügen.
3. Wählen Sie auf der Seite Create policy die Registerkarte Visual editor aus.
4. Wählen Sie Service (Service) und anschließend die Option IAM aus.
5. Wählen Sie Select actions (Aktionen auswählen) aus.
6. Geben Sie in das Textfeld Aktionen filtern die PassRoleOption ein**`PassRole`**, und wählen Sie sie dann aus.

7. Wählen Sie Resources aus. Stellen Sie sicher, dass Specific ausgewählt ist und wählen Sie dann Add ARN aus.
8. Fügen Sie im Feld Specify ARN for role (ARN für die Rolle angeben) den ARN der Automation-Rolle ein, den Sie am Ende von Aufgabe 1 kopiert haben. Das System füllt die Felder Account (Konto) und Role name with path (Rollenname mit Pfad) automatisch aus.

   > **Note**
   >
   > Wenn Sie möchten, dass die Automation-Servicerolle eine IAM-Instance-Profilrolle an eine EC2-Instance anfügt, müssen Sie den ARN der IAM-Instance-Profilrolle hinzufügen. Auf diese Weise kann die Automation-Servicerolle die IAM-Instance-Profilrolle an die Ziel-EC2-Instance übergeben.

9. Wählen Sie Hinzufügen aus.
10. Wählen Sie Richtlinie prüfen.
11. Geben Sie auf der Seite Review Policy einen Namen ein und wählen Sie anschließend Create Policy aus.

Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst

Standardmäßig können Sie mit Automation bis zu 100 nebenläufige Automatisierungen gleichzeitig ausführen. Automation bietet zudem eine optionale Einstellung, mit der Sie Ihr Kontingent für nebenläufige Automatisierungen automatisch anpassen können. Mit dieser Einstellung kann Ihr Kontingent je nach verfügbaren Ressourcen bis zu 500 nebenläufige Automatisierungen umfassen.

> **Note**
>
> Wenn Ihre Automatisierung API-Vorgänge aufruft, kann eine adaptive Skalierung entsprechend Ihren Zielen zu Drosselungsausnahmen führen. Wenn beim Ausführen von Automatisierungen mit aktivierter adaptiver Nebenläufigkeit wiederholt Drosselungsausnahmen auftreten, müssen Sie möglicherweise Kontingenterhöhungen für den API-Vorgang anfordern, sofern verfügbar.

So aktivieren Sie die adaptive Nebenläufigkeit (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Aktivieren Sie das Kontrollkästchen neben Enable adaptive concurrency (Adaptive Nebenläufigkeit aktivieren).
5. Wählen Sie Save (Speichern).

Implementieren von Änderungskontrollen für Automatisierung

Standardmäßig ermöglicht Automatisierung die Verwendung von Runbooks ohne Datums- und Zeitbeschränkungen. Durch die Integration der Automatisierung mit Change Calendar können Sie Änderungskontrollen für alle Automatisierungen in Ihrem AWS-Konto implementieren. Mit dieser Einstellung können AWS Identity and Access Management (IAM)-Prinzipale in Ihrem Konto Automatisierungen nur während der von Ihrem Änderungskalender zugelassenen Zeiträume ausführen. Weitere Informationen zum Arbeiten mit Change Calendar finden Sie unter Arbeiten mit Change Calendar.

So aktivieren Sie Änderungskontrollen (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Aktivieren Sie das Kontrollkästchen neben Change Calendar-Integration aktivieren.
5. Wählen Sie in der Dropdown-Liste Änderungskalender auswählen den Änderungskalender aus, dem die Automatisierung folgen soll.
6. Wählen Sie Save (Speichern).

Ausführen von Automatisierungen

Dieser Abschnitt enthält Informationen dazu, wie Sie Runbooks ausführen. Automatisierung ist eine Funktion von AWS Systems Manager. Ausführlichere Tutorials zum Ausführen von Automatisierungen für Ihren Anwendungsfall finden Sie unter Tutorials.

Inhalt

- Führen Sie eine Automatisierung aus
- Eine Automatisierung mit Genehmigern ausführen
- Ausführen von Automatisierungen im großen Maßstab
- Ausführen von Automatisierungen in mehreren AWS-Regionen-Regionen und -Konten
- Ausführen von Automatisierungen basierend auf Ereignissen
- Führen Sie eine Automatisierung manuell aus

Führen Sie eine Automatisierung aus

Wenn Sie eine Automatisierung ausführen, wird die Automatisierung standardmäßig im Kontext des Benutzers ausgeführt, der die Automatisierung initiiert hat. Das bedeutet beispielsweise, wenn Ihr Benutzer über Administratorrechte verfügt, wird die Automatisierung mit Administratorrechten und vollständigem Zugriff auf die von der Automatisierung konfigurierten Ressourcen ausgeführt. Als bewährte Sicherheitsmaßnahme empfehlen wir, dass Sie Automatisierungen ausführen, indem Sie eine IAM-Servicerolle, die in diesem Fall auch als angenommene Rolle bekannt ist, verwenden, die mit der verwalteten Richtlinie AmazonSSMAutomationRole konfiguriert ist. Möglicherweise müssen Sie Ihrer angenommenen Rolle zusätzliche IAM-Richtlinien hinzufügen, um verschiedene Runbooks verwenden zu können. Die Verwendung einer IAM-Servicerolle zur Ausführung der Automatisierung wird als delegierte Administration bezeichnet.

Wenn Sie eine Servicerolle verwenden, darf die Automatisierung zwar für AWS-Ressourcen laufen, aber der Benutzer, der die Automatisierung ausgeführt hat, verfügt über einen eingeschränkten Zugriff (oder besitzt keinen Zugriff) auf diese Ressourcen. Beispielsweise können Sie eine Servicerolle konfigurieren und sie mit Automatisierung verwenden, um eine oder mehrere Amazon Elastic Compute Cloud (Amazon EC2)-Instances neu zu starten. Automation ist eine Funktion von AWS Systems Manager. Die Automatisierung startet die Instances neu, aber die Servicerolle gibt dem Benutzer nicht die Berechtigung, auf diese Instances zuzugreifen.

Sie können eine Servicerolle zur Laufzeit angeben, wenn Sie eine Automatisierung ausführen, oder Sie können benutzerdefinierte Runbooks erstellen und die Servicerolle direkt im Runbook angeben.

Wenn Sie zur Laufzeit oder in einem Runbook eine Servicerolle angeben, dann wird der Service im Kontext der angegebenen Servicerolle ausgeführt. Wenn Sie keine Servicerolle angeben, dann legt das System im Kontext des Benutzers eine temporäre Sitzung an und führt die Automatisierung aus.

> **Note**
>
> Für Automatisierungen, die voraussichtlich länger als 12 Stunden laufen, müssen Sie eine Servicerolle angeben. Wenn Sie eine lang laufende Automatisierung im Kontext eines Benutzers starten, läuft die temporäre Sitzung des Benutzers nach 12 Stunden ab.

Delegierte Administration sorgt für mehr Sicherheit und Kontrolle Ihrer AWS-Ressourcen. Sie erlaubt auch eine verbesserte Prüfungserfahrung, da Aktionen für Ihre Ressourcen von einer zentralen Servicerolle statt von mehreren IAM-Konten ausgeführt werden.

Bevor Sie beginnen

Bevor Sie die folgenden Verfahren ausführen, müssen Sie die IAM-Servicerolle erstellen und eine Vertrauensstellung für die Automatisierung, eine Funktion von AWS Systems Manager, erstellen. Weitere Informationen finden Sie unter Aufgabe 1: Erstellen einer Servicerolle für Automation.

In den folgenden Verfahren wird beschrieben, wie Sie die Systems-Manager-Konsole oder Ihr bevorzugtes Befehlszeilen-Tool zum Ausführen einer einfachen Automatisierung verwenden.

Ausführen einer einfachen Automatisierung (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine einfache Automatisierung ausführen.

Ausführen einer einfachen Automatisierung

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Automation (Automatisierung) und Execute automation (Automatisierung ausführen) aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Mit mir geteilt-Registerkarte. Um alle Runbooks anzuzeigen, wählen Sie die Alle Dokumente-Registerkarte.

> **Note**
>
> Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Document details (Dokument-Details), ob Document version (Dokumentversion) auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.

5. Wählen Sie Next (Weiter).
6. Wählen Sie im Abschnitt Execution Mode (Ausführungsmodus) die Option Simple execution (Einfache Ausführung) aus.
7. Geben Sie im Abschnitt Input Parameters (Eingabeparameter) die erforderlichen Eingaben an. Sie können optional eine IAM-Servicerolle aus der Liste AutomationAssumeRole auswählen.
8. (Optional) Wählen Sie einen CloudWatch-Alarm aus, der auf Ihre Automatisierung zur Überwachung angewendet werden soll. Um einen CloudWatch-Alarm an Ihre Automatisierung anzuhängen, muss der IAM-Prinzipal, der die Automatisierung startet, über die Berechtigung für die `iam:createServiceLinkedRole`-Aktion verfügen. Weitere Informationen zu CloudWatch-Alarmen erhalten Sie unter Verwendung von Amazon-CloudWatch-Alarmen. Beachten Sie, dass die Automatisierung gestoppt wird, wenn Ihr Alarm aktiviert wird. Wenn Sie AWS CloudTrail verwenden, sehen Sie den API-Aufruf in Ihrem Trail.
9. Wählen Sie Execute (Ausführen).

Die Konsole zeigt den Status der Automatisierung an. Wenn Automatisierung nicht ausgeführt werden kann, finden Sie weitere Informationen unter Fehlerbehebung für Systems Manager Automation..

Ausführen einer einfachen Automatisierung (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie Sie die AWS CLI (unter Linux oder Windows) oder AWS Tools for PowerShell verwenden, um eine einfache Automatisierung auszuführen.

Ausführen einer einfachen Automatisierung

1. Installieren und konfigurieren Sie die AWS CLI oder AWS Tools for PowerShell, falls noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

2. Führen Sie den folgenden Befehl aus, um eine einfache Automatisierung zu starten. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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
   ```

   Hier sehen Sie ein Beispiel, wie Sie das `AWS-RestartEC2Instance`-Runbook verwenden, um die angegebene EC2-Instance neu zu starten.

   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"}
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
    "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab"
}
```

Windows

```
{
    "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab"
}
```

PowerShell

```
4105a4fc-f944-11e6-9d32-0123456789ab
```

3. Führen Sie den folgenden Befehl aus, um den Status der Automatisierung abzurufen.

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"}
```

Das System gibt unter anderem folgende Informationen zurück

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                     : {}
```

Eine Automatisierung mit Genehmigern ausführen

In den folgenden Verfahren wird beschrieben, wie Sie mit der AWS Systems Manager-Konsole, AWS Command Line Interface (AWS CLI) und einer Automatisierung mit Genehmigungen mithilfe einer einfachen Ausführung ausführen. Die Automatisierung verwendet die Automatisierungsaktion `aws:approve`, die die Automatisierung vorübergehend unterbricht, bis die Aktion von den designierten Prinzipalen entweder genehmigt oder abgelehnt wird. Die Automatisierung wird im Kontext des aktuellen Benutzers ausgeführt. Das bedeutet, dass Sie keine zusätzlichen IAM-Berechtigungen konfigurieren müssen, solange Sie über die Berechtigung zum Ausführen des Runbooks verfügen und alle Aktionen von dem Runbook aufgerufen werden. Wenn Sie über Administrator-Berechtigungen in IAM verfügen, haben Sie bereits die Berechtigung zur Verwendung dieses Runbooks.

Bevor Sie beginnen

Zusätzlich zu den Standardeingaben, die für das Runbook erforderlich sind, erfordert die Aktion `aws:approve` die beiden folgenden Parameter:

- Eine Liste der Genehmiger. Die Liste der Genehmiger muss mindestens einen Genehmiger in Form eines Benutzernamens oder eines Benutzer-ARN enthalten. Wenn mehrere Genehmiger angegeben sind, muss im Runbook eine entsprechende minimale Genehmigungsanzahl festgelegt werden.
- Ein Amazon Simple Notification Service (Amazon SNS)-Thema ARN Der Name des Amazon SNS-Themas muss mit `Automation` beginnen.

Bei diesem Verfahren wird davon ausgegangen, dass Sie bereits ein Amazon SNS-Thema erstellt haben. Dies ist erforderlich, um den Genehmigungsprozess bereitzustellen. Weitere Informationen finden Sie unter Erstellen eines Themas im Amazon Simple Notification Service-Entwicklerhandbuch.

Ausführen einer Automatisierung mit Genehmigern (Konsole)

So führen Sie eine Automatisierung mit Genehmigern aus

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine Automatisierung mit Genehmigern ausführen.

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Automation (Automatisierung) und Execute automation (Automatisierung ausführen) aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Mit mir geteilt-Registerkarte. Um alle Runbooks anzuzeigen, wählen Sie die Alle Dokumente-Registerkarte.

   > **Note**
   >
   > Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Document details (Dokument-Details), ob Document version (Dokumentversion) auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.

5. Wählen Sie Next (Weiter).

6. Klicken Sie auf der Seite Execute automation document (Automation-Dokument ausführen) auf Simple execution (Einfache Ausführung).
7. Geben Sie im Abschnitt Input parameters (Eingabeparameter) die erforderlichen Eingabeparameter an.

   Wenn Sie beispielsweise das **`AWS-StartEC2InstanceWithApproval`**-Runbook ausgewählt haben, müssen Sie Instance-IDs für den Parameter InstanceId angeben oder auswählen.
8. Geben Sie im Abschnitt Genehmiger die Benutzernamen oder Benutzer-ARNs der Genehmiger für die Automatisierungsaktion an.
9. Geben Sie im Abschnitt SNSTopicARN den SNS-Themen-ARN an, der für das Senden von Genehmigungsbenachrichtigungen verwendet werden soll. Der SNS-Themenname muss mit Automation beginnen.
10. Sie können optional eine IAM-Servicerolle aus der Liste AutomationAssumeRole auswählen. Wenn Sie auf mehr als 100 Konten und Regionen abzielen, müssen Sie die `AWS-SystemsManager-AutomationAdministrationRole` angeben.
11. Wählen Sie Execute automation (Automatisierung ausführen).

Der angegebene Genehmiger erhält eine Amazon SNS-Benachrichtigung mit Details zum Genehmigen oder Ablehnen der Automatisierung. Diese Genehmigungsaktion gilt für sieben Tage ab dem Ausstellungsdatum und kann über die Systems Manager-Konsole oder AWS Command Line Interface (AWS CLI). erfolgen.

Wenn Sie die Automatisierung genehmigen, führt die Automatisierung die im angegebenen Runbook enthaltenen Schritte aus. Die Konsole zeigt den Status der Automatisierung an. Wenn Automatisierung nicht ausgeführt werden kann, finden Sie weitere Informationen unter Fehlerbehebung für Systems Manager Automation..

So genehmigen Sie eine Automatisierung oder lehnen sie ab

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation und wählen Sie dann die Automatisierung aus, die im vorherigen Verfahren ausgeführt wurde.
3. Wählen Sie Actions (Aktionen) und dann Approve/Deny (Genehmigen/ablehnen) aus.
4. Wählen Sie entweder Approve (Genehmigen) oder Deny (Ablehnen) aus und geben Sie bei Bedarf einen Kommentar ein.

5. Wählen Sie Submit (Absenden) aus.

Ausführen einer Automatisierung mit Genehmigern (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie Sie die AWS CLI (unter Linux oder Windows) oder AWS Tools for PowerShell verwenden, um eine Automatisierung mit Genehmigern auszuführen.

So führen Sie eine Automatisierung mit Genehmigern aus

1. Installieren und konfigurieren Sie die AWS CLI oder AWS Tools for PowerShell, falls noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

2. Verwenden Sie den folgenden Befehl, um eine Automatisierung mit Genehmigern auszuführen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Geben Sie im Abschnitt Dokumentname ein Runbook an, das die Automatisierungsaktion `aws:approve` enthält.

   Geben Sie für `Approvers` die Benutzernamen oder Benutzer-ARNs der Genehmiger für die Aktion an. Geben Sie für `SNSTopic` den SNS-Themen-ARN an, der zum Senden von Genehmigungsbenachrichtigungen verwendet werden soll. Der Name des Amazon SNS-Themas muss mit `Automation` beginnen.

   > **Note**
   >
   > Die spezifischen Namen der Parameterwerte für Genehmiger und das SNS-Thema hängen von den im ausgewählten Runbook angegebenen Werten ab.

   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"
    }
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
    "AutomationExecutionId": "df325c6d-b1b1-4aa0-8003-6cb7338213c6"
}
```

Windows

```
{
    "AutomationExecutionId": "df325c6d-b1b1-4aa0-8003-6cb7338213c6"
}
```

PowerShell

```
df325c6d-b1b1-4aa0-8003-6cb7338213c6
```

So genehmigen Sie eine Automatisierung

- Führen Sie den folgenden Befehl aus, um eine Automatisierung zu genehmigen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

  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"}
  ```

  Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

So lehnen Sie eine Automatisierung ab

- Führen Sie den folgenden Befehl aus, um eine Automatisierung abzulehnen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

  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"}
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Ausführen von Automatisierungen im großen Maßstab

Mit AWS Systems Manager Automation können Sie Automatisierungen für eine Flotte von AWS-Ressourcen mithilfe von Zielen ausführen. Außerdem können Sie die Bereitstellung der Automatisierung innerhalb Ihrer Flotte steuern, indem Sie einen Gleichzeitigkeitswert und einen Fehlergrenzwert angeben. Die Gleichzeitigkeits- und die Fehlergrenzwertfeature werden gemeinsam als Ratensteuerungen bezeichnet. Der Gleichzeitigkeitswert legt fest, wie viele Ressourcen die Automatisierung gleichzeitig ausführen kann. Automation bietet außerdem einen adaptiven Nebenläufigkeitsmodus, den Sie aktivieren können. Die adaptive Nebenläufigkeit skaliert Ihr Automatisierungskontingent automatisch von 100 gleichzeitig ausgeführten Automatisierungen auf bis zu 500. Ein Fehlergrenzwert legt fest, wie viele Automatisierungsausführungen fehlschlagen dürfen, bevor Systems Manager damit aufhört, die Automatisierung an andere Ressourcen zu senden.

Weitere Informationen über Gleichzeitigkeits- und Fehlergrenzwerte finden Sie unter Steuern von Automatisierungen im großen Maßstab. Weitere Informationen über Ziele finden Sie unter Zuordnen von Zielen für eine Automatisierung.

Die folgenden Verfahren veranschaulichen, wie Sie die adaptive Nebenläufigkeit aktivieren und eine Automatisierung mit Zielen und Ratensteuerelementen über die Systems-Manager-Konsole und AWS Command Line Interface (AWS CLI) ausführen.

Ausführen einer Automatisierung mit Zielen und Ratensteuerungen (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mit der Systems Manager-Konsole eine Automatisierung mit Ziel- und Ratensteuerungen ausführen.

So führen Sie eine Automatisierung mit Zielen und Ratensteuerungen aus

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Automation (Automatisierung) und Execute automation (Automatisierung ausführen) aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Mit mir geteilt-Registerkarte. Um alle Runbooks anzuzeigen, wählen Sie die Alle Dokumente-Registerkarte.

   > **Note**
   >
   > Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Document details (Dokument-Details), ob Document version (Dokumentversion) auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.

5. Wählen Sie Next (Weiter).
6. Wählen Sie im Abschnitt Execution Mode (Ausführungsmodus) die Option Rate Control (Ratensteuerung) aus. Sie müssen diesen Modus oder die Option Multi-account and Region

(Mehrere Konten und Regionen) verwenden, wenn Sie Ziele und Ratensteuerungen nutzen möchten.

7. Wählen Sie im Abschnitt Targets (Ziele) die Ausrichtung auf die AWS-Ressourcen für die Ausführung der Automation. Diese Optionen sind erforderlich.

   a. Wählen Sie in der Liste Parameter (Parameter) einen Parameter aus. Die Elemente in der Liste Parameter richten sich nach den Parametern in dem Automation-Runbook, das Sie zu Beginn dieses Verfahrens ausgewählt haben. Durch Auswahl eines Parameters legen Sie den Typ der Ressource fest, für die der Automation-Workflow ausgeführt wird.

   b. Wählen Sie in der Liste Targets (Ziele)aus, wie Sie Ressourcen als Ziele verwenden möchten.

      i. Wenn Sie die Zielressourcen mithilfe von Parameterwerten ausgewählt haben, geben Sie den Parameterwert für den gewählten Parameter im Feld Eingabeparameter eingeben ein.

      ii. Wenn Sie die Zielressourcen mit AWS Resource Groups ausgewählt haben, wählen Sie den Namen der Gruppe aus der Liste Resource Group (Ressourcengruppe) aus.

      iii. Wenn Sie die Zielressourcen mithilfe von Tags ausgewählt haben, geben Sie den Tag-Schlüssel und (optional) den Tag-Wert in die entsprechenden Felder ein. Wählen Sie Add (Hinzufügen) aus.

      iv. Wenn Sie ein Automatisierungs-Runbook für alle Instances im aktuellen AWS-Konto und AWS-Region aus. Wählen Sie und anschließend Alle Instances aus.

8. Geben Sie im Abschnitt Input Parameters (Eingabeparameter) die erforderlichen Eingaben an. Sie können optional eine IAM-Servicerolle aus der Liste AutomationAssumeRole auswählen.

> **Note**
>
> Möglicherweise müssen Sie einige der Optionen im Abschnitt Input parameters (Eingabeparameter) nicht auswählen. Dies liegt daran, dass Sie Ressourcen mithilfe von Tags oder einer Ressourcengruppe als Ziele ausgewählt haben. Wenn Sie beispielsweise das `AWS-RestartEC2Instance`-Runbook ausgewählt haben, müssen Sie keine Instance-IDs im Abschnitt Input parameters (Eingabeparameter) angeben oder auswählen. Die Automation-Ausführung sucht die Instances für den Neustart mit den von Ihnen angegebenen Tags oder Ressourcengruppen.

9. Verwenden Sie die Optionen im Abschnitt Rate control (Ratensteuerung), um die Anzahl der AWS-Ressourcen zu beschränken, welche die Automatisierung mit jedem Konto-Region-Paar ausführen können.

   Wählen Sie im Abschnitt Concurrency (Gleichzeitigkeit) eine Option aus:

   - Wählen Sie targets (Ziele) aus, um eine absolute Anzahl von Zielen einzugeben, die den Automation-Workflow gleichzeitig ausführen können.
   - Wählen Sie percentage (Prozentsatz) aus, um einen Prozentsatz der Ziele anzugeben, die den Automation-Workflow gleichzeitig ausführen können.
10. Wählen Sie im Abschnitt Error threshold (Fehlerschwellenwert) eine Option aus:

    - Wählen Sie errors (Fehler), um eine absolute Anzahl von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, den Workflow an andere Ressourcen zu senden.
    - Wählen Sie percentage (Prozentsatz) aus, um einen Prozentsatz von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, den Workflow an andere Ressourcen zu senden.
11. (Optional) Wählen Sie einen CloudWatch-Alarm aus, der auf Ihre Automatisierung zur Überwachung angewendet werden soll. Um einen CloudWatch-Alarm an Ihre Automatisierung anzuhängen, muss der IAM-Prinzipal, der die Automatisierung startet, über die Berechtigung für die `iam:createServiceLinkedRole`-Aktion verfügen. Weitere Informationen zu CloudWatch-Alarmen erhalten Sie unter Verwendung von Amazon-CloudWatch-Alarmen. Beachten Sie, dass die Automatisierung gestoppt wird, wenn Ihr Alarm aktiviert wird. Wenn Sie AWS CloudTrail verwenden, sehen Sie den API-Aufruf in Ihrem Trail.
12. Wählen Sie Execute (Ausführen).

Um Automatisierungen anzuzeigen, die von der Automatisierung der Ratensteuerung gestartet wurden, wählen Sie im Navigationsbereich Automation (Automatisierung) und wählen Sie dann Anzeigen von untergeordneten Automatisierungen.

Ausführen einer Automatisierung mit Zielen und Ratensteuerungen (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie Sie die AWS CLI (unter Linux oder Windows) oder AWS Tools for PowerShell verwenden, um eine Automatisierung mit Ziel- und Ratensteuerungen auszuführen.

So führen Sie eine Automatisierung mit Zielen und Ratensteuerungen aus

1. Installieren und konfigurieren Sie die AWS CLI oder AWS Tools for PowerShell, falls noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

2. Nutzen Sie den folgenden Befehl, um eine Liste der Dokumente anzuzeigen.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

   Windows

   ```
   aws ssm list-documents
   ```

   PowerShell

   ```
   Get-SSMDocumentList
   ```

   Beachten Sie den Namen des Runbooks, das Sie verwenden möchten.

3. Führen Sie den folgenden Befehl aus, um Details des Runbooks einsehen zu können: Ersetzen Sie *runbook name* mit dem Namen des Runbooks, dessen Details Sie anzeigen möchten. Notieren Sie auch einen Parameternamen (z. B. `InstanceId`), den Sie für die Option `--target-parameter-name` verwenden möchten. Dieser Parameter bestimmt den Typ der Ressource, für die die Automatisierung ausgeführt wird.

   Linux & macOS

   ```
   aws ssm describe-document \
       --name runbook name
   ```

   Windows

   ```
   aws ssm describe-document ^
       --name runbook name
   ```

PowerShell

```
Get-SSMDocumentDescription `
    -Name runbook name
```

4. Erstellen Sie einen Befehl, der die Ziel- und Ratensteuerungsoptionen verwendet, die Sie ausführen möchten. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Ausrichtung mithilfe von 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%"
```

Ausrichtung mithilfe von Parameterwerten

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%"
```

Ausrichtung mithilfe von 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%"
```

Ausrichtung auf alle Amazon-EC2-Instances im aktuellen AWS-Konto und der AWS-Region

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%"
```

Der Befehl gibt eine Ausführungs-ID zurück. Kopieren Sie diese ID in die Zwischenablage. Sie können diese ID zum Anzeigen des Status der Automatisierung verwenden.

Linux & macOS

```
{
```

```
    "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE"
}
```

Windows

```
{
    "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE"
}
```

PowerShell

```
a4a3c0e9-7efd-462a-8594-01234EXAMPLE
```

5. Führen Sie den folgenden Befehl aus, um die Automatisierung anzuzeigen. Ersetzen Sie jede *Automatisierungs-Ausführungs-ID* mit Ihren eigenen Informationen.

   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. Führen Sie den folgenden Befehl aus, um Details über den Automatisierungsprozess anzuzeigen. Ersetzen Sie jede *Automatisierungs-Ausführungs-ID* mit Ihren eigenen Informationen.

   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
```

Das System gibt unter anderem folgende Informationen zurück

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**
>
> Sie können auch den Status der Automatisierung in der Konsole überwachen. Wählen Sie in der Liste Automatisierungs-Ausführung die Automatisierung, die Sie gerade ausgeführt haben, und wählen Sie dann die Registerkarte Execution steps (Ausführungsschritte). Diese Registerkarte zeigt Ihnen den Status der Automatisierungs-Aktionen.

Zuordnen von Zielen für eine Automatisierung

Verwenden Sie den `Targets`-Parameter, um schnell zu definieren, auf welche Ressourcen eine Automatisierung abzielt. Wenn Sie beispielsweise eine Automatisierung ausführen möchten, die Ihre verwalteten Instances neu startet, können Sie, anstatt manuell Dutzende von Instance-IDs in der Konsole oder in einem Befehl einzugeben, Ziel-Instances festlegen, indem Sie Amazon Elastic Compute Cloud (Amazon EC2)-Tags mit dem `Targets`-Parameter verwenden.

Wenn Sie eine Automatisierung ausführen, die ein Ziel verwendet, AWS Systems Manager wird für jedes Ziel eine untergeordnete Automatisierung erstellt. Wenn Sie z. B. mithilfe von Tags Amazon Elastic Block Store (Amazon EBS)-Volume angeben und diese Tags auf 100 Amazon EBS-Volumes aufgelöst werden, dann erstellt Systems Manager 100 untergeordnete Automatisierungen. Die übergeordnete Automatisierung ist abgeschlossen, wenn alle untergeordneten Automatisierungen einen endgültigen Status erreicht haben.

> **Note**
>
> Alle `input parameters`, die Sie zur Laufzeit angeben (entweder im Abschnitt Input parameters (Eingabeparameter) der Konsole oder mithilfe der Option `parameters` auf der Befehlszeile) werden automatisch von allen untergeordneten Automatisierungen verarbeitet.

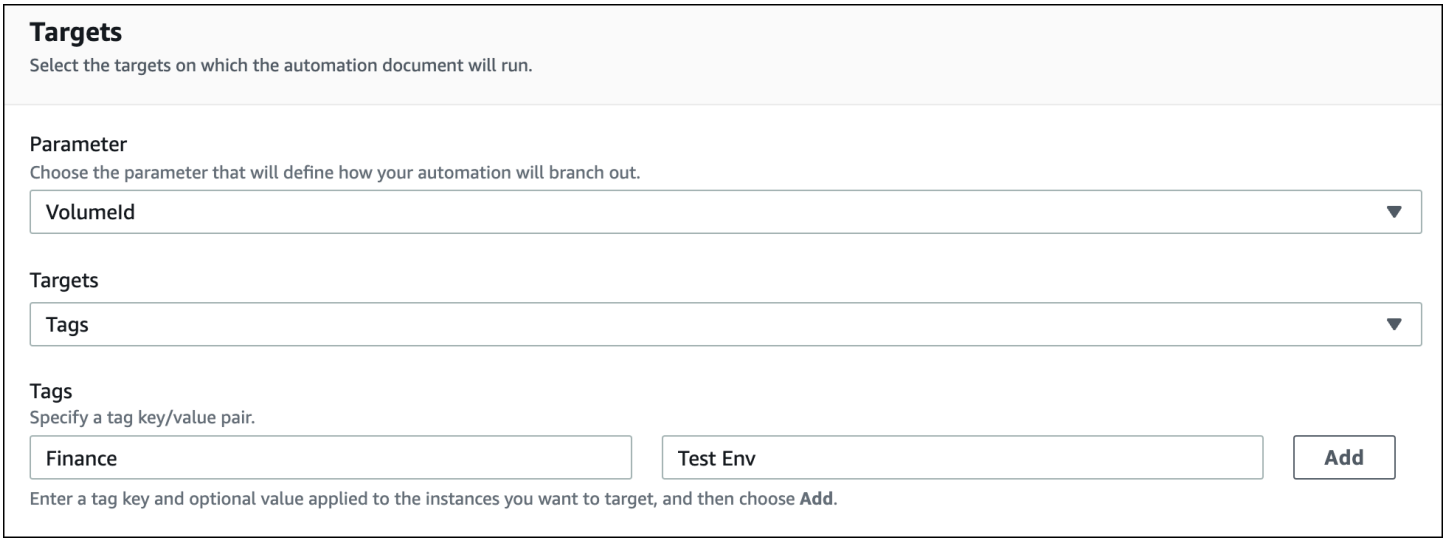
Sie können Ressourcen für eine Automatisierung gezielt einsetzen, indem Sie Tags, Resource Groups und Parameterwerte verwenden. Darüber hinaus können Sie mit der Option `TargetMaps` mehrere Parameterwerte über die Befehlszeile oder eine Datei als Ziel einrichten. Der folgende Abschnitt beschreibt die einzelnen Targeting-Optionen eingehender.

Anzielen eines Tags

Sie können einen einzelnen Tag als Ziel einer Automatisierung bestimmen. Viele AWS -Ressourcen unterstützen Tags, einschließlich Amazon Elastic Compute Cloud (Amazon EC2) und Amazon Relational Database Service (Amazon RDS) -Instances, Amazon Elastic Block Store (Amazon EBS) -Volumes und -Snapshots, Resource Groups und Amazon Simple Storage Service (Amazon S3) -Buckets. Sie können Automatisierungen schnell auf Ihren AWS -Ressourcen ausführen, indem Sie einen Tag anzielen. Ein Tag ist ein Schlüssel-Wert-Paar, z. B. Operating_System:Linux oder Department:Finance. Wenn Sie einer Ressource einen bestimmten Namen zuweisen, können Sie auch das Wort „Name“ als Schlüssel und den Namen der Ressource als Wert verwenden.

Wenn Sie einen Tag als Ziel für eine Automatisierung angeben, geben Sie auch einen Ziel-Parameter an. Der Ziel-Parameter verwendet die Option `TargetParameterName`. Durch Auswahl eines Zielparameters legen Sie den Typ der Ressource fest, für die die Automatisierung ausgeführt wird. Der Zielparameter, den Sie mit dem Tag angeben, muss ein im Runbook definierter gültiger Parameter sein. Wenn Sie beispielsweise Tags für Dutzende von EC2-Instances verwenden möchten, wählen Sie den Zielparameter `InstanceId`. Durch die Auswahl dieses Parameters legen Sie Instances als Ressourcentyp für die Automatisierung fest. Beim Erstellen eines benutzerdefinierten Runbooks müssen Sie den Zieltyp als `/AWS::EC2::Instance` angeben, um sicherzustellen, dass nur Instances verwendet werden. Andernfalls werden alle Ressourcen mit demselben Tag als Ziel ausgewählt. Wenn Sie auf Instances mit einem Tag abzielen, werden möglicherweise beendete Instances eingeschlossen.

Im folgenden Screenshot werden die `AWS-DetachEBSVolume`-Runbook verwendet. Der logische Ziel-Parameter ist `VolumeId`.



Das `AWS-DetachEBSVolume`-Runbook enthält auch eine spezielle Eigenschaft namens Zieltyp, welche auf `/AWS::EC2::Volume` gesetzt wird. Dies bedeutet: Wenn das Tag-Schlüssel-Paar `Finance:TestEnv` unterschiedliche Ressourcentypen zurückgibt (zum Beispiel EC2-Instances, Amazon EBS-Volumes, Amazon EBS-Snapshots), werden nur Amazon EBS-Volumes verwendet.

> **Important**
>
> Bei Zielparameternamen muss die Groß- und Kleinschreibung beachtet werden. Wenn Sie Automatisierungen entweder mit AWS Command Line Interface (AWS CLI) oder ausführen AWS Tools for Windows PowerShell, müssen Sie den Namen des Zielparameters genau so eingeben, wie er im Runbook definiert ist. Andernfalls gibt das System einen `InvalidAutomationExecutionParametersException`-Fehler aus. Sie können den DescribeDocumentAPI-Vorgang verwenden, um Informationen zu den verfügbaren Zielparametern in einem bestimmten Runbook abzurufen. Im Folgenden finden Sie einen AWS CLI Beispielbefehl, der Informationen über das `AWS-DeleteSnapshot` Dokument bereitstellt.
>
> ```
> aws ssm describe-document \
>     --name AWS-DeleteSnapshot
> ```

Im Folgenden finden Sie einige AWS CLI Beispielbefehle, die mithilfe eines Tags auf Ressourcen abzielen.

Beispiel 1: Zielgerichtete Tags mit einem Schlüssel-Wert-Paar zum Neustarten von Amazon-EC2-Instances

In diesem Beispiel werden alle Amazon EC2 EC2-Instances neu gestartet, die mit dem Schlüssel Department und dem Wert gekennzeichnet sind. HumanResources Der Zielparameter verwendet den InstanceIdParameter aus dem Runbook. Im Beispiel wird ein zusätzlicher Parameter für die Ausführung der Automation mithilfe einer Automation-Servicerolle (auch als Übernahmerolle bezeichnet) verwendet.

```
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"
```

Beispiel 2: Zielgerichtete Tags mit einem Schlüssel-Wert-Paar zum Löschen von Amazon-EBS-Snapshots

Das folgende Beispiel verwendet das `AWS-DeleteSnapshot`-Runbook zum Löschen aller Snapshots mit dem Schlüssel Name und dem Wert January2018Backups. Der Zielparameter verwendet den VolumeIdParameter.

```
aws ssm start-automation-execution \
    --document-name AWS-DeleteSnapshot \
    --targets Key=tag:Name,Values=January2018Backups \
    --target-parameter-name VolumeId
```

Targeting AWS Resource Groups

Sie können eine einzelne AWS Ressourcengruppe als Ziel einer Automatisierung angeben. Systems Manager erstellt eine untergeordnete Automatisierung für jedes Objekt in der Ziel-Ressourcengruppe.

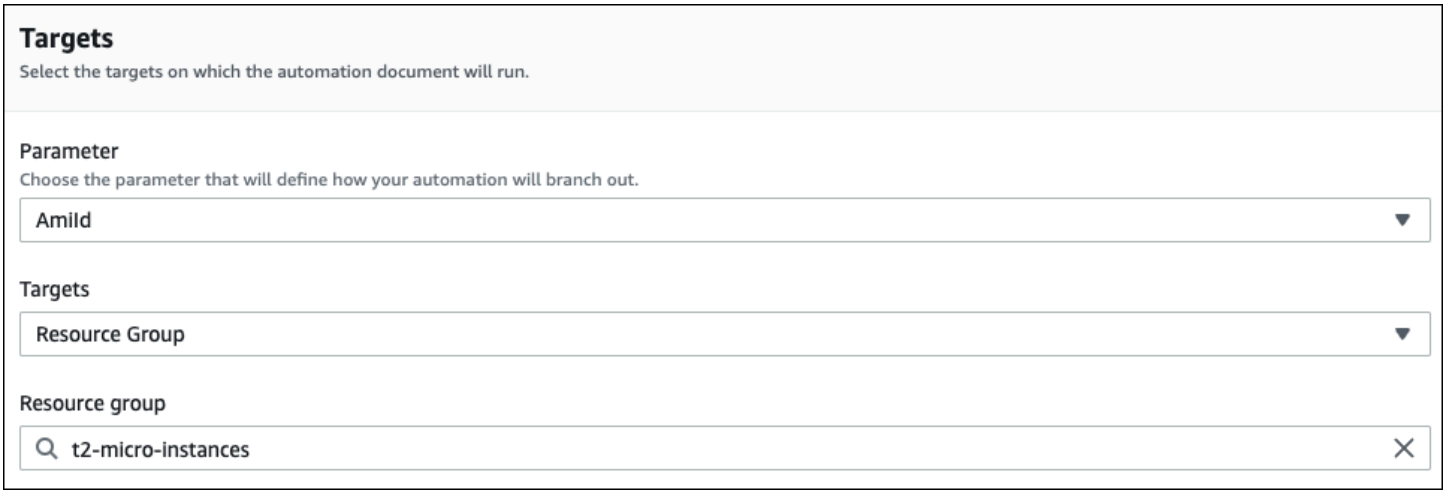
Beispiel: Angenommen, eine Ihrer Ressourcengruppen ist PatchedAMIs. Diese Ressourcengruppe enthält eine Liste von 25 Windows Amazon Machine Images (AMIs), die routinemäßig gepatcht werden. Wenn Sie eine Automatisierung ausführen, die das `AWS-CreateManagedWindowsInstance`-Runbook verwendet, und Sie diese auf diese Resource Group ausrichten, erstellt Systems Manager eine untergeordnete Automatisierung für jede der 25 AMIs. Dies bedeutet, dass die Automatisierung aufgrund der Ausrichtung auf die Resource Group PatchedAMIs 25 Instances aus einer Liste von gepatchten AMIserstellt. Die übergeordnete Automatisierung ist

abgeschlossen, wenn alle untergeordneten Automatisierungen abgeschlossen sind oder einen endgültigen Status erreicht haben.

Der folgende AWS CLI Befehl bezieht sich auf das Beispiel PatchAMIs Resource Group. Der Befehl verwendet den AmiIdParameter für die `--target-parameter-name` Option. Der Befehl enthält keinen zusätzlichen Parameter, der festlegt, welche Art von Instance aus jeder AMI erstellt werden soll. Das `AWS-CreateManagedWindowsInstance`-Runbook verwendet standardmäßig den Instance-Typ t2.medium, so dass dieser Befehl 25 t2.medium Amazon-EC2-Instances für Windows Server erstellt.

```
aws ssm start-automation-execution \
    --document-name AWS-CreateManagedWindowsInstance \
    --targets Key=ResourceGroup,Values=PatchedAMIs  \
    --target-parameter-name AmiId
```

Das folgende Konsolenbeispiel verwendet eine Ressourcengruppe mit dem Namen t2-micro-instances.



Ausrichtung auf Parameterwerte

Sie können auch einen Parameterwert zur Ausrichtung verwenden. Geben Sie `ParameterValues` als Schlüssel und dann den spezifischen Ressourcenwert für die Ausführung der Automatisierung ein. Wenn Sie mehrere Werte angeben, führt Systems Manager eine untergeordnete Automatisierung für jeden angegebenen Wert aus.

Nehmen Sie beispielsweise an, dass das Runbook einen InstanceID-Parameter enthält. Wenn Sie die Werte des InstanceID-Parameters beim Ausführen von Automation verwenden, führt Systems Manager eine untergeordnete Automatisierung für jeden angegebenen Instance-ID-Wert aus. Die

übergeordnete Automatisierung ist abgeschlossen, wenn Automatisierung die Ausführung jeder angegebenen Instance abgeschlossen hat oder wenn die Automatisierung fehlschlägt. Sie können maximal 50 Parameterwerte für die Ausrichtung verwenden.

Im folgenden Beispiel wird das `AWS-CreateImage`-Runbook verwendet. Der angegebene Zielparametername lautet InstanceId. Der Schlüssel verwendet ParameterValues. Die Werte sind zwei Amazon-EC2-Instance-IDs. Dieser Befehl erstellt einen Automatisierung für jede Instance, wodurch eine AMI von jeder Instance erzeugt wird.

```
aws ssm start-automation-execution
    --document-name AWS-CreateImage \
    --target-parameter-name InstanceId \
    --targets Key=ParameterValues,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE
```

> **Note**
>
> `AutomationAssumeRole` ist kein gültiger Parameter. Wählen Sie dieses Element nicht aus, wenn Sie die Automatisierung ausführen, die auf einen Parameterwert abzielt.

Ausrichtung auf Parameterwert-Maps

Die Option `TargetMaps` erweitert die Möglichkeiten zur Ausrichtung auf `ParameterValues`. Sie können ein Array von Parameterwerten mithilfe von `TargetMaps` auf der Befehlszeile eingeben. Sie können maximal 50 Parameterwerte in der Befehlszeile angeben. Wenn Sie Befehle ausführen möchten, die mehr als 50 Parameterwerte angeben, können Sie die Werte in einer JSON-Datei eingeben. Sie können dann die Datei von der Befehlszeile aus aufrufen.

> **Note**
>
> Die `TargetMaps`-Option wird in der Konsole nicht unterstützt.

Verwenden Sie das folgende Format, um mehrere Parameterwerte angeben, indem Sie die Option `TargetMaps` in einem Befehl verwenden. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

```
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”
```

Wenn Sie mehr als 50 Parameterwerte für die Option `TargetMaps` angeben möchten, geben Sie die Werte mit dem folgenden JSON-Format an. Die Verwendung einer JSON-Datei verbessert auch die Lesbarkeit bei mehreren Parameterwerten.

```
[

    {“parameter”: "value", “parameter 2”: "value", “parameter 3”: "value"},

    {“parameter 4”: "value", “parameter 5”: "value", "parameter 6": "value"}

]
```

Speichen Sie die Datei mit der Dateierweiterung .json. Sie können die Datei mit dem folgenden Befehl ausführen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --parameters input parameters \
    --target-maps path to file/file name.json
```

Sie können die auch aus einem Amazon Simple Storage Service (Amazon S3)-Bucket herunterladen, sofern Sie über die Berechtigung zum lesen von Daten aus dem Bucket verfügen. Verwenden Sie das folgende Befehlsformat. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --target-maps http://DOC-EXAMPLE-BUCKET.s3.amazonaws.com/file_name.json
```

Hier sehen Sie ein Beispiel für ein Szenario, das Ihnen dabei hilft, die Option `TargetMaps` zu verstehen. In diesem Szenario möchte ein Benutzer Amazon EC2-Instances verschiedener Typen aus verschiedenen AMIs erstellen. Für diese Aufgabe erstellt der Benutzer erstellt ein Runbook mit dem Namen AMI_Testing. Dieses Runbook definiert zwei Eingabeparameter: `instanceType` und `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"
  ]
}
```

Dann gibt der Benutzer die folgenden Ziel-Parameterwerte in einer Datei mit dem Namen `AMI_instance_types.json` an.

```
[
  {
```

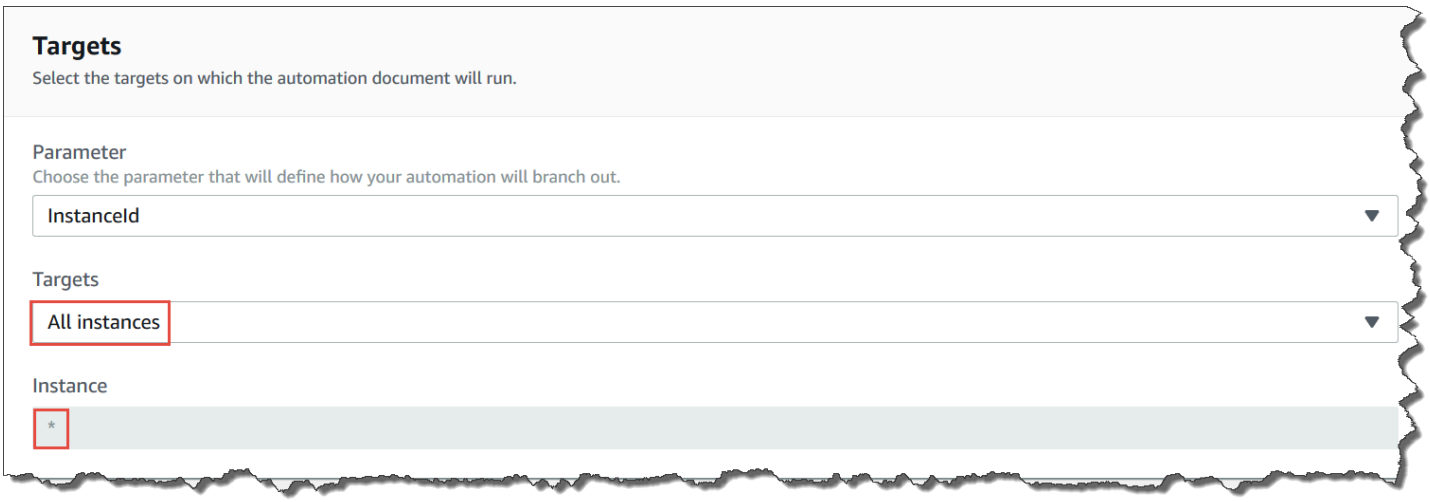
```
    "instanceType" : ["t2.micro"],
    "imageId" : ["ami-b70554c8"]
  },
  {
    "instanceType" : ["t2.small"],
    "imageId" : ["ami-b70554c8"]
  },
  {
    "instanceType" : ["t2.medium"],
    "imageId" : ["ami-cfe4b2b0"]
  },
  {
    "instanceType" : ["t2.medium"],
    "imageId" : ["ami-cfe4b2b0"]
  },
  {
    "instanceType" : ["t2.medium"],
    "imageId" : ["ami-cfe4b2b0"]
  }
]
```

Mit dem folgenden Befehl kann der Benutzer die Automatisierung ausführen und die fünf EC2-Instances erstellen, die in `AMI_instance_types.json` definiert sind.

```
aws ssm start-automation-execution \
    --document-name AMI_Testing \
    --target-parameter-name imageId \
    --target-maps file:///home/TestUser/workspace/runinstances/AMI_instance_types.json
```

Ausrichtung auf alle Amazon-EC2-Instances

Sie können eine Automatisierung auf allen Amazon EC2 EC2-Instances in der aktuellen AWS-Konto Version ausführen, AWS-Region indem Sie in der Zielliste Alle Instances auswählen. Wenn Sie beispielsweise alle Amazon EC2 EC2-Instances, Ihre AWS-Konto und die aktuelle AWS-Region, neu starten möchten, können Sie das **`AWS-RestartEC2Instance`** Runbook und dann Alle Instances aus der Liste Ziele auswählen.



Nachdem Sie Alle Instances gewählt haben, versieht Systems Manager das Instance-Feld einem Sternchen (*) und macht das Feld für Änderungen nicht verfügbar (das Feld ist ausgegraut). Systems Manager macht außerdem das InstanceIdFeld im Feld Eingabeparameter für Änderungen nicht verfügbar. Diese Felder für Änderungen nicht verfügbar zu machen, ist ein erwartetes Verhalten, wenn Sie sich dafür entscheiden, alle Instances abzudecken.

Steuern von Automatisierungen im großen Maßstab

Sie können die Bereitstellung einer Automatisierung innerhalb einer Flotte von AWS-Ressourcen steuern, indem Sie einen Gleichzeitigkeitswert und einen Fehlergrenzwert angeben. Die Gleichzeitigkeits- und die Fehlergrenzwertfunktion werden gemeinsam als Ratensteuerungen bezeichnet.

Nebenläufigkeit

Mit dem Gleichzeitigkeitswert können Sie angeben, wie viele Ressourcen eine Automatisierung gleichzeitig ausführen können. Die Gleichzeitigkeitsfunktion hilft dabei, die Auswirkungen auf Ihre Ressourcen oder Ausfälle werden der Ausführung einer Automatisierung zu begrenzen. Sie können entweder eine absolute Anzahl an Ressourcen, z. B. 20, oder einen Prozentsatz des festgelegten Ziels, beispielsweise 10 %, festlegen.

Das Warteschlangensystem übermittelt die Automatisierung an eine einzelne Ressource und wartet, bis der erste Aufruf abgeschlossen ist, bevor die Automatisierung an zwei weitere Ressourcen geschickt wird. Das System sendet die Automatisierung exponentiell an mehrere Ressourcen, bis der Gleichzeitigkeitswert erreicht ist.

Fehlerschwellenwerte

Verwenden Sie einen Fehlerschwellenwert, um festzulegen, wie viele Automatisierungen fehlschlagen dürfen, bevor AWS Systems Manager damit aufhört, die Automatisierung an andere Ressourcen zu senden. Sie können entweder eine absolute Anzahl an Fehlern, z. B. 10, oder einen Prozentsatz des festgelegten Ziels, beispielsweise 10 % festlegen.

Wenn Sie z. B. die absolute Zahl von 3 Fehlern angeben, führt das System keine Automatisierung mehr aus, wenn der vierte Fehler empfangen wird. Wenn Sie 0 angeben, führt das System keine weitere Automatisierung auf zusätzlichen Zielen aus, nachdem das erste Fehlerergebnis zurückgegeben wird.

Wenn Sie eine Automatisierung etwa an 50 Instances senden und den Fehlerschwellenwert auf 10 % festlegen, sendet das System keinen Befehl mehr an weitere Instances, wenn der fünfte Fehler empfangen wird. Aufrufe, die bereits eine Automatisierung ausführen, wenn ein Fehlerschwellenwert erreicht wird, können abgeschlossen werden, einige dieser Automatisierungen können jedoch dennoch fehlschlagen. Wenn Sie sicherstellen müssen, dass nicht mehr Fehlern als der angegebene Wert für den Fehlergrenzwert auftreten, setzen Sie den Wert für die Concurrency (Gleichzeitigkeit) auf 1, sodass die Automatisierungen jeweils einzeln ausgeführt werden.

Ausführen von Automatisierungen in mehreren AWS-Regionen-Regionen und -Konten

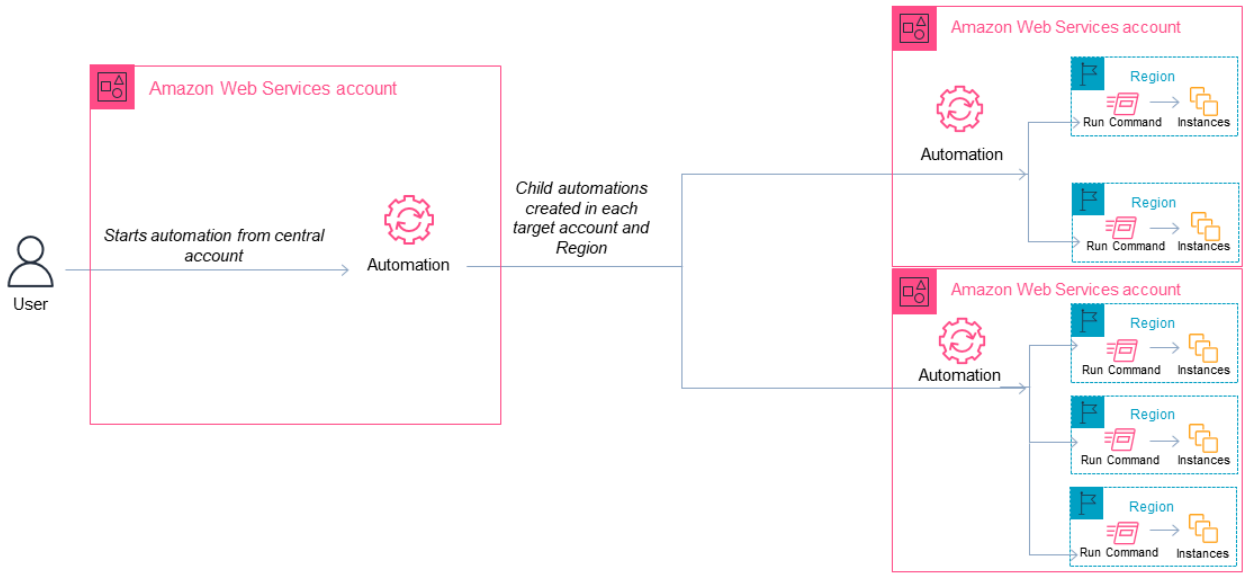
Sie können AWS Systems Manager-Automatisierungen über mehrere AWS-Regionen und AWS-Konten oder AWS Organizations-Organisationseinheiten (OUs) von einem zentralen Konto aus ausführen. Automation ist eine Funktion von AWS Systems Manager. Die Ausführung von Automatisierungen in mehreren Regionen und Konten oder OUs verkürzt die Zeit für die Bereitstellung Ihrer AWS-Ressourcen und verbessert die Sicherheit Ihrer Computingumgebung.

Sie können z. B. Folgendes tun, indem Sie Automatisierungs-Runbooks verwenden:

- Implementieren Sie Patches und Sicherheitsupdates zentral.
- Korrigieren Sie Compliance-Abweichungen bei VPC-Konfigurationen oder Amazon-S3-Bucket-Richtlinien.
- Verwalten Sie Ressourcen wie Amazon Elastic Compute Cloud (Amazon EC2)-Instances im großen Maßstab.

Das folgende Diagramm zeigt ein Beispiel für einen Benutzer, der das `AWS-RestartEC2Instances`-Runbook in mehreren Regionen und Konten von einem zentralen Konto aus ausführt. Die Automatisierung sucht die Instances unter Verwendung der angegebenen Tags in den Zielregionen und -konten.

> **Note**
>
> Wenn Sie eine Automatisierung mehrere Regionen und Konten hinweg ausführen, verwenden Sie Tags oder den Namen einer AWS-Ressourcengruppe für die Ressourcenausrichtung. Die Ressourcengruppe muss in jedem Zielkonto und jeder Region vorhanden sein. Der Name der Ressourcengruppe muss in jedem Zielkonto und jeder Region identisch sein. Die Automatisierung schlägt für Ressourcen fehl, die nicht über das angegebene Tag verfügen oder die nicht in der angegebenen Ressourcengruppe enthalten sind.



Auswählen eines zentralen Kontos für Automation

Wenn Sie Automatisierungen über Organisationseinheiten hinweg ausführen möchten, muss das zentrale Konto über Berechtigungen zum Auflisten aller Konten in den OUs verfügen. Dies ist nur über ein delegiertes Administratorkonto oder das Verwaltungskonto der Organisation möglich. Es wird empfohlen, die bewährten Methoden von AWS Organizations zu befolgen und ein delegiertes Administratorkonto zu verwenden. Weitere Informationen zu den bewährten Methoden von AWS Organizations finden Sie unter Bewährte Methoden für das Verwaltungskonto im AWS Organizations

Benutzerhandbuch. Um ein delegiertes Administratorkonto für Systems Manager zu erstellen, können Sie den Befehl `register-delegated-administrator` mit der AWS CLI verwenden, wie im folgenden Beispiel gezeigt.

```
aws organizations register-delegated-administrator \
    --account-id delegated admin account ID \
    --service-principal ssm.amazonaws.com
```

Wenn Sie Automatisierungen für mehrere Konten ausführen möchten, die nicht von AWS Organizations verwaltet werden, empfehlen wir, ein dediziertes Konto für die Automatisierungsverwaltung zu erstellen. Die Ausführung aller kontoübergreifenden Automatisierungen über ein dediziertes Konto vereinfacht die Verwaltung von IAM-Berechtigungen, die Fehlerbehebung und schafft eine Trennungsebene zwischen Betrieb und Verwaltung. Dieser Ansatz wird auch empfohlen, wenn Sie AWS Organizations verwenden, aber nur einzelne Konten und keine Organisationseinheiten ansprechen möchten.

So funktioniert das Ausführen von Automatisierungen

Die Ausführung von Automatisierungen über mehrere Regionen und Konten oder OUs hinweg funktioniert wie folgt:

1. Stellen Sie sicher, dass alle Ressourcen, auf denen Sie die Automatisierung ausführen möchten, in allen Regionen und Konten oder OUs identische Tags verwenden. Ist dies nicht der Fall, können Sie sie einer AWS-Ressourcengruppe hinzufügen und diese Gruppe als Ziel verwenden. Weitere Informationen finden Sie unter Was sind Ressourcengruppen? im AWS Resource Groups- und Tags-Benutzerhandbuch.
2. Melden Sie sich bei dem Konto an, das Sie als zentrales Automation-Konto konfigurieren möchten.
3. Verwenden Sie das Einrichten von Managementkonto-Berechtigungen für regionen- und kontenübergreifende Automatisierungen.-Verfahren in diesem Thema, um die folgenden IAM-Rollen zu erstellen:
   - **`AWS-SystemsManager-AutomationAdministrationRole`** – Diese Rolle gewährt dem Benutzer die Berechtigung zur Ausführung von Automatisierungen in mehreren Konten und OUs.
   - **`AWS-SystemsManager-AutomationExecutionRole`** – Diese Rolle erteilt dem Benutzer die Berechtigung, Automatisierungen in den Zielkonten auszuführen.
4. Wählen Sie das Runbook, die Regionen und Konten oder OUs, in denen Sie die Automatisierung ausführen möchten.

> **Note**
>
> Automatisierungen laufen nicht rekursiv über Organisationseinheiten. Stellen Sie sicher, dass die Zielorganisationseinheit die gewünschten Konten enthält. Wenn Sie ein benutzerdefiniertes Runbook auswählen, muss das Runbook für alle Zielkonten freigegeben werden. Weitere Informationen zum Teilen von Runbooks finden Sie unter Freigeben von SSM-Dokumenten. Weitere Informationen zur Verwendung von freigegebenen Runbooks finden Sie unter Verwenden von freigegebenen SSM-Dokumenten.

5. Führen Sie die Automatisierung aus.

> **Note**
>
> Wenn Sie Automatisierungen über mehrere Regionen, Konten oder OUs hinweg ausführen, startet die Automatisierung, die Sie über das primäre Konto ausführen, untergeordnete Automatisierungen in jedem der Zielkonten. Die Automatisierung im primären Konto enthält `aws:executeAutomation`-Schritte für jedes der Zielkonten. Wenn Sie eine Automatisierung aus neuen Regionen starten, die nach dem 20. März 2019 gestartet wurden, und auf eine Region abzielen, die standardmäßig aktiviert ist, schlägt die Automatisierung fehl. Wenn Sie eine Automatisierung aus einer Region starten, die standardmäßig aktiviert ist, und auf eine Region abzielen, die Sie aktiviert haben, wird die Automatisierung erfolgreich ausgeführt.

6. Verwenden Sie die API-Vorgänge GetAutomationExecution, DescribeAutomationStepExecutions und DescribeAutomationExecutions über die AWS Systems Manager-Konsole oder die AWS CLI, um den Fortschritt der Automatisierung zu überwachen. Die Ausgabe der Schritte für die Automatisierung in Ihrem primären Konto wird die `AutomationExecutionId` der untergeordneten Automatisierungen sein. Um die Ausgabe der untergeordneten Automatisierungen anzuzeigen, die in Ihren Zielkonten erstellt wurden, müssen Sie das entsprechende Konto, die Region und die `AutomationExecutionId` In Ihrer Anfrage angeben.

Einrichten von Managementkonto-Berechtigungen für regionen- und kontenübergreifende Automatisierungen.

Verwenden Sie das folgende Verfahren, um die erforderlichen IAM-Rollen für die Systems Manager Automation regionen- und kontenübergreifende Ausführung von Automation mit

AWS CloudFormation zu erstellen. In diesem Verfahren wird beschrieben, wie Sie die **`AWS-SystemsManager-AutomationAdministrationRole`**-Rolle erstellen. Sie müssen nur diese Rolle im zentralen Automation-Konto erstellen. In diesem Verfahren wird auch beschrieben, wie Sie die **`AWS-SystemsManager-AutomationExecutionRole`**-Rolle erstellen. Sie müssen diese Rolle in jedem Konto erstellen, das für die Ausführung von regionen- und kontenübergreifenden Automatisierungen verwendet werden soll. Wir empfehlen die Verwendung von AWS CloudFormation-StackSets, um die **`AWS-SystemsManager-AutomationExecutionRole`**-Rolle in den Konten, für die Sie regionen- und kontenübergreifende Automatisierungen ausführen möchten.

So erstellen Sie die erforderlichen IAM-Administrator-Rollen für regionen- und kontenübergreifende Automatisierungen mit AWS CloudFormation

1. Laden Sie das `AWS-SystemsManager-AutomationAdministrationRole.zip` herunter und entpacken Sie es. Oder wenn Ihre Konten von AWS Organizations `AWS-SystemsManager-AutomationAdministrationRole (org).zip` verwaltet werden. Diese Datei enthält die `AWS-SystemsManager-AutomationAdministrationRole.yaml` AWS CloudFormation-Vorlagendatei.
2. Öffnen Sie die AWS CloudFormation-Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie Stack erstellen aus.
4. Wählen Sie im Abschnitt Specify template (Vorlage angeben) die Option Upload a template (Vorlage hochladen).
5. Wählen Sie Choose file (Datei auswählen) und dann die `AWS-SystemsManager-AutomationAdministrationRole.yaml` AWS CloudFormation-Vorlagendatei.
6. Wählen Sie Next (Weiter).
7. Geben Sie auf der Seite Specify Stack details (Stack-Details angeben) im Feld Stack name (Stack-Name) einen Namen ein.
8. Wählen Sie Next (Weiter).
9. Geben Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Werte für die Optionen ein, die Sie verwenden möchten. Wählen Sie Next (Weiter).
10. Scrollen Sie auf der Seite Review (Prüfung) nach unten, und wählen Sie die Option I acknowledge that AWS CloudFormation might create IAM resources with custom names (Ich bin mir bewusst, dass IAM-Ressourcen mit benutzerdefinierten Namen erstellen kann) aus.
11. Wählen Sie Stack erstellen aus.

AWS CloudFormation zeigt etwa drei Minuten den Status CREATE_IN_PROGRESS an. Der Status wechselt zu CREATE_COMPLETE.

Sie müssen die folgende Vorgehensweise in jedem Konto, für das Sie regionen- und kontenübergreifende Automatisierungen ausführen möchten, wiederholen.

So erstellen Sie die erforderlichen IAM-Automatisierungsrollen für regionen- und kontenübergreifende Automatisierungen mit AWS CloudFormation

1. Laden Sie das `AWS-SystemsManager-AutomationExecutionRole.zip` herunter. Oder wenn Ihre Konten von AWS Organizations `AWS-SystemsManager-AutomationExecutionRole (org).zip` verwaltet werden. Diese Datei enthält die `AWS-SystemsManager-AutomationExecutionRole.yaml` AWS CloudFormation-Vorlagendatei.
2. Öffnen Sie die AWS CloudFormation-Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie Stack erstellen aus.
4. Wählen Sie im Abschnitt Specify template (Vorlage angeben) die Option Upload a template (Vorlage hochladen).
5. Wählen Sie Choose file (Datei auswählen) und dann die `AWS-SystemsManager-AutomationExecutionRole.yaml` AWS CloudFormation-Vorlagendatei.
6. Wählen Sie Next (Weiter).
7. Geben Sie auf der Seite Specify Stack details (Stack-Details angeben) im Feld Stack name (Stack-Name) einen Namen ein.
8. Geben Sie im Abschnitt Parameters (Parameter) im Feld AdminAccountId (Admin-Konto-ID) die ID für das Automation-Zentralkonto ein.
9. Wenn Sie diese Rolle für eine AWS Organizations-Umgebung einrichten, gibt es ein anderes Feld im Abschnitt namens OrganizationID (Organisations-ID). Geben Sie die ID Ihrer AWS-Organisation ein.
10. Wählen Sie Next (Weiter).
11. Geben Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Werte für die Optionen ein, die Sie verwenden möchten. Wählen Sie Next (Weiter).
12. Scrollen Sie auf der Seite Review (Prüfung) nach unten, und wählen Sie die Option I acknowledge that AWS CloudFormation might create IAM resources with custom names (Ich bin mir bewusst, dass IAM-Ressourcen mit benutzerdefinierten Namen erstellen kann) aus.
13. Wählen Sie Stack erstellen aus.

AWS CloudFormation zeigt etwa drei Minuten den Status CREATE_IN_PROGRESS an. Der Status wechselt zu CREATE_COMPLETE.

Ausführen von Automatisierungen in mehreren Regionen und Konten (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine Automatisierung in mehreren Regionen und Konten über das Automation-Managementkonto ausführen.

Bevor Sie beginnen

Bevor Sie das folgende Verfahren ausführen, beachten Sie die folgenden Informationen:

- Der Benutzer oder die Rolle, mit der Sie eine Automation für mehrere Regionen oder Konten ausführen, muss über die Berechtigung `iam:PassRole` für die Rolle `AWS-SystemsManager-AutomationAdministrationRole` verfügen.
- AWS-Konto-IDs oder OUs, in denen Sie die Automatisierung ausführen möchten.
- Von Systems Manager unterstützte Regionen, in denen Sie die Automatisierung ausführen möchten.
- Den Tag-Schlüssel und den Tag-Wert oder den Namen der Ressourcengruppe für die Ausführung der Automatisierung.

So führen Sie eine Automatisierung in mehreren Regionen und Konten aus

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Automation (Automatisierung) und Execute automation (Automatisierung ausführen) aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Mit mir geteilt-Registerkarte. Um alle Runbooks anzuzeigen, wählen Sie die Alle Dokumente-Registerkarte.

> **Note**
>
> Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Document details (Dokument-Details), ob Document version (Dokumentversion) auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.

5. Wählen Sie Next (Weiter).
6. Wählen Sie auf der Seite Execute automation document (Automation-Dokument ausführen) die Option Multi-account and Region (Mehrere Konten und Regionen).
7. Verwenden Sie im Abschnitt Target accounts and Regions (Ziel-Konten und -Regionen) das Feld Accounts and organizational (OUs) (Konten und Organisationseinheiten) zur Angabe der verschiedenen AWS-Konten oder AWS-Organisationseinheiten für die Ausführung der Automatisierung an einem bestimmten Ort. Separieren Sie mehrere Konten oder OUs durch Kommata.
8. Verwenden Sie die Liste AWS-Regionen zur Auswahl einer oder mehrerer Regionen für die Ausführung der Automatisierung.
9. Verwenden Sie die Optionen für Multi-Region and account rate control (Regionen- und kontenübergreifende Ratensteuerung), um die Automatisierungen auf eine begrenzte Anzahl von Konten in einer begrenzten Anzahl von Regionen zu beschränken. Diese Optionen schränken nicht die Anzahl der AWS-Ressourcen ein, die die Automatisierungen ausführen können.

   a. Wählen Sie im Abschnitt Location (account-Region pair) concurrency (Standort- (Konto-Region-Paar) Gleichzeitigkeit) eine Option, um die Anzahl der Automatisierungen zu begrenzen, die gleichzeitig in mehreren Konten und Regionen ausgeführt werden können. Beispiel: Wenn Sie eine Automatisierung in fünf (5) AWS-Konten-Konten in vier (4) AWS-

Regionen ausführen möchten, führt Systems Manager die Automatisierungen in insgesamt 20 Konto-Region-Paaren aus. Sie können diese Option verwenden, um eine absolute Zahl anzugeben, z. B. **2**, sodass die Automatisierung nur in 2 Konto-Region-Paaren gleichzeitig ausgeführt wird. Sie können aber auch einen Prozentsatz der Konto-Region-Paare angeben, der gleichzeitig ausgeführt werden kann. Beispielsweise geben Sie bei 20 Konto-Region-Paaren 20 % an: Dann wird die Automatisierung in maximal fünf (5) Konto-Region-Paaren gleichzeitig ausgeführt.

- Wählen Sie targets (Ziele) aus, um eine absolute Anzahl von Konto-Region-Paaren einzugeben, die die Automatisierung gleichzeitig ausführen können.
- Wählen Sie percent (Prozent) aus, um einen Prozentsatz von Konto-Region-Paaren einzugeben, die die Automatisierung gleichzeitig ausführen können.

b. Wählen Sie im Abschnitt Error threshold (Fehlerschwellenwert) eine Option aus:

- Wählen Sie errors (Fehler), um eine absolute Anzahl von zulässigen Fehlern anzugeben, bevor die Automation damit aufhört, die Automatisierung an andere Ressourcen zu senden.
- Wählen Sie percentage aus, um einen Prozentsatz von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, die Automatisierung an andere Ressourcen zu senden.

10. Wählen Sie im Abschnitt Targets (Ziele) die Ausrichtung auf die AWS-Ressourcen für die Ausführung der Automation. Diese Optionen sind erforderlich.

    a. Wählen Sie in der Liste Parameter (Parameter) einen Parameter aus. Die Elemente in der Liste Parameter richten sich nach den Parametern in dem Automation-Runbook, das Sie zu Beginn dieses Verfahrens ausgewählt haben. Durch Auswahl eines Parameters legen Sie den Typ der Ressource fest, für die der Automation-Workflow ausgeführt wird.

    b. Wählen Sie in der Liste Targets (Ziele)aus, wie Sie Ressourcen als Ziele verwenden möchten.

        i. Wenn Sie die Zielressourcen mithilfe von Parameterwerten ausgewählt haben, geben Sie den Parameterwert für den gewählten Parameter im Feld Eingabeparameter eingeben ein.

        ii. Wenn Sie die Zielressourcen mit AWS Resource Groups ausgewählt haben, wählen Sie den Namen der Gruppe aus der Liste Resource Group (Ressourcengruppe) aus.

iii. Wenn Sie die Zielressourcen mithilfe von Tags ausgewählt haben, geben Sie den Tag-Schlüssel und (optional) den Tag-Wert in die entsprechenden Felder ein. Wählen Sie Add (Hinzufügen) aus.

iv. Wenn Sie ein Automatisierungs-Runbook für alle Instances im aktuellen AWS-Konto und AWS-Region aus. Wählen Sie und anschließend Alle Instances aus.

11. Geben Sie im Abschnitt Input Parameters (Eingabeparameter) die erforderlichen Eingaben an. Wählen Sie die `AWS-SystemsManager-AutomationAdministrationRole`-IAM-Servicerolle aus der Liste AutomationAssumeRole.

> **Note**
>
> Möglicherweise müssen Sie einige der Optionen im Abschnitt Input parameters (Eingabeparameter) nicht auswählen. Dies liegt daran, dass Sie Ressourcen in mehreren Regionen und Konten mithilfe von Tags oder einer Ressourcengruppe als Ziele ausgewählt haben. Wenn Sie beispielsweise das `AWS-RestartEC2Instance`-Runbook ausgewählt haben, müssen Sie keine Instance-IDs im Abschnitt Input parameters (Eingabeparameter) angeben oder auswählen. Die Automatisierung sucht die Instances für den Neustart mit den von Ihnen angegebenen Tags.

12. (Optional) Wählen Sie einen CloudWatch-Alarm aus, der auf Ihre Automatisierung zur Überwachung angewendet werden soll. Um einen CloudWatch-Alarm an Ihre Automatisierung anzuhängen, muss der IAM-Prinzipal, der die Automatisierung startet, über die Berechtigung für die `iam:createServiceLinkedRole`-Aktion verfügen. Weitere Informationen zu CloudWatch-Alarmen erhalten Sie unter Verwendung von Amazon-CloudWatch-Alarmen. Beachten Sie, dass, wenn Ihr Alarm ausgelöst wird, die Automatisierung abgebrochen wird und alle von Ihnen definierten `0nCancel`-Schritte ausgeführt werden. Wenn Sie AWS CloudTrail verwenden, sehen Sie den API-Aufruf in Ihrem Trail.

13. Verwenden Sie die Optionen im Abschnitt Rate control (Ratensteuerung), um die Anzahl der AWS-Ressourcen zu beschränken, welche die Automatisierung mit jedem Konto-Region-Paar ausführen können.

    Wählen Sie im Abschnitt Concurrency (Gleichzeitigkeit) eine Option aus:

    - Wählen Sie targets (Ziele) aus, um eine absolute Anzahl von Zielen einzugeben, die den Automation-Workflow gleichzeitig ausführen können.
    - Wählen Sie percentage (Prozentsatz) aus, um einen Prozentsatz der Ziele anzugeben, die den Automation-Workflow gleichzeitig ausführen können.

14. Wählen Sie im Abschnitt Error threshold (Fehlerschwellenwert) eine Option aus:
    - Wählen Sie errors (Fehler), um eine absolute Anzahl von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, den Workflow an andere Ressourcen zu senden.
    - Wählen Sie percentage (Prozentsatz) aus, um einen Prozentsatz von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, den Workflow an andere Ressourcen zu senden.
15. Wählen Sie Execute (Ausführen).

Ausführen von Automatisierungen in mehreren Regionen und Konten (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie Sie die AWS CLI (unter Linux oder Windows) oder AWS Tools for PowerShell verwenden, um eine Automatisierung in mehreren Regionen und Konten über das Automation-Managementkonto auszuführen.

Bevor Sie beginnen

Bevor Sie das folgende Verfahren ausführen, beachten Sie die folgenden Informationen:

- AWS-Konto-IDs oder OUs, in denen Sie die Automatisierung ausführen möchten.
- Von Systems Manager unterstützte Regionen, in denen Sie die Automatisierung ausführen möchten.
- Den Tag-Schlüssel und den Tag-Wert oder den Namen der Ressourcengruppe für die Ausführung der Automatisierung.

So führen Sie eine Automatisierung in mehreren Regionen und Konten aus

1. Installieren und konfigurieren Sie die AWS CLI oder AWS Tools for PowerShell, falls noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.
2. Verwenden Sie das folgende Format, um eine Automatisierung in mehreren Regionen und Konten auszuführen. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

   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" }
```

Im Folgenden finden Sie einige Beispiele.

Beispiel 1: In diesem Beispiel werden EC2-Instances in den Konten `123456789012` und `987654321098` in den Regionen `us-east-2` und `us-west-1` neu gestartet. Die Instances müssen mit dem Tag-Schlüsselpaarwert `Env-PROD` markiert sein.

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" }
```

Beispiel 2: In diesem Beispiel werden EC2-Instances in den Konten `123456789012` und `987654321098` in der Region `eu-central-1` neu gestartet. Die Instances müssen Mitglieder der AWS-Ressourcengruppe `prod-instances` sein.

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" }
```

Beispiel 3: In diesem Beispiel werden EC2-Instances in der AWS-Organisationseinheit (OU) `ou-1a2b3c-4d5e6c` neu gestartet. Die Instances befinden sich in den Regionen `us-west-1` und `us-west-2`. Die Instances müssen Mitglieder der AWS-Ressourcengruppe `WebServices` sein.

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" }
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

Linux & macOS

```
{
        "AutomationExecutionId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
    }
```

Windows

```
{
        "AutomationExecutionId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
    }
```

PowerShell

```
4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

3. Führen Sie den folgenden Befehl aus, um Details zu der Automatisierung anzuzeigen. Ersetzen Sie *automation execution ID* (Automatisierungs-Ausführungs-ID) mit Ihren eigenen Informationen.

   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. Führen Sie den folgenden Befehl aus, um Details über den Automatisierungsprozess anzuzeigen.

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**
>
> Sie können auch den Status der Automatisierung in der Konsole überwachen. Wählen Sie in der Liste Automatisierungs-Ausführung die Automatisierung, die Sie gerade ausgeführt haben, und wählen Sie dann die Registerkarte Execution steps (Ausführungsschritte). Diese Registerkarte zeigt Ihnen den Status der Automatisierungs-Aktionen.

Weitere Informationen

- Zentralisiertes regions- und kontenübergreifendes Patching mit AWS Systems Manager Automation

Ausführen von Automatisierungen basierend auf Ereignissen

Sie können eine Automatisierung starten, indem Sie ein Runbook als Ziel eines Amazon EventBridge-Ereignisses angeben. Sie können Automatisierungen nach einem Zeitplan oder beim Eintreten eines bestimmten AWS -Ereignisses starten. Angenommen, Sie erstellen ein Runbook mit dem Namen BootStrapInstances, das Software auf einer Instance installiert, wenn eine Instance gestartet wird. Um das BootStrapInstances Runbook (und die entsprechende Automatisierung) als Ziel eines EventBridge Ereignisses anzugeben, erstellen Sie zunächst eine neue EventBridge Regel. (Beispiel für eine Regel: Service name (Servicename): EC2, Event Type (Ereignistyp): EC2 Instance State-change Notification, Specific state(s) (Bestimmte Status): running, Any instance (Beliebige Instance).) Anschließend verwenden Sie die folgenden Verfahren, um das BootStrapInstances Runbook mithilfe der EventBridge Konsole und AWS Command Line Interface () als Ziel des Ereignisses anzugebenAWS CLI. Beim Starten einer neuen Instance führt das System die Automatisierung aus und installiert Software.

Weitere Informationen zum Erstellen eines Runbooks finden Sie unter Erstellen Ihrer eigenen Runbooks.

Erstellen eines EventBridge Ereignisses, das ein Runbook verwendet (Konsole)

Gehen Sie wie folgt vor, um ein Runbook als Ziel eines EventBridge Ereignisses zu konfigurieren.

So konfigurieren Sie ein Runbook als Ziel einer EventBridge Ereignisregel

1. Öffnen Sie die Amazon- EventBridge Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
5. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf übereinstimmende Ereignisse reagiert, die von Ihrem eigenen

stammen AWS-Konto, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.

6. Wählen Sie aus, wie die Regel ausgelöst wird.

So erstellen Sie eine Regel auf der Basis von …	Vorgehensweise	
Ereignis	a. Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus. b. Wählen Sie Weiter aus. c. Wählen Sie für Ereignisquelle die Option AWS Ereignisse oder EventBridge Partnerereignisse aus. d. Führen Sie im Abschnitt Event pattern (Ereignismuster) einen der folgenden Schritte aus: • Um eine Vorlage zum Erstellen Ihres Ereignismusters zu verwenden, wählen Sie Event pattern form (Ereignismusterformular) und wählen Sie Event source (Ereignisquelle), AWS service (-Service) und Event type (Ereignistyp). Wenn Sie Alle Ereignisse als Ereignistyp auswählen, stimmen	

So erstellen Sie eine Regel auf der Basis von …	Vorgehensweise	
	alle vom ausgegebenen Ereignisse mit der Regel AWS-Service überein. Um die Vorlage anzupassen, wählen Sie Custom pattern (JSON editor) (Benutzerdefiniertes Muster (JSON-Editor)) und nehmen Sie die erforderlichen Änderungen vor. • Wenn Sie ein benutzerdefiniertes Ereignismuster verwenden möchten, wählen Sie Custom pattern (JSON editor) (Benutzerdefiniertes Muster (JSON-Editor)) und erstellen Sie Ihr Ereignismuster.	

So erstellen Sie eine Regel auf der Basis von …	Vorgehensweise	
Plan	a. Wählen Sie unter Rule type (Regeltyp) die Option Schedule (Zeitplan) aus. b. Wählen Sie Weiter aus. c. Gehen Sie bei Schedule pattern (Zeitplanmuster) wie folgt vor: • Um den Zeitplan mithilfe eines Cron-Ausdrucks zu definieren, wählen Sie A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month (Detaillierter Zeitplan, der zu einem bestimmten Zeitpunkt (z. B. 8:00 Uhr) PST am ersten Montag jedes Monats PST ausgeführt wird) und geben Sie den Cron-Ausdruck ein. • Um den Zeitplan mithilfe eines Rate-Ausdrucks zu definieren, wählen Sie A schedule that runs at a regular rate, such as every 10 minutes (Zeitplan, der mit einer regulären Rate läuft, z. B. alle 10 Minuten)	

So erstellen Sie eine Regel auf der Basis von …	Vorgehensweise	
	und geben Sie den Rate-Ausdruck ein.	

7. Wählen Sie Weiter aus.
8. Bei Target types (Zieltypen) wählen Sie AWS -Service aus.
9. Für Select target (Ziel auswählen), wählen Sie Systems Manager Automation.
10. Wählen Sie für Dokument ein Runbook aus, das Sie verwenden möchten, wenn das Ziel aufgerufen wird.
11. Behalten Sie im Abschnitt Configure automation parameter(s) (Automatisierungsparameter konfigurieren) entweder die Standardparameterwerte bei (sofern verfügbar) oder geben Sie Ihre eigenen Werte ein.

> **Note**
>
> Um ein Ziel zu erstellen, müssen Sie bei jedem erforderlichen Parameter einen Wert angeben. Wenn Sie dies nicht tun, erstellt das System die Regel, aber die Regel wird nicht ausgeführt.

12. Bei vielen Zieltypen EventBridge benötigt Berechtigungen zum Senden von Ereignissen an das Ziel. In diesen Fällen EventBridge kann die IAM-Rolle erstellen, die für die Ausführung Ihrer Regel erforderlich ist. Führen Sie eine der folgenden Aktionen aus:

    - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Create a new role for this specific resource (Eine neue Rolle für diese spezifische Ressource erstellen).
    - Wenn Sie eine zuvor erstellte IAM-Rolle verwenden möchten, wählen Sie Use existing role (Vorhandene Rolle verwenden) und wählen Sie die vorhandene Rolle aus der Dropdown-Liste aus. Beachten Sie, dass Sie möglicherweise die Vertrauensrichtlinie für Ihre IAM-Rolle aktualisieren müssen, um einzuschließen EventBridge. Im Folgenden wird ein Beispiel gezeigt:

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

```
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "events.amazonaws.com",
                    "ssm.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

13. Wählen Sie Weiter aus.
14. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Markieren Ihrer Amazon- EventBridge Ressourcen im Amazon- EventBridge Benutzerhandbuch.
15. Wählen Sie Weiter aus.
16. Überprüfen Sie die Details der Regel und wählen Sie dann Create rule (Regel erstellen) aus.

Erstellen eines EventBridge Ereignisses, das ein Runbook verwendet (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie Sie die AWS CLI (unter Linux oder Windows) oder verwenden AWS Tools for PowerShell , um eine EventBridge Ereignisregel zu erstellen und ein Runbook als Ziel zu konfigurieren.

So konfigurieren Sie ein Runbook als Ziel einer EventBridge Ereignisregel

1. Installieren und konfigurieren Sie die AWS CLI oder die AWS Tools for PowerShell, falls noch nicht geschehen.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.
2. Erstellen Sie einen Befehl, um eine neue EventBridge Ereignisregel anzugeben. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Auslöser nach Zeitplan

   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"
```

Im folgenden Beispiel wird eine EventBridge Ereignisregel erstellt, die jeden Tag um 9:00 Uhr (UTC) beginnt.

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 * * ? *)"
```

Auslöser basierend auf einem Ereignis

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"]}'
```

Im folgenden Beispiel wird eine EventBridge Ereignisregel erstellt, die beginnt, wenn eine EC2-Instance in der Region den Status ändert.

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"]}'
```

Der Befehl gibt Details für die neue EventBridge Regel ähnlich der folgenden zurück.

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. Erstellen Sie einen Befehl, um ein Runbook als Ziel der EventBridge Ereignisregel anzugeben, die Sie in Schritt 2 erstellt haben. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

```
aws events put-targets \
--rule rule name \
--targets '{"Arn": "arn:aws:ssm:region:account ID:automation-definition/runbook
 name","Input":"{\"input parameter\":[\"value\"],\"AutomationAssumeRole\":
[\"arn:aws:iam::123456789012:role/AutomationServiceRole\"]}","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":"{\"input parameter\":[\"value\"],\"AutomationAssumeRole\":
[\"arn:aws:iam::123456789012:role/AutomationServiceRole\"]}","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
```

Im folgenden Beispiel wird ein EventBridge Ereignisziel erstellt, das die angegebene Instance-ID mit dem Runbook startet`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
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
"FailedEntries": [],
"FailedEntryCount": 0
}
```

Windows

```
{
"FailedEntries": [],
"FailedEntryCount": 0
}
```

PowerShell

Es gibt keine Ausgabe, wenn der Befehl für erfolgreich ist PowerShell.

Führen Sie eine Automatisierung manuell aus

In den folgenden Verfahren wird beschrieben, wie Sie mit der AWS Systems Manager-Konsole und AWS Command Line Interface (AWS CLI) eine Automatisierung mithilfe des manuellen Ausführungsmodus ausführen. Im manuellen Ausführungsmodus startet die Automatisierung in einem Wartestatus und verharrt zwischen den einzelnen Schritten im Wartestatus. So können Sie steuern, wann der Automatisierung fortgesetzt wird. Dies ist hilfreich, wenn Sie das Ergebnis eines Schritts überprüfen müssen, bevor Sie fortfahren.

Die Automatisierung wird im Kontext des aktuellen Benutzers ausgeführt. Das bedeutet, dass Sie keine zusätzlichen IAM-Berechtigungen konfigurieren müssen, solange Sie über die Berechtigung zum Ausführen des Runbooks verfügen und alle Aktionen von dem Runbook aufgerufen werden. Wenn Sie über Administrator-Berechtigungen in IAM verfügen, haben Sie bereits die Berechtigung zum Ausführen dieser Automatisierung.

Ausführen einer Automatisierung Schritt für Schritt (Konsole)

Das folgende Verfahren zeigt, wie Sie mithilfe der Systems Manager-Konsole eine Automatisierung Schritt für Schritt manuell ausführen.

So führen Sie einen Automatisierung Schritt für Schritt aus

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Automation (Automatisierung) und Execute automation (Automatisierung ausführen) aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Mit mir geteilt-Registerkarte. Um alle Runbooks anzuzeigen, wählen Sie die Alle Dokumente-Registerkarte.

   > **Note**
   >
   > Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Document details (Dokument-Details), ob Document version (Dokumentversion) auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.
5. Wählen Sie Next (Weiter).
6. Wählen Sie im Abschnitt Execution mode (Ausführungsmodus) die Option Manual execution (Manuelle Ausführung) aus.
7. Geben Sie im Abschnitt Input Parameters (Eingabeparameter) die erforderlichen Eingaben an. Sie können optional eine IAM-Servicerolle aus der Liste AutomationAssumeRole auswählen.
8. Wählen Sie Execute (Ausführen).
9. Wählen Sie Execute this step (Diesen Schritt ausführen) aus, wenn Sie zum ersten Schritt der Automatisierung bereit sind. Die Automatisierung fährt mit Schritt 1 fort und hält an, bevor die weiteren Schritte des Runbooks, das Sie in Schritt 3 dieses Verfahrens ausgewählt haben, ausgeführt werden. Wenn das Runbook mehrere Schritte umfasst, müssen Sie für jeden Schritt Execute this step (Diesen Schritt ausführen) auswählen, damit die Automatisierung fortgesetzt wird. Jedes Mal, wenn Sie diesen Schritt ausführen, wird die Aktion ausgeführt.

   > **Note**
   >
   > Die Konsole zeigt den Status der Automatisierung an. Wenn die Automatisierung einen Schritt nicht ausführen kann, finden Sie weitere Informationen unter Fehlerbehebung für Systems Manager Automation..
10. Nachdem Sie alle Schritte im Runbook abgeschlossen haben, wählen Sie Complete and view results (Abschließen und Ergebnisse anzeigen) aus, um die Automatisierung zu beenden und die Ergebnisse anzuzeigen.

Ausführen einer Automatisierung Schritt für Schritt (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie Sie die AWS CLI (auf Linux, macOS oder Windows) oder AWS Tools for PowerShell verwenden, um eine Automatisierung Schritt für Schritt manuell auszuführen.

So führen Sie einen Automatisierung Schritt für Schritt aus

1. Installieren und konfigurieren Sie die AWS CLI oder AWS Tools for PowerShell, falls noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

2. Führen Sie den folgenden Befehl aus, um eine manuelle Automatisierung zu starten. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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
   ```

   Hier sehen Sie ein Beispiel, wie Sie das `AWS-RestartEC2Instance`-Runbook verwenden, um die angegebene EC2-Instance neu zu starten.

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"}
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
    "AutomationExecutionId": "ba9cd881-1b36-4d31-a698-0123456789ab"
}
```

Windows

```
{
    "AutomationExecutionId": "ba9cd881-1b36-4d31-a698-0123456789ab"
}
```

PowerShell

```
ba9cd881-1b36-4d31-a698-0123456789ab
```

3. Führen Sie den folgenden Befehl aus, wenn Sie bereit sind, den ersten Schritt der Automatisierung zu starten. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Die Automatisierung fährt mit Schritt 1 fort und hält an, bevor die weiteren Schritte des Runbooks, das Sie in Schritt 1 dieses Verfahrens ausgewählt haben, ausgeführt werden. Wenn das Runbook mehrere Schritte umfasst, müssen Sie den folgenden Befehl für jeden Schritt ausführen, damit die Automatisierung fortfahren kann.

   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"}
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

4. Führen Sie den folgenden Befehl aus, um den Status jeder Schrittausführung in der Automatisierung abzurufen.

   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
```

Das System gibt unter anderem folgende Informationen zurück

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. Führen Sie den folgenden Befehl aus, um die Automatisierung abzuschließen, nachdem alle im ausgewählten Runbook angegebenen Schritte abgeschlossen sind. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Planung von Automatisierungen

Die folgenden Themen enthalten Informationen darüber, wie Sie Automatisierungen so planen, dass sie in einem bestimmten Intervall oder zu einem bestimmten von Ihnen angegebenen Zeitpunkt ausgeführt werden.

Inhalt

- Planen von Automatisierungen mit State Manager-Zuordnungen
- Planen von Automatisierungen mit Wartungsfenstern

Planen von Automatisierungen mit State Manager-Zuordnungen

Sie können eine Automatisierung starten, indem Sie eine State Manager-Verknüpfung mit einem Runbook erstellen. State Manager ist eine Funktion von AWS Systems Manager. Durch das Erstellen einer AWS-Verknüpfung mit einem Runbook können Sie verschiedene Arten von State Manager-Ressourcen als Ziel verwenden. Beispielsweise können Sie Zuordnungen erstellen, die einen gewünschten Status auf einer AWS-Ressource erzwingen, einschließlich Folgendem:

- Fügen Sie eine Systems Manager Rolle an Amazon Elastic Compute Cloud (Amazon EC2) - Instances an, um sie auf verwaltete Instances zu ändern.
- Erzwingen Sie die gewünschten Eingangs- und Ausgangsregeln für eine Sicherheitsgruppe.
- Erstellen oder löschen Sie Amazon DynamoDB-Backups.
- Erstellen oder löschen Sie Amazon Elastic Block Store (Amazon EBS)-Snapshots.
- Deaktivieren Sie Lese- und Schreibberechtigungen für Amazon Simple Storage Service (Amazon S3)-Buckets.
- Starten, Stoppen oder starten Sie verwaltete Instances und Amazon Relational Database Service (Amazon RDS)-Instances neu.
- Anwenden von Patches auf Linux, macOS und Windows AMIs.

Gehen Sie wie folgt vor, um eine State Manager-Zuordnung zu erstellen, die eine Automatisierung mithilfe der AWS Systems Manager-Konsole oder AWS Command Line Interface (AWS CLI) ausführt.

Bevor Sie beginnen

Beachten Sie die folgenden wichtigen Details, bevor Sie eine Automatisierung mithilfe von State Manager ausführen:

- Bevor Sie eine Zuordnung erstellen können, die ein Runbook verwendet, stellen Sie sicher, dass Sie Berechtigungen für Automation konfiguriert haben, eine Funktion von AWS Systems Manager. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
- State Manager-Zuordnungen, die Runbooks verwenden, zählen zu der maximalen Anzahl der gleichzeitig ausgeführten Automatisierungen in Ihrem AWS-Konto hinzu. Sie können maximal 100 Automatisierungen gleichzeitig ausführen. Informationen finden Sie unter Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz.
- Beim Ausführen einer Automatisierung protokolliert State Manager nicht die API-Operationen, die von der Automatisierung in AWS CloudTrail initiiert wurden.

- Systems Manager erstellt automatisch eine serviceverknüpfte Rolle, damit State Manager die Berechtigung hat, API-Vorgänge von Systems Manager Automation aufzurufen. Wenn Sie möchten, können Sie die serviceverknüpfte Rolle selbst erstellen, indem Sie den folgenden Befehl über die AWS CLI oder AWS Tools for PowerShell ausführen.

  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
  ```

  Weitere Informationen zu Service-verknüpften Rollen finden Sie unter Verwenden von serviceverknüpften Rollen für Systems Manager.

Erstellen einer Zuordnung, die eine Automatisierung ausführt (Konsole)

Im folgenden Verfahren wird beschrieben, wie mithilfe der Systems Manager-Konsole eine State Manager-Zuordnung erstellt wird, die eine Automatisierung ausführt.

So erstellen Sie eine State Manager-Zuordnung zum Ausführen einer Automatisierung

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager und anschließend Create association (Zuordnung erstellen) aus.
3. Geben Sie im Feld Name einen Namen an. Dies ist zwar optional, wird aber empfohlen.
4. Wählen Sie in der Liste Document ein Runbook aus. Verwenden Sie die Suchleiste, um nach allen Runbooks mit Document type : Equal : Automation zu filtern. Zur Anzeige von weiteren Runbooks verwenden Sie die Zahlen rechts neben der Suchleiste.

> **Note**
>
> Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

5. Wählen Sie Simple execution (Einfache Ausführung) aus, um die Automatisierung auf einem oder mehreren Zielen auszuführen, indem Sie die Ressourcen-ID für diese Ziele angeben. Wählen Sie Rate control (Ratensteuerung) aus, um die Automatisierung über eine Flotte von AWS-Ressourcen auszuführen, indem Sie eine Ausrichtungsoption wie z. B. Tags oder AWS Resource Groups angeben. Sie können auch die Operation der Automatisierung auf Ihren Ressourcen steuern, indem Sie Gleichzeitigkeits- und Fehlergrenzwerte angeben.

   Wenn Sie Rate control (Ratensteuerung) auswählen, wird der Abschnitt Targets (Ziele) angezeigt.

6. Wählen Sie im Abschnitt Targets (Ziele) eine Methode zur Ausrichtung der Ressourcen aus.

   a. (Erforderlich) Wählen Sie in der Liste Parameter einen Parameter aus. Die Elemente in der Liste Parameter richten sich nach den Parametern in dem Runbook, das Sie zu Beginn dieses Verfahrens ausgewählt haben. Durch Auswahl eines Parameters legen Sie den Typ der Ressource fest, für die die Automatisierung ausgeführt wird.

   b. (Erforderlich) Wählen Sie in der Liste Targets (Ziele) ein Verfahren für die Ausrichtung auf Ressourcen aus.

      - Resource Group (Ressourcengruppe): Wählen Sie den Namen der Gruppe aus der Liste Resource Group (Ressourcengruppe) aus. Weitere Informationen zum Targeting in AWS Resource Groups-Runbooks finden Sie unter Targeting AWS Resource Groups.
      - Tags: Geben Sie den Tag-Schlüssel und (optional) den Tag-Wert in die dafür vorgesehenen Felder ein. Wählen Sie Add (Hinzufügen) aus. Weitere Informationen zum Targeting von Tags in Runbooks finden Sie unter Anzielen eines Tags.
      - Parameter Values (Parameterwerte): Geben Sie die Werte im Abschnitt Input parameters (Eingabeparameter) ein. Wenn Sie mehrere Werte angeben, führt Systems Manager eine untergeordnete Automatisierung für jeden angegebenen Wert aus.

        Nehmen Sie beispielsweise an, dass das Runbook einen InstanceID-Parameter enthält. Wenn Sie die Werte des InstanceID-Parameters beim Ausführen der Automatisierung verwenden, führt Systems Manager eine untergeordnete Automatisierung für

jeden angegebenen Instance-ID-Wert aus. Die übergeordnete Automatisierung ist abgeschlossen, wenn Automatisierung die Ausführung jeder angegebenen Instance abgeschlossen hat oder wenn die Automatisierung fehlschlägt. Sie können maximal 50 Parameterwerte für die Ausrichtung verwenden. Weitere Informationen zum Targeting von Parameterwerten in Runbooks finden Sie unter Ausrichtung auf Parameterwerte.

7. Geben Sie im Abschnitt Input parameters (Eingabeparameter) die erforderlichen Eingabeparameter an.

   Wenn Sie die Zielressourcen mithilfe von Tags oder einer Ressourcengruppe ausgewählt haben, müssen Sie möglicherweise keine der Optionen im Abschnitt Input parameters (Eingabeparameter) auswählen. Wenn Sie beispielsweise das `AWS-RestartEC2Instance`-Runbook und die Ziel-Instances mithilfe von Tags ausgewählt haben, müssen Sie keine Instance-IDs im Abschnitt Input parameters (Eingabeparameter) angeben. Die Automatisierung sucht die Instances für den Neustart mit den von Ihnen angegebenen Tags.

   > **Important**
   >
   > Sie müssen eine Rollen-ARN im Feld AutomationAssumeRole angeben. State Manager verwendet die Übernahmerolle, um AWS-Services aufzurufen, das im Runbook angegeben ist, und führt Automation-Zuordnungen in Ihrem Namen aus.

8. Wählen Sie im Abschnitt Specify schedule (Zeitplan angeben) die Option On Schedule (Nach Zeitplan) aus, wenn Sie die Zuordnungen in regelmäßigen Abständen ausführen möchten. Wenn Sie diese Option auswählen, verwenden Sie die bereitgestellten Optionen zum Erstellen des Zeitplans mithilfe von Cron- oder Rate-Ausdrücken. Weitere Informationen zu Cron- und Rate-Ausdrücken für State Manager finden Sie unter Cron- und Rate-Ausdrücke für Zuordnungen.

   > **Note**
   >
   > Rate-Ausdrücke werden bevorzugt zur Planung für State Manager-Zuordnungen verwendet, die Automatisierungen verwenden ausführen. Rate-Ausdrücke ermöglichen mehr Flexibilität für die Ausführung von Zuordnungen für den Fall, dass Sie die maximale Anzahl von gleichzeitig ausgeführten Automatisierungen erreichen. Mit einem Ratenzeitplan kann Systems Manager die Automatisierung kurz nach dem Empfangen der Benachrichtigungen, dass gleichzeitige Automatisierungen das Maximum erreicht haben und gedrosselt wurden, wiederholen.

Wählen Sie No schedule (Kein Zeitplan)aus, wenn Sie die Zuordnung einmalig ausführen möchten.

9. (Optional) Wählen Sie im Abschnitt Rate Control (Ratenkontrolle) die Optionen Concurrency (Nebenläufigkeit) und Error threshold (Fehlergrenzwert) aus, um die Automatisierungsbereitstellung für Ihre AWS-Ressourcen zu steuern.

   a. Wählen Sie im Abschnitt Concurrency (Gleichzeitigkeit) eine Option aus:

      - Wählen Sie targets (Ziele) aus, um eine absolute Anzahl von Zielen einzugeben, die die Automatisierung gleichzeitig ausführen können.
      - Wählen Sie percentage (Prozentsatz) aus, um einen Prozentsatz der Ziele anzugeben, die den Automatisierung gleichzeitig ausführen können.

   b. Wählen Sie im Abschnitt Error threshold (Fehlerschwellenwert) eine Option aus:

      - Wählen Sie errors (Fehler), um eine absolute Anzahl von zulässigen Fehlern anzugeben, bevor die Automation damit aufhört, die Automatisierung an andere Ressourcen zu senden.
      - Wählen Sie percentage (Prozentsatz) aus, um einen Prozentsatz von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, die Automatisierung an andere Ressourcen zu senden.

   Weitere Informationen zur Verwendung von Zielen und Ratensteuerungen mit Automation finden Sie unter Ausführen von Automatisierungen im großen Maßstab.

10. Wählen Sie Create Association.

> **Important**
>
> Wenn Sie eine Zuordnung erstellen, wird die Zuordnung sofort für die ausgewählten Ziele ausgeführt. Die Zuordnung wird anschließend auf Grundlage des ausgewählten Cron- oder Rate-Ausdrucks ausgeführt. Wenn Sie No schedule (Kein Zeitplan) ausgewählt haben, wird die Zuordnung nicht mehr ausgeführt.

Erstellen einer Zuordnung, die eine Automatisierung ausführt (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie mithilfe der AWS CLI (unter Linux oder Windows) oder AWS Tools for PowerShell eine State Manager-Zuordnung erstellt wird, die eine Automatisierung ausführt.

Bevor Sie beginnen

Bevor Sie das folgende Verfahren ausführen, stellen Sie sicher, dass Sie eine IAM-Servicerolle erstellt haben, die die zum Ausführen des Runbooks erforderlichen Berechtigungen enthält, und eine Vertrauensstellung für die Automatisierung konfiguriert haben, eine Funktion von AWS Systems Manager. Weitere Informationen finden Sie unter Aufgabe 1: Erstellen einer Servicerolle für Automation.

So erstellen Sie eine Zuordnung zum Ausführen einer Automatisierung

1. Installieren und konfigurieren Sie die AWS CLI oder AWS Tools for PowerShell, falls noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.
2. Nutzen Sie den folgenden Befehl, um eine Liste der Dokumente anzuzeigen.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

   Windows

   ```
   aws ssm list-documents
   ```

   PowerShell

   ```
   Get-SSMDocumentList
   ```

   Notieren Sie den Namen des Runbooks, das Sie für die Zuordnung verwenden möchten.
3. Führen Sie den folgenden Befehl aus, um Details des Runbooks einsehen zu können: Ersetzen Sie im folgenden Befehl *runbook name* mit Ihren eigenen Informationen.

Linux & macOS

```
aws ssm describe-document \
--name runbook name
```

Notieren Sie einen Parameternamen (z. B. `InstanceId`), den Sie für die Option `--automation-target-parameter-name` verwenden möchten. Dieser Parameter bestimmt den Typ der Ressource, für die die Automatisierung ausgeführt wird.

Windows

```
aws ssm describe-document ^
--name runbook name
```

Notieren Sie einen Parameternamen (z. B. `InstanceId`), den Sie für die Option `--automation-target-parameter-name` verwenden möchten. Dieser Parameter bestimmt den Typ der Ressource, für die die Automatisierung ausgeführt wird.

PowerShell

```
Get-SSMDocumentDescription `
-Name runbook name
```

Notieren Sie einen Parameternamen (z. B. `InstanceId`), den Sie für die Option `AutomationTargetParameterName` verwenden möchten. Dieser Parameter bestimmt den Typ der Ressource, für die die Automatisierung ausgeführt wird.

4. Erstellen Sie einen Befehl, der eine Automatisierung mithilfe einer State Manager-Zuordnung ausführt. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Ausrichtung mithilfe von 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**
>
> Wenn Sie eine Zuordnung mit der AWS CLI erstellen, können Sie über den Parameter `--targets` auf Ziel-Instances für die Zuordnung angeben. Verwenden Sie nicht den Parameter `--instance-id`. Der Parameter `--instance-id` ist veraltet.

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**
>
> Wenn Sie eine Zuordnung mit der AWS CLI erstellen, können Sie über den Parameter `--targets` auf Ziel-Instances für die Zuordnung angeben. Verwenden Sie nicht den Parameter `--instance-id`. Der Parameter `--instance-id` ist veraltet.

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**
>
> Wenn Sie eine Zuordnung mit der AWS Tools for PowerShell erstellen, können Sie über den Parameter `Target` auf Ziel-Instances für die Zuordnung angeben. Verwenden Sie nicht den Parameter `InstanceId`. Der Parameter `InstanceId` ist veraltet.

Ausrichtung mithilfe von Parameterwerten

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"
```

Ausrichtung mithilfe von 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"
```

Targeting mehrerer Konten und Regionen

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"]
```

Der Befehl gibt Details für die neue Zuordnung zurück, die den folgenden ähneln.

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**
>
> Wenn Sie auf einer oder mehreren Instances eine Zuordnung anhand von Tags erstellen und von einer dieser Instances die Tags entfernen, wird die Zuordnung auf dieser Instance nicht mehr ausgeführt. Die Zuordnung zwischen der Instance und dem State Manager-Dokument ist aufgehoben.

Fehlerbehebung bei Automatisierungen, die von State Manager-Zuordnungen ausgeführt werden

Systems Manager setzt ein Limit von 100 gleichzeitigen Automatisierungen und 1.000 Automatisierungen in der Warteschlange pro Konto und Region. Wenn eine State Manager-Zuordnung, die ein Runbook verwendet, den Status Failed (Fehlgeschlagen) und den detaillierten Status AutomationExecutionLimitExceeded anzeigt, hat die Automatisierung möglicherweise das Limit erreicht. Daher drosselt Systems Manager die Automatisierungen. Führen Sie folgende Schritte aus, um dieses Problem zu lösen:

- Verwenden Sie einen anderen Rate- oder Cron-Ausdruck für Ihre Zuordnung. Beispiel: Wenn die Zuordnung alle 30 Minuten ausgeführt werden soll, ändern Sie den Ausdruck so, dass er jede Stunde oder alle zwei Stunden ausgeführt wird.
- Löschen Sie vorhandene Automatisierungen mit dem Status Pending (Ausstehend). Durch Löschen dieser Automatisierungen bereinigen Sie die aktuelle Warteschlange.

Planen von Automatisierungen mit Wartungsfenstern

Sie starten eine Automatisierung, indem Sie ein Runbook als registrierte Aufgabe für ein Wartungsfenster konfigurieren. Durch die Registrierung des Runbooks als registrierte Aufgabe führt ein Wartungsfenster die Automatisierung während des geplanten Wartungszeitraums aus.

Angenommen, Sie erstellen beispielsweise ein Runbook mit dem Namen `CreateAMI`, welches eine Amazon Machine Image (AMI) von Instances erstellt, welche als Ziele für das Wartungsfenster registriert sind. Um ein `CreateAMI`-Runbook (und die entsprechende Automatisierung) als eine registrierte Aufgabe eines Wartungsfensters angeben zu können, müssen Sie zunächst ein Wartungsfenster erstellen und Ziele registrieren. Im Anschluss daran geben Sie mit den folgenden Schritten das Dokument `CreateAMI` als registrierte Aufgabe innerhalb des Wartungsfensters an. Wenn das Wartungsfenster während des geplanten Zeitraums gestartet wird, führt das System die Automatisierung aus und erstellt ein der AMI registrierten Ziele.

Weitere Informationen zum Erstellen eines Automation-Runbooks finden Sie unter Erstellen Ihrer eigenen Runbooks. Automatisierung ist eine Fähigkeit von AWS Systems Manager.

Gehen Sie wie folgt vor, um eine Automatisierung mithilfe der AWS Systems Manager Konsole AWS Command Line Interface (AWS CLI) oder als registrierte Aufgabe für ein Wartungsfenster zu konfigurieren AWS Tools for Windows PowerShell.

Registrieren einer Automatisierungsaufgabe für ein Wartungsfenster (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine Automatisierung als registrierte Aufgabe für ein Wartungsfenster konfigurieren.

Bevor Sie beginnen

Bevor Sie die folgenden Schritte ausführen, müssen Sie ein Wartungsfenster erstellen und mindestens ein Ziel registrieren. Weitere Informationen finden Sie in den folgenden Verfahren:

- Erstellen eines Wartungsfensters (Konsole).
- Zuweisen von Zielen zu einem Wartungsfenster (Konsole)

So konfigurieren Sie einen Automatisierung als registrierte Aufgabe für ein Wartungsfenster

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im linken Navigationsbereich Maintenance Windows und dann das Wartungsfenster aus, für das Sie eine Automation-Aufgabe registrieren möchten.
3. Wählen Sie Aktionen. Wählen Sie dann Register Automation task (Automation-Aufgabe registrieren) aus, um die gewünschte Automatisierung mithilfe eines Runbooks auf den Zielen auszuführen.
4. Geben Sie unter Name einen Namen für die Aufgabe ein.
5. Geben Sie im Feld Description (Beschreibung) eine Beschreibung ein.
6. Wählen Sie für Document (Dokument) das Runbook aus, das die auszuführende Aufgabe definiert.
7. Wählen Sie für Document version (Dokumentversion) die zu verwendende Runbook-Version aus.
8. Wählen Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe aus. 1 ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.
9. Geben Sie im Abschnitt Targets (Ziele) die Ziele an, auf denen Sie diesen Automation-Workflow ausführen möchten, wenn das von Ihnen gewählte Runbook eines ist, das Aufgaben auf Ressourcen aufführt. Hierzu können Sie entweder Tags angeben oder die Instances manuell auswählen.

   > **Note**
   >
   > Wenn Sie die Ressourcen über Eingabeparameter anstelle von Zielen übergeben möchten, müssen Sie kein Wartungsfensterziel angeben.
   > In vielen Fällen müssen Sie kein Ziel für eine Automation-Aufgabe explizit angeben. Angenommen, Sie erstellen beispielsweise eine Automation-Aufgabe, um eine Amazon Machine Image (AMI) für Linux mit dem `AWS-UpdateLinuxAmi`-Runbook zu aktualisieren. Wenn die Aufgabe ausgeführt wird, wird AMI mit den neuesten verfügbaren Linux-Verteilungspaketen und Amazon-Software aktualisiert. Neue Instances, die aus der AMI erstellt wurden, haben diese Updates bereits installiert. Da die ID des AMI in den Eingabeparametern für das Runbook angegeben ist, muss in der Wartungsfenster-Aufgabe kein Ziel erneut angegeben werden.

   Informationen zu Wartungsfenster-Tasks, für die keine Ziele erforderlich sind, finden Sie unter the section called “Wartungsfenster-Tasks ohne Ziele registrieren”.
10. (Optional) Für Rate control (Ratenregelung):

> **Note**
>
> Wenn die ausgeführte Aufgabe keine Ziele angibt, müssen Sie keine Ratensteuerungen angeben.

- Geben Sie für Concurrency (Gleichzeitigkeit) entweder eine Anzahl oder einen Prozentsatz der Ziele ein, auf denen die Automatisierung gleichzeitig ausgeführt wird.

  Wenn Sie Ziele anhand von Tag-Schlüssel-Wert-Paaren ausgewählt haben und nicht sicher sind, von wie vielen Zielen die ausgewählten Tags verwendet werden, sollten Sie die Anzahl der Automatisierungen, die gleichzeitig ausgeführt werden können, durch einen Prozentsatz begrenzen.

  Wenn das Wartungsfenster ausgeführt wird, wird pro Ziel eine neue Automatisierung eingeleitet. Es gibt ein Limit von 100 gleichzeitigen Automatisierungen pro. AWS-Konto Wenn Sie einen Gleichzeitigkeitswert über 100 angeben, werden alle gleichzeitigen Automatisierungen über die 100. hinaus automatisch zur Automatisierungswarteschlange hinzugefügt. Informationen finden Sie unter Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz.
- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung der Automatisierung auf anderen Zielen beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Zielen ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, führt Systems Manager keine Automatisierungen mehr aus, wenn der vierte Fehler empfangen wird. Von Zielen, auf denen die Automatisierung noch ausgeführt wird, werden unter Umständen ebenfalls Fehler gesendet.

11. Geben Sie im Abschnitt Input Parameters die Parameter für das Runbook an. Bei Runbooks werden die Werte vom System automatisch gefüllt. Sie können diese Werte beibehalten oder ersetzen.

> **Important**
>
> Für Runbooks können Sie optional eine Automatisierungsübernahmerolle angeben. Wenn Sie keine Rolle für diesen Parameter angeben, übernimmt die Automatisierung die Wartungsfenster-Servicerolle, die Sie in Schritt 11 gewählt haben. Daher müssen Sie sicherstellen, dass die von Ihnen gewählte Wartungsfenster-Servicerolle über die

entsprechenden AWS Identity and Access Management (IAM-) Berechtigungen verfügt, um die im Runbook definierten Aktionen auszuführen.
Beispiel: Die serviceverknüpfte Rolle für Systems Manager verfügt nicht über die IAM-Berechtigung `ec2:CreateSnapshot`, die zur Verwendung des Runbooks `AWS-CopySnapshot` benötigt wird. Hier müssen Sie entweder eine benutzerdefinierte Wartungsfenster-Servicerolle verwenden oder eine Automation-Übernahmerolle angeben, die über `ec2:CreateSnapshot`-Berechtigungen verfügt. Weitere Informationen finden Sie unter Einrichten der Automatisierung.

12. Wählen Sie im Bereich IAM service role (IAM-Servicerolle) eine Rolle aus, um Systems Manager Berechtigungen zum Starten der Automatisierung zu erteilen.

    Informationen zum Erstellen einer Servicerolle für Wartungsfenster-Aufgaben finden Sie unter Konfigurieren Sie mit der Konsole Berechtigungen für Wartungsfenster.

13. Wählen Sie Register Automation task (Automation-Aufgabe registrieren) aus.

Registrieren einer Automation-Aufgabe für ein Wartungsfenster (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie Sie die AWS CLI (unter Linux oder Windows) verwenden oder AWS Tools for PowerShell eine Automatisierung als registrierte Aufgabe für ein Wartungsfenster konfigurieren.

Bevor Sie beginnen

Bevor Sie die folgenden Schritte ausführen, müssen Sie ein Wartungsfenster erstellen und mindestens ein Ziel registrieren. Weitere Informationen finden Sie in den folgenden Verfahren:

- Schritt 1: Erstellen des Wartungsfensters (AWS CLI).
- Schritt 2: Registrieren eines Ziel-Knotens mit dem Wartungsfenster (AWS CLI)

So konfigurieren Sie einen Automatisierung als registrierte Aufgabe für ein Wartungsfenster

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS Tools for PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

2. Erstellen Sie einen Befehl, um eine Automatisierung als registrierte Aufgabe für ein Wartungsfenster zu konfigurieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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**
   >
   > Wenn Sie eine Automatisierung mithilfe von als registrierte Aufgabe konfigurieren AWS CLI, verwenden Sie den Parameter, um `--Task-Invocation-Parameters` Parameter anzugeben, die an eine Aufgabe übergeben werden, wenn sie ausgeführt wird. Verwenden Sie nicht den Parameter `--Task-Parameters`. Der Parameter `--Task-Parameters` ist veraltet.
   >
   > Bei Wartungsfensteraufgaben ohne festgelegtes Ziel können Sie keine Werte für `--max-errors` und `--max-concurrency` bereitstellen. Stattdessen fügt das System den Platzhalterwert 1 ein, der in der Antwort auf Befehle wie describe-maintenance-window-tasks und get-maintenance-window-task gemeldet wird. Diese Werte wirken sich nicht auf die Ausführung Ihrer Aufgabe aus und können ignoriert werden. Informationen zu Wartungsfenster-Tasks, für die keine Ziele erforderlich sind, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

   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**
>
> Wenn Sie eine Automatisierung mithilfe von als registrierte Aufgabe konfigurieren AWS CLI, verwenden Sie den Parameter, um `--task-invocation-parameters` Parameter anzugeben, die an eine Aufgabe übergeben werden, wenn sie ausgeführt wird. Verwenden Sie nicht den Parameter `--task-parameters`. Der Parameter `--task-parameters` ist veraltet.
>
> Bei Wartungsfensteraufgaben ohne festgelegtes Ziel können Sie keine Werte für `--max-errors` und `--max-concurrency` bereitstellen. Stattdessen fügt das System den Platzhalterwert 1 ein, der in der Antwort auf Befehle wie describe-maintenance-window-tasks und get-maintenance-window-task gemeldet wird. Diese Werte wirken sich nicht auf die Ausführung Ihrer Aufgabe aus und können ignoriert werden. Informationen zu Wartungsfenster-Tasks, für die keine Ziele erforderlich sind, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

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**
>
> Wenn Sie eine Automatisierung mithilfe von als registrierte Aufgabe konfigurieren AWS Tools for PowerShell, verwenden Sie den Parameter, um `-Automation_Parameter` Parameter anzugeben, die an eine Aufgabe übergeben werden, wenn die Aufgabe ausgeführt wird. Verwenden Sie nicht den Parameter `-TaskParameters`. Der Parameter `-TaskParameters` ist veraltet.
> Bei Wartungsfensteraufgaben ohne festgelegtes Ziel können Sie keine Werte für `-MaxError` und `-MaxConcurrency` bereitstellen. Stattdessen fügt das System den Platzhalterwert 1 ein, der in der Antwort auf Befehle wie `Get-SSMMaintenanceWindowTaskList` und `Get-SSMMaintenanceWindowTask` gemeldet wird. Diese Werte wirken sich nicht auf die Ausführung Ihrer Aufgabe aus und können ignoriert werden.
> Informationen zu Wartungsfenster-Tasks, für die keine Ziele erforderlich sind, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

Im folgenden Beispiel wird eine Automatisierung als registrierte Aufgabe für ein Wartungsfenster mit Priorität 1 konfiguriert. Es zeigt auch, dass die `--targets`, `--max-errors` und `--max-concurrency`- Optionen für eine ziellose Wartungsfensteraufgabe weggelassen werden. Der Automatisierung verwendet das Runbook `AWS-StartEC2Instance` und die angegebene Automation-Übernahmerolle, um EC2-Instances zu starten, die als Ziele für das Wartungsfenster registriert sind. Das Wartungsfenster führt die Automatisierung gleichzeitig auf maximal 5 Instances zu einem bestimmten Zeitpunkt aus. Die Ausführung dieser registrierten Aufgabe wird außerdem für ein bestimmtes Intervall auf weiteren Instances gestoppt, wenn die Fehlerzählung 1 überschreitet.

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
```

Der Befehl gibt Details für die neue registrierte Aufgabe zurück, die den folgenden ähneln.

Linux & macOS

```
{
"WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

Windows

```
{
"WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
```

```
}
```

PowerShell

```
4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

3. Um die registrierte Aufgabe anzuzeigen, führen Sie den folgenden Befehl aus. Ersetzen Sie *maintenance window ID* (ID des Wartungsfensters) mit Ihren eigenen Informationen.

   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
   ```

   Das System gibt unter anderem folgende Informationen zurück

   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
```

Systems Manager Automation Aktionen-Referenz

Diese Referenz beschreibt die Automation-Aktionen, die Sie in einem Runbook angeben können. Automation ist eine Funktion von AWS Systems Manager. Diese Aktionen können nicht in anderen Arten von Systems Manager (SSM)-Dokumenten verwendet werden. Weitere Informationen zu Plug-Ins für andere Arten von SSM-Dokumente finden Sie unter Referenz für Befehlsdokument-Plug-ins.

Die Systems Manager Automation führt Schritte aus, die in Automation-Runbooks definiert sind. Jeder Schritt ist einer bestimmten Aktion zugeordnet. Die Aktion bestimmt die Eingaben, das Verhalten und die Ausgaben des Schritts. Die Schritte sind im `mainSteps`-Bereich Ihres Runbooks definiert.

Sie müssen die Ausgaben einer Aktivität oder eines Schritts nicht angeben. Die Ausgaben werden im Voraus durch die dem Schritt zugeordnete Aktivität bestimmt. Wenn Sie Schritteingaben in Ihren Runbooks festlegen, können Sie auf mindestens eine Ausgabe aus einem früheren Schritt verweisen. Beispielsweise können Sie die Ausgabe von `aws:runInstances` für eine spätere `aws:runCommand`-Aktion verfügbar machen. Sie können auch auf Ausgaben aus früheren Schritten im Abschnitt `Output` des Runbooks verweisen.

> **Important**
>
> Wenn Sie einen automatisierten Workflow ausführen, der andere Services mithilfe einer AWS Identity and Access Management-(IAM)-Servicerolle aufruft, muss die Servicerolle mit der Berechtigung zum Aufrufen dieser Services konfiguriert sein. Diese Anforderung gilt für alle AWS Automation-Runbooks (`AWS-*`-Runbooks), wie zum Beispiel `AWS-ConfigureS3BucketLogging`, `AWS-CreateDynamoDBBackup` und `AWS-RestartEC2Instance`-Runbooks, um nur einige zu nennen. Diese Anforderung gilt auch für alle von Ihnen erstellten benutzerdefinierten Automation-Runbooks, die andere AWS-Services mithilfe von Aktionen aufrufen, die andere Services aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:createStack`- oder `aws:copyImage`-Aktionen verwenden, konfigurieren Sie die Dienstrolle mit der Berechtigung zum Aufrufen solcher Services. Sie können anderen AWS-Services Berechtigungen erteilen, indem Sie der Rolle eine eingebundene IAM-Richtlinie hinzufügen. Weitere Informationen finden Sie unter

(Optional) Fügen Sie eine Inline-Automatisierungsrichtlinie oder eine vom Kunden verwaltete Richtlinie hinzu, um andere aufzurufen AWS-Services.

Themen

- Von allen Aktionen gemeinsam genutzte Eigenschaften
- aws:approve - Unterbrechen einer Automatisierung zur manuellen Genehmigung
- aws:assertAwsResourceProperty - Geltendmachung eines AWS-Ressourcenstatus oder Ereignisstatus
- aws:branch - Ausführen bedingter Automatisierungsschritte
- aws:changeInstanceState – Instance-Status ändern oder geltend machen
- aws:copyImage - Kopieren oder Verschlüsseln eines Amazon Machine Image
- aws:createImage - Erstellen eines Amazon Machine Image
- aws:createStack— Erstelle einen AWS CloudFormation Stapel
- aws:createTags - Erstellen von Tags für AWS-Ressourcen
- aws:deleteImage - Löschen eines Amazon Machine Image
- aws:deleteStack – Löschen Sie ein AWS CloudFormation-Stack
- aws:executeAutomation - Führen Sie eine weitere Automatisierung durch
- aws:executeAwsApi – Aufrufen und Ausführen von AWS API-Operationen
- aws:executeScript - Führen Sie ein Skript aus
- aws:executeStateMachine – Führen Sie eine AWS Step Functions-State Machine aus.
- aws:invokeWebhook – Automation-Webhook-Integration aufrufen
- aws:invokeLambdaFunction – Aufrufen einer AWS Lambda-Funktion
- aws:loop – Über Schritte in einer Automatisierung iterieren
- aws:pause - Pausieren einer Automatisierung
- aws:runCommand - Führt einen Befehl auf einer verwalteten Instance aus
- aws:runInstances - So starten Sie eine Amazon-EC2-Instance
- aws:sleep - Verzögerung einer Automatisierung
- aws:updateVariable – Aktualisiert einen Wert für eine Runbook-Variable
- aws:waitForAwsResourceProperty - Warten Sie auf eine AWS-Ressourceneigenschaft
- Systemvariablen für Automation

Von allen Aktionen gemeinsam genutzte Eigenschaften

Allgemeine Eigenschaften sind Parameter oder Optionen, die in allen Aktionen gefunden werden. Einige Optionen definieren das Verhalten für einen Schritt, etwa wie lange auf den Abschluss eines Schritts gewartet werden muss und was zu tun ist, wenn der Schritt fehlschlägt. Die folgenden Eigenschaften sind allen Aktionen gemeinsam.

description

Informationen, die Sie angeben, um den Zweck eines Runbooks oder eines Schritts zu beschreiben.

Typ: Zeichenfolge

Required: No

name

Ein Bezeichner, der für alle Schrittnamen im Runbook eindeutig sein muss.

Typ: Zeichenfolge

Zulässiges Muster: [a-zA-Z0-9_]+$

Erforderlich: Ja

action

Der Name der Aktion, die der Schritt ausführt. `aws:runCommand` - Führt einen Befehl auf einer verwalteten Instance aus ist ein Beispiel für eine Aktion, die Sie hier angeben können. Dieses Dokument enthält detaillierte Informationen über alle verfügbaren Aktionen.

Typ: Zeichenfolge

Erforderlich: Ja

maxAttempts

Die Anzahl der Wiederholungen des Schritt bei einem Fehler. Wenn der Wert größer als 1 ist, wird der Schritt erst als fehlgeschlagen betrachtet, wenn alle Wiederholungsversuche fehlgeschlagen sind. Der Standardwert lautet 1.

Typ: Ganzzahl

Required: No

timeoutSeconds

Der Wert für das Timeout des Schritts. Wenn das Timeout erreicht ist und der Wert von `maxAttempts` größer als 1 ist, wird der Schritt erst als abgelaufen betrachtet, wenn alle Wiederholungen durchgeführt wurden.

Typ: Ganzzahl

Required: No

onFailure

Gibt an, ob die Automatisierung bei einem Fehler abgebrochen, fortgesetzt oder bis zu einem bestimmten Schritt übersprungen werden soll. Der Standardwert für diese Option ist "abort".

Typ: Zeichenfolge

Gültige Werte: Abort | Continue | step:*Schritt-Name*

Required: No

onCancel

Gibt an, zu welchem Schritt die Automatisierung gehen soll, falls ein Benutzer die Automatisierung abbricht. Die Automatisierung führt den Stornierungs-Workflow für maximal zwei Minuten aus.

Typ: Zeichenfolge

Gültige Werte: Abort | Continue | step:*step_name*

Required: No

Die `onCancel`-Eigenschaft unterstützt das Verschieben zu den folgenden Aktionen nicht:

- `aws:approve`
- `aws:copyImage`
- `aws:createImage`
- `aws:createStack`
- `aws:createTags`
- `aws:loop`
- `aws:pause`

- `aws:runInstances`
- `aws:sleep`

isEnd

Diese Option stoppt eine Automatisierung am Ende eines bestimmten Schrittes. Die Automatisierung stoppt, egal ob der Schritt erfolgreich oder gar nicht ausgeführt werden konnte. Der Standardwert von "false".

Typ: Boolesch

Zulässige Werte: true | false

Required: No

nextStep

Gibt an, welcher Schritt in einer Automatisierung nach dem erfolgreichem Abschluss eines Schritts als nächster auszuführen ist.

Typ: Zeichenfolge

Required: No

isCritical

Bezeichnet einen Schritt als kritisch für den erfolgreichen Abschluss der Automation. Wenn ein Schritt mit dieser Bezeichnung fehlschlägt, dann wird der endgültige Status der Automation als fehlgeschlagen gemeldet. Diese Eigenschaft wird nur ausgewertet, wenn Sie diese explizit in Ihrem Schritt definieren. Wenn die `onFailure`-Eigenschaft auf `Continue` in einem Schritt gesetzt ist, lautet der Standardwert „false“. Der Standardwert für diese Option ist sonst „true“.

Typ: Boolesch

Zulässige Werte: true | false

Required: No

inputs

Die für die Aktivität spezifischen Eigenschaften.

Typ: Zuordnung

Erforderlich: Ja

Beispiel

```
---
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` - Unterbrechen einer Automatisierung zur manuellen Genehmigung

Hält eine Automatisierung zeitweise an, bis die Aktion von designierten Prinzipalen genehmigt oder abgelehnt wird. Nach Erreichen der erforderlichen Anzahl an Genehmigungen wird die Automatisierung fortgesetzt. Sie können den Genehmigungsschritt an jeder beliebigen Stelle im `mainSteps`-Bereich Ihres Runbooks ansetzen.

> **Note**
>
> Diese Aktion unterstützt keine Automatisierungen für mehrere Konten und Regionen. Das Standard-Timeout für diese Aktion beträgt 7 Tage (604 800 Sekunden) und der Höchstwert ist 30 Tage (2 592 000 Sekunden). Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:approve`-Schritt anpassen. Wenn der

Automatisierungsschritt den Zeitüberschreitungswert erreicht, bevor alle notwendigen Genehmigungsentscheidungen getroffen wurden, werden der Schritt und die gesamte Automatisierung gestoppt und der Status „Timed Out“ zurückgegeben.

Im folgenden Beispiel hält die Aktion `aws:approve` die Automatisierung vorübergehend an, bis ein Genehmiger die Automatisierung entweder akzeptiert oder ablehnt. Nach der Genehmigung führt die Automatisierung einen einfachen PowerShell Befehl aus.

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

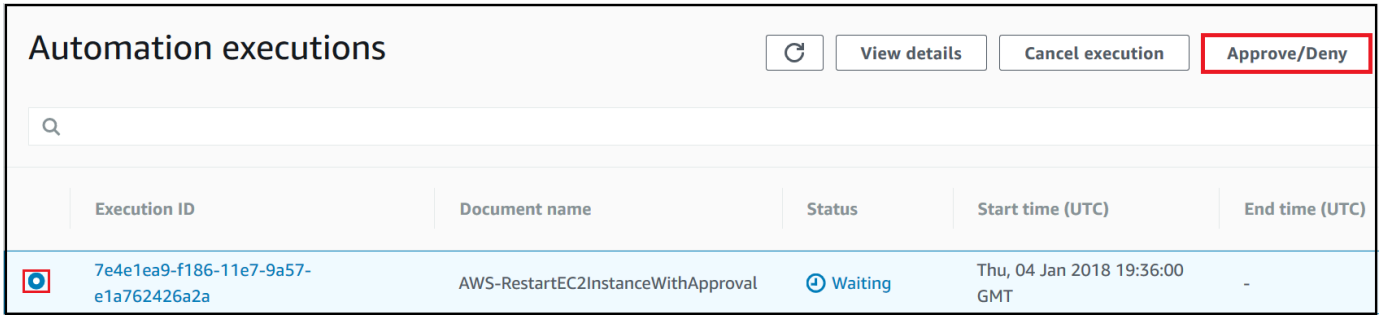
```
{
   "description":"RunInstancesDemo1",
   "schemaVersion":"0.3",
   "assumeRole":"{{ assumeRole }}",
   "parameters":{
      "assumeRole":{
         "type":"String"
      },
      "message":{
         "type":"String"
      }
   },
   "mainSteps":[
      {
         "name":"approve",
         "action":"aws:approve",
         "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"
               ]
            }
         }
```

```
        }
    ]
}
```

Sie können Automatisierungen, die in der Konsole noch nicht genehmigt wurden, genehmigen oder ablehnen.

So genehmigen Sie Automatisierungen oder lehnen sie ab

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Option neben einer Automation mit dem Status Waiting (Warten).



4. Wählen Sie Approve/Deny aus.
5. Überprüfen Sie die Details der Automation.
6. Wählen Sie Approve (Genehmigen) oder Deny (Verweigern), geben Sie einen optionalen Kommentar ein und wählen Sie dann Submit (Absenden) aus.

Eingabebeispiel

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

Der Amazon Resource Name (ARN) eines Amazon Simple Notification Service (Amazon SNS) Themas für Automation-Genehmigungen. Wenn Sie einen `aws:approve`-Schritt in einer Automatisierung festlegen, sendet Automation eine Nachricht an dieses Thema und informiert die Prinzipale darüber, dass sie einen Automation-Schritt entweder genehmigen oder zurückweisen müssen. Die Bezeichnung des Amazon-SNS-Themas muss das Präfix „Automatisierung“ aufweisen.

Typ: Zeichenfolge

Erforderlich: Nein

Fehlermeldung

Die Informationen, die Sie in das Amazon-SNS-Thema einbeziehen möchten, wenn die Genehmigungsanforderung gesendet wird. Die maximale Länge der Nachricht beträgt 4096 Zeichen.

Typ: Zeichenfolge

Erforderlich: Nein

MinRequiredApprovals

Die erforderliche Mindestanzahl an Genehmigungen zum Fortsetzen der Automatisierung. Wenn Sie keinen Wert angeben, verwendet das System standardmäßig den Wert 1. Der Wert für diesen

Parameter muss eine positive Zahl sein. Der Wert für diesen Parameter darf nicht größer sein als die Anzahl der Genehmiger, die anhand des `Approvers`-Parameters definiert sind.

Typ: Ganzzahl

Erforderlich: Nein

Genehmiger

Eine Liste AWS authentifizierter Principals, die die Aktion entweder genehmigen oder ablehnen können. Die maximale Anzahl an Genehmigern ist 10. Sie können Prinzipale anhand eines der folgenden Formate festlegen:

- Ein Benutzername
- Ein Benutzer-ARN
- Ein IAM-Rollen-ARN
- Ein IAM-Rollenübernahme-ARN

Typ: StringList

Erforderlich: Ja

EnhancedApprovals

Diese Eingabe wird nur für Change Manager Vorlagen verwendet. Eine Liste der AWS authentifizierten Principals, die die Aktion entweder genehmigen oder ablehnen können, den Typ des IAM-Prinzipals und die Mindestanzahl von Genehmiger. Im Folgenden wird ein Beispiel gezeigt:

```
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
```

Typ: StringList

Erforderlich: Ja

Ausgabe

ApprovalStatus

Der Genehmigungsstatus des Schritts. Der Status kann einer der folgenden sein: Genehmigt, Abgelehnt oder Warten. Warten bedeutet, dass Automation auf eine Eingabe der Genehmiger wartet.

Typ: Zeichenfolge

ApproverDecisions

Eine JSON-Karte, die den Genehmigungsbescheid der einzelnen Genehmiger enthält.

Typ: MapList

`aws:assertAwsResourceProperty` - Geltendmachung eines AWS-Ressourcenstatus oder Ereignisstatus

Die Aktion `aws:assertAwsResourceProperty` erlaubt Ihnen, einen bestimmten Ressourcen- oder Ereignisstatus für einen bestimmten Automation-Schritt zu prüfen. Sie können beispielsweise angeben, dass ein Automation-Schritt darauf wartet, dass eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance gestartet wird. Dann ruft es den Amazon EC2 DescribeInstanceStatus-API-Vorgang mit der DesiredValue Eigenschaft von `running` auf. Auf diese Weise wird sichergestellt,

dass der Automatisierung auf eine laufende Instance wartet und fortfährt, wenn diese Instance tatsächlich ausgeführt wird.

Weitere Beispiele zur Verwendung dieser Aktion finden Sie unter Weitere Runbook-Beispiele.

Eingabe

Eingaben werden von der ausgewählten API-Operation bestimmt.

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"
    ]
  }
}
```

Service

Der AWS-Service-Namespace, der die API-Operation enthält, die Sie ausführen möchten. Beispielsweise lautet der Namespace für Systems Manager `ssm`. Der Namespace für Amazon EC2 lautet `ec2`. Sie finden eine Liste der unterstützten AWS-Service-Namespaces im Abschnitt Verfügbare Services der AWS CLI-Befehlsreferenz.

Typ: Zeichenfolge

Erforderlich: Ja

Api

Der Name der API-Operation, die Sie ausführen möchten. Sie können die API-Operationen (auch als Methoden bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise werden alle API-Vorgänge (Methoden) für Amazon Relational Database Service (Amazon RDS) auf der folgenden Seite aufgelistet: Amazon RDS-Methoden.

Typ: Zeichenfolge

Erforderlich: Ja

API-Operation-Eingaben

Eine oder mehrere API-Eingaben. Sie können die verfügbaren Eingaben (auch als Parameter bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie abwärts, um die verfügbaren Parameter zu sehen, wie etwa DBInstanceIdentifier, Name und Values (Werte). Verwenden Sie das folgende Format, um mehr als eine Eingabe anzugeben.

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"
}
```

Typ: Abhängig von der gewählten API-Operation

Erforderlich: Ja

PropertySelector

Der JSONPath zu einem bestimmten Attribut im Antwortobjekt. Sie können die Antwortobjekte anzeigen indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie abwärts zum Abschnitt Response Structure (Antwortstruktur). DBInstances wird als Antwortobjekt aufgeführt.

Typ: Zeichenfolge

Erforderlich: Ja

DesiredValues

Die erwartete Status oder Zustand, bei dem die Automatisierung fortgesetzt werden soll. Wenn Sie einen booleschen Wert angeben, müssen Sie einen Großbuchstaben verwenden, wie z. B. True oder False.

Typ: StringList

Erforderlich: Ja

`aws:branch` - Ausführen bedingter Automatisierungsschritte

Die Aktion `aws:branch` erlaubt das Erstellen einer dynamischen Automatisierung, der verschiedene Auswahlmöglichkeiten in einem einzigen Schritt evaluiert und dann auf der Grundlage dieser Evaluierung zu einem anderen Schritt in dem Runbook springt.

Wenn Sie die Aktion `aws:branch` für einen Schritt angeben, geben Sie die `Choices` an, die die Automatisierung evaluieren muss. Die `Choices` können auf einem Wert basieren, den Sie im Abschnitt `Parameters` des Runbooks angegeben haben, oder auf einem als Ausgabe von

dem vorherigen Schritt generierten dynamischen Wert basieren. Die Automatisierung evaluiert jede Auswahl mithilfe eines booleschen Ausdrucks. Wenn die erste Auswahl „wahr“ ist, springt die Automatisierung zu dem für diese Auswahl vorgesehenen Schritt. Wenn die erste Auswahl „false“ ist, evaluiert die Automatisierung die nächste Auswahl. Die Automatisierung evaluiert weiterhin jede Auswahl, bis eine Auswahl als „true“ verarbeitet wird. Die Automatisierung springt dann zu dem für die als „true“ evaluierte Auswahl angegebenen Schritt.

Wenn keine Auswahl als „true“ evaluiert wird, prüft die Automatisierung, ob der Schritt einen `default`-Wert enthält. Ein Default-Wert definiert einen Schritt, zu dem die Automatisierung springen soll, wenn keine der Auswahlmöglichkeiten als „true“ evaluiert wird. Wenn kein `default`-Wert für den Schritt definiert ist, verarbeitet die Automatisierung den nächsten Schritt in dem Runbook.

Die Aktion `aws:branch` unterstützt komplexe Auswahlevaluierungen durch Verwendung einer Kombination der Operatoren `And`, `Not` und `Or`. Weitere Informationen über die Verwendung von `aws:branch`, mit Beispielrunbooks und Beispielen, die unterschiedliche Operatoren verwenden, finden Sie unter Verwendung bedingter Anweisungen in Runbooks.

Eingabe

Geben Sie eine oder mehrere `Choices` in einem Schritt an. Die `Choices` können auf einem Wert basieren, den Sie im Abschnitt `Parameters` des Runbooks angegeben haben, oder auf einem als Ausgabe von dem vorherigen Schritt generierten dynamischen Wert basieren. Hier ist ein YAML-Beispiel, das einen Parameter evaluiert.

```
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
```

Hier ist ein YAML-Beispiel, das die Ausgabe von einem vorherigen Schritt evaluiert.

```
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
```

Auswahlen

Ein oder mehrere Ausdrücke, die die Automatisierung evaluieren soll, wenn der nächste zu verarbeitende Schritt bestimmt wird. Auswahlen werden mit einem booleschen Ausdruck evaluiert. Jede Auswahl muss die folgenden Optionen definieren:

- NextStep: Der nächste Schritt in dem Runbook, der zu verarbeiten ist, wenn die betreffende Auswahl „true“ ist.
- Variable: Geben Sie entweder den Namen eines Parameters an, der im Abschnitt `Parameters` des Runbooks definiert ist, Oder geben Sie ein Ausgabeobjekt von einem vorherigen Schritt im Runbook an. Weitere Informationen zum Erstellen von Variablen für `aws:branch` finden Sie unter Informationen zum Erstellen der Ausgabevariable.
- Operation: Die Kriterien für die Evaluierung der Auswahl. Die Aktion `aws:branch` unterstützt die folgenden Operationen:

  Zeichenfolgenoperationen

  - StringEquals
  - EqualsIgnoreCase
  - StartsWith
  - EndsWith
  - Enthält

  Numerische Operationen

  - NumericEquals

- NumericGreater
- NumericLesser
- NumericGreaterOrEquals
- NumericLesser
- NumericLesserOrEquals

Boolesche Operation

- BooleanEquals

> **Important**
>
> Wenn Sie ein Runbook erstellen, validiert das System alle Operationen im Runbook. Wenn eine Operation nicht unterstützt wird, gibt das System einen Fehler aus, wenn Sie versuchen, das Runbook zu erstellen.

Standard

Der Name eines Schritts, zu dem die Automatisierung springen soll, wenn keine der `Choices` „true“ ist.

Typ: Zeichenfolge

Required: No

> **Note**
>
> Die Aktion `aws:branch` unterstützt die Operatoren `And`, `Or` und `Not`. Beispiele für `aws:branch` unter Verwendung von Operatoren finden Sie unter Verwendung bedingter Anweisungen in Runbooks.

`aws:changeInstanceState` – Instance-Status ändern oder geltend machen

Ändert oder klärt den Status der Instance.

Diese Aktivität kann im Assert-Modus verwendet werden (führt jedoch die API nicht aus, um den Status zu ändern, sondern prüft, ob die Instance den gewünschten Status aufweist.) Um den Assert-Modus zu verwenden, setzen Sie den Parameter `CheckStateOnly` auf "true". Dieser Modus ist

nützlich, wenn der Sysprep-Befehl unter Windows ausgeführt wird. Bei diesem Befehl handelt es sich um einen asynchronen Befehl, der lange Zeit im Hintergrund ausgeführt werden kann. Sie können sicherstellen, dass die Instance angehalten wird, bevor Sie ein Amazon Machine Image (AMI) erstellen.

> **Note**
>
> Der Standardwert für die Zeitüberschreitung für diese Aktion beträgt 3 600 Sekunden (eine Stunde). Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:changeInstanceState`-Schritt anpassen.

Eingabe

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

Die IDs der Instances.

Typ: StringList

Erforderlich: Ja

CheckStateOnly

Wenn „false“, wird der Instance-Status auf den gewünschten Status festgelegt. Wenn „true“, wird der gewünschte Status anhand einer Abfrage überprüft.

Standard: `false`

Typ: Boolesch

Required: No

DesiredState

Der gewünschte Status. Bei der Einstellung `running` wartet diese Aktion auf den Amazon EC2 Status `Running`, den Instance-Status `OK` und den Systemstatus `OK` vor dem Abschluss.

Typ: Zeichenfolge

Zulässige Werte: `running` | `stopped` | `terminated`

Erforderlich: Ja

Force

Wenn festgelegt, wird das Anhalten der Instances erzwungen. Die Instances haben keine Gelegenheit, die Caches oder Metadaten des Dateisystems zu leeren. Wenn Sie diese Option verwenden, müssen Sie eine Überprüfung und Reparatur des Dateisystems durchführen. Diese Option wird für EC2-Instances für Windows Server nicht empfohlen.

Typ: Boolesch

Required: No

AdditionalInfo

Reserved Instances.

Typ: Zeichenfolge

Required: No

Ausgabe

Keine

`aws:copyImage` - Kopieren oder Verschlüsseln eines Amazon Machine Image

Kopiert ein Amazon Machine Image (AMI) aus einer beliebigen AWS-Region in die aktuelle Region. Diese Aktion kann auch das neue AMI verschlüsseln.

Eingabe

Diese Aktion unterstützt die meisten `CopyImage`-Parameter. Weitere Informationen hierzu finden Sie unter CopyImage.

Das folgende Beispiel erstellt eine Kopie eines AMI in der Region Seoul (`SourceImageID`: ami-0fe10819. `SourceRegion`: ap-northeast-2). Das neue AMI wird in die Region kopiert, in der Sie die Automation-Aktivität gestartet haben. Das kopierte AMI wird verschlüsselt, da das optionale `Encrypted`-Flag auf `true` gesetzt ist.

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

Die Region, in der Quell-AMI derzeit vorhanden ist.

Typ: Zeichenfolge

Erforderlich: Ja

SourceImageId

Die AMI-ID, die aus der Quellregion kopiert werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

ImageName

Der Name für das neue Image.

Typ: Zeichenfolge

Erforderlich: Ja

ImageDescription

Eine Beschreibung des Ziel-Image.

Typ: Zeichenfolge

Required: No

Encrypted

Verschlüsseln Sie das AMI.

Typ: Boolesch

Required: No

KmsKeyId

Der vollständige Amazon-Ressourcenname (ARN) der AWS KMS key für die Verschlüsselung der Snapshots eines Image während eines Kopiervorgangs. Weitere Informationen hierzu finden Sie unter CopyImage.

Typ: Zeichenfolge

Required: No

ClientToken

Ein eindeutiger Bezeichner, bei dem die Groß- und Kleinschreibung beachtet werden muss, um die Idempotenz der Anforderung sicherzustellen. Weitere Informationen hierzu finden Sie unter CopyImage.

Typ: Zeichenfolge

Required: No

Ausgabe

ImageId

Die ID des kopierten Image.

ImageState

Der Status des kopierten Image.

Zulässige Werte: `available` | `pending` | `failed`

`aws:createImage` - Erstellen eines Amazon Machine Image

Erstellt ein Amazon Machine Image (AMI) aus einer Instance, die entweder derzeit ausgeführt wird, angehalten wird oder angehalten wurde.

Eingabe

Diese Aktion unterstützt die folgenden `CreateImage`-Parameter. Weitere Informationen hierzu finden Sie unter 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

Die ID der Instance.

Typ: Zeichenfolge

Erforderlich: Ja

ImageName

Der Name für das Image.

Typ: Zeichenfolge

Erforderlich: Ja

ImageDescription

Eine Beschreibung des Image.

Typ: Zeichenfolge

Required: No

NoReboot

Ein boolesches Literal.

Standardmäßig versucht Amazon Elastic Compute Cloud (Amazon EC2) vor dem Erstellen des Images, die Instance herunterzufahren und neu zu starten. Wenn die Option No Reboot (Kein Neustart) auf `true` eingestellt ist, fährt Amazon EC2 die Instance vor dem Erstellen des Images nicht herunter. Wenn diese Option verwendet wird, kann die Integrität des Dateisystems auf dem erstellten Image nicht garantiert werden.

Wenn Sie nicht möchten, dass die Instance ausgeführt wird, nachdem Sie ein AMIdavon erstellt haben, halten Sie die Instance zunächst mit der `aws:changeInstanceState` – Instance-Status ändern oder geltend machen-Aktion an und verwenden Sie diese `aws:createImage`-Aktion dann mit der Option NoReboot auf `true`.

Typ: Boolesch

Required: No

BlockDeviceMappings

Die Blockgeräte für die Instance.

Typ: Zuordnung

Required: No

Ausgabe

ImageId

Die ID des neu erstellten Image.

Typ: Zeichenfolge

ImageState

Der aktuelle Status des Image. Wenn der Status verfügbar ist, wird das Image erfolgreich registriert und kann zum Starten einer Instance verwendet werden.

Typ: Zeichenfolge

aws:createStack— Erstelle einen AWS CloudFormation Stapel

Erzeugt einen AWS CloudFormation Stapel aus einer Vorlage.

Zusätzliche Informationen zum Erstellen von CloudFormation Stacks finden Sie CreateStackin der AWS CloudFormation API-Referenz.

Eingabe

YAML

```
name: makeStack
action: aws:createStack
maxAttempts: 1
onFailure: Abort
inputs:
  Capabilities:
  - CAPABILITY_IAM
  StackName: myStack
  TemplateURL: http://s3.amazonaws.com/DOC-EXAMPLE-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/DOC-EXAMPLE-BUCKET/myStackTemplate",
        "TimeoutInMinutes": 5,
        "Parameters": [
          {
            "ParameterKey": "LambdaRoleArn",
            "ParameterValue": "{{LambdaAssumeRole}}"
          },
          {
            "ParameterKey": "createdResource",
            "ParameterValue": "createdResource-{{automation:EXECUTION_ID}}"
          }
    }
}
```

Funktionen

Mit einer Liste von Werten, die Sie zuvor angegeben haben, CloudFormation können Sie bestimmte Stacks erstellen. Einige Stack-Vorlagen enthalten Ressourcen, die sich auf Ihre AWS-Konto Berechtigungen auswirken können. Für einige Stacks müssen Sie deren Fähigkeiten mithilfe dieses Parameters explizit bestätigen.

Gültige Werte sind: `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM` und `CAPABILITY_AUTO_EXPAND`.

CAPABILITY_IAM und CAPABILITY_NAMED_IAM

Wenn Sie &IAM;-Ressourcen besitzen, können Sie jede Fähigkeit angeben.
Wenn Sie IAM-Ressourcen mit benutzerdefinierten Namen besitzen, müssen Sie `CAPABILITY_NAMED_IAM` angeben. Wenn Sie diesen Parameter angeben, gibt die Aktivität einen `InsufficientCapabilities`-Fehler zurück. Für die folgenden Ressourcen müssen Sie entweder `CAPABILITY_IAM` oder `CAPABILITY_NAMED_IAM` angeben.

- `AWS::IAM::AccessKey`
- `AWS::IAM::Group`
- `AWS::IAM::InstanceProfile`
- `AWS::IAM::Policy`

- AWS::IAM::Role
- AWS::IAM::User
- AWS::IAM::UserToGroupAddition

Wenn Ihre Stack-Vorlage diese Ressourcen enthält, empfehlen wir, dass Sie alle ihnen zugeordneten Berechtigungen überprüfen und ihre Berechtigungen bei Bedarf bearbeiten.

Weitere Informationen finden Sie unter Bestätigung von IAM-Ressourcen in AWS CloudFormation Vorlagen.

CAPABILITY_AUTO_EXPAND

Einige Vorlagen enthalten Makros. Makros führen eine benutzerdefinierte Verarbeitung von Vorlagen durch. Dies kann einfache Aktionen wie find-and-replace Operationen bis hin zu umfangreichen Transformationen ganzer Vorlagen umfassen. Aus diesem Grund erstellt der Benutzer normalerweise einen Änderungssatz aus der verarbeiteten Vorlage, sodass er die aus den Makros resultierenden Änderungen überprüfen kann, bevor er den Stack tatsächlich erstellt. Wenn Ihre Stack-Vorlage ein oder mehrere Makros enthält und Sie sich dafür entscheiden, einen Stack direkt aus der verarbeiteten Vorlage zu erstellen, ohne vorher die resultierenden Änderungen in einem Änderungssatz zu überprüfen, müssen Sie diese Funktion berücksichtigen.

Weitere Informationen finden Sie im Benutzerhandbuch unter Verwenden von AWS CloudFormation Makros zur benutzerdefinierten Verarbeitung von Vorlagen.AWS CloudFormation

Typ: Zeichenfolge-Array

Zulässige Werte: `CAPABILITY_IAM | CAPABILITY_NAMED_IAM | CAPABILITY_AUTO_EXPAND`

Erforderlich: Nein

ClientRequestToken

Eine eindeutige Kennung für diese CreateStack Anfrage. Geben Sie dieses Token an, wenn Sie maxAttempts in diesem Schritt auf einen Wert größer als 1 festlegen. Durch die Angabe dieses Tokens CloudFormation weiß, dass Sie nicht versuchen, einen neuen Stack mit demselben Namen zu erstellen.

Typ: Zeichenfolge

Erforderlich: Nein

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Muster: [a-zA-Z0-9][-a-zA-Z0-9]*

DisableRollback

Legen Sie den Parameter auf `true` fest, um ein Rollback des Stacks zu deaktivieren, wenn ein Fehler bei der Erstellung des Stacks aufgetreten ist.

Bedingt: Sie können entweder den `DisableRollback`-Parameter oder den `OnFailure`-Parameter festlegen, aber nicht beide.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

NotificationARNs

Die Amazon Simple Notification Service (Amazon SNS)-Thema-ARNs zum Veröffentlichen von Stack-bezogenen Ereignissen. Sie finden SNS-Thema-ARNs mithilfe der Amazon SNS Konsole, https://console.aws.amazon.com/sns/v3/home.

Typ: Zeichenfolge-Array

Array-Mitglieder: Maximale Anzahl von 5 Elementen.

Erforderlich: Nein

OnFailure

Bestimmt die Aktion, die ergriffen werden muss, wenn ein Fehler am Stack auftritt. Sie müssen `DO_NOTHING`, `ROLLBACK` oder `DELETE` angeben.

Bedingt: Sie können entweder den `OnFailure`-Parameter oder den `DisableRollback`-Parameter festlegen, aber nicht beide.

Standard: `ROLLBACK`

Typ: Zeichenfolge

Zulässige Werte: `DO_NOTHING | ROLLBACK | DELETE`

Erforderlich: Nein

Parameter

Eine Liste der `Parameter`-Strukturen, die Eingabeparameter für den Stack angeben. Weitere Informationen finden Sie im Datentyp Parameter.

Typ: Array von Parameter-Objekten

Erforderlich: Nein

ResourceTypes

Die Vorlagenressourcentypen für diese Aktion zum Erstellen von Stacks, für die Sie über Berechtigungen verfügen. Beispiel: `AWS::EC2::Instance`, `AWS::EC2::*` oder `Custom::MyCustomInstance`. Verwenden Sie die folgende Syntax zum Beschreiben von Vorlagenressourcentypen.

- Für alle AWS Ressourcen:

```
AWS::*
```

- Für alle benutzerdefinierten Ressourcen:

```
Custom::*
```

- Für eine bestimmte benutzerdefinierte Ressource:

```
Custom::logical_ID
```

- Für alle Ressourcen eines bestimmten AWS-Service:

```
AWS::service_name::*
```

- Für eine bestimmte AWS Ressource:

```
AWS::service_name::resource_logical_ID
```

Wenn die Liste der Ressourcentypen keine Ressource enthält, die Sie erstellen, schlägt die Erstellung des Stacks fehl. CloudFormation Gewährt standardmäßig Berechtigungen für alle Ressourcentypen. IAM verwendet diesen Parameter für CloudFormation -spezifische Bedingungsschlüssel in IAM-Richtlinien. Weitere Informationen finden Sie unter Zugriffskontrolle mit. AWS Identity and Access Management

Typ: Zeichenfolge-Array

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

RoleARN

Der Amazon-Ressourcenname (ARN) einer IAM-Rolle, die CloudFormation davon ausgeht, den Stack zu erstellen. CloudFormation verwendet die Anmeldeinformationen der Rolle, um in Ihrem Namen Anrufe zu tätigen. CloudFormation verwendet diese Rolle immer für alle future Operationen auf dem Stack. Solange Benutzer berechtigt sind, auf dem Stack zu arbeiten, CloudFormation verwendet diese Rolle auch dann, wenn die Benutzer nicht berechtigt sind, sie weiterzugeben. Stellen Sie sicher, dass die Rolle die geringstmögliche Menge an Berechtigungen gewährt.

Wenn Sie keinen Wert angeben, CloudFormation verwendet die Rolle, die zuvor dem Stack zugeordnet war. Wenn keine Rolle verfügbar ist, CloudFormation verwendet eine temporäre Sitzung, die anhand Ihrer Benutzeranmeldedaten generiert wird.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 20. Maximale Länge beträgt 2048 Zeichen.

Erforderlich: Nein

StackName

Der dem Stack zugeordnete Name. Der Name muss in der Region eindeutig sein, in der Sie den Stack erstellen.

> **Note**
>
> Ein Stack-Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Er muss mit einem alphabetischen Zeichen beginnen und darf nicht mehr als 128 Zeichen umfassen.

Typ: Zeichenfolge

Erforderlich: Ja

StackPolicyKörper

Struktur, die die Stack-Richtlinie enthält. Weitere Informationen finden Sie unter Verhindern von Aktualisierungen der Stack-Ressourcen.

Bedingt: Sie können entweder den `StackPolicyBody`-Parameter oder den `StackPolicyURL`-Parameter festlegen, aber nicht beide.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 16384 Zeichen.

Erforderlich: Nein

StackPolicyURL

Speicherort einer Datei, die die Stack-Richtlinie enthält. Die URL muss auf eine Richtlinie in einem S3-Bucket in derselben Region wie der Stack verweisen. Die maximal zulässige Dateigröße für die Stack-Richtlinie ist 16 KB.

Bedingt: Sie können entweder den `StackPolicyBody`-Parameter oder den `StackPolicyURL`-Parameter festlegen, aber nicht beide.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 1350 Zeichen.

Erforderlich: Nein

Tags

Schlüssel-Wert-Paare, die diesem Stack zugeordnet werden sollen. CloudFormation überträgt diese Tags auch auf die im Stack erstellten Ressourcen. Sie können höchstens 10 Tags angeben.

Typ: Array von Tag-Objekten

Erforderlich: Nein

TemplateBody

Struktur, die den Vorlagetext mit einer Mindestlänge von 1 Byte und einer Höchstlänge von 51.200 Byte enthält. Weitere Informationen finden Sie unter Aufbau einer Vorlage.

Bedingt: Sie können entweder den `TemplateBody`-Parameter oder den `TemplateURL`-Parameter festlegen, aber nicht beide.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen.

Erforderlich: Nein

TemplateURL

Speicherort einer Datei, die den Vorlagentext enthält. Die URL muss auf eine Vorlage verweisen, die sich in einem S3-Bucket befindet. Die maximal zulässige Größe für die Vorlage ist 460.800 Byte. Weitere Informationen finden Sie unter Aufbau einer Vorlage.

Bedingt: Sie können entweder den `TemplateBody`-Parameter oder den `TemplateURL`-Parameter festlegen, aber nicht beide.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Erforderlich: Nein

TimeoutInMinuten

Die Zeit, die verstreichen kann, bevor der Stack-Status zu `CREATE_FAILED` wird. Falls `DisableRollback` nicht festgelegt ist oder auf `false` festgelegt ist, wird für den Stack ein Rollback ausgeführt.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1.

Erforderlich: Nein

Outputs

StackId

Eindeutiger Bezeichner des Stacks.

Typ: Zeichenfolge

StackStatus

Aktueller Status des Stacks.

Typ: Zeichenfolge

Zulässige Werte: `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`

Erforderlich: Ja

StackStatusGrund

Erfolgs- oder Fehlermeldung im Zusammenhang mit dem Stack-Status.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter CreateStack.

Sicherheitsüberlegungen

Bevor Sie die Aktion `aws:createStack` verwenden können, müssen Sie folgende Richtlinie der IAM-Automation-Assume-Role zuweisen. Weitere Informationen über die Übernahmerolle finden Sie unter Aufgabe 1: Erstellen einer Servicerolle für Automation.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "sqs:*",
            "cloudformation:CreateStack",
            "cloudformation:DescribeStacks"
         ],
         "Resource":"*"
```

```
      }
   ]
}
```

`aws:createTags` - Erstellen von Tags für AWS-Ressourcen

Erstellt neue Tags für Amazon Elastic Compute Cloud (Amazon EC2)-Instances oder AWS Systems Manager verwalteten Instances.

Eingabe

Diese Aktion unterstützt die meisten Amazon EC2-`CreateTags` und Systems Manager `AddTagsToResource`-Parameter. Weitere Informationen finden Sie unter CreateTags und AddTagsToResource.

Das folgende Beispiel zeigt, wie Sie ein Amazon Machine Image (AMI) und eine Instance als Produktionsressourcen für eine bestimmte Abteilung taggen.

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

Die IDs der Ressource(n), die getaggt werden soll(en). Wenn der Ressourcentyp nicht „EC2“ lautet, kann dieses Feld nur ein einzelnes Element enthalten.

Typ: StringList

Erforderlich: Ja

Tags (Markierungen)

Die Tags, die der/den Ressource(n) zugeordnet werden sollen.

Typ: Liste von Karten

Erforderlich: Ja

ResourceType

Der Typ der Ressource(n), die getaggt werden soll(en). Wenn nichts angegeben ist, wird der Standardwert „EC2“ verwendet.

Typ: Zeichenfolge

Required: No

Zulässige Werte: `EC2` | `ManagedInstance` | `MaintenanceWindow` | `Parameter`

Ausgabe

Keine

`aws:deleteImage` - Löschen eines Amazon Machine Image

Löschen Sie das angegebene Amazon Machine Image (AMI) und alle dazugehörigen Snapshots.

Eingabe

Diese Aktion unterstützt nur einen Parameter. Weitere Informationen finden Sie in der Dokumentation zu DeregisterImage und 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

Die ID des Image, das zerstört werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

Ausgabe

Keine

`aws:deleteStack` – Löschen Sie ein AWS CloudFormation-Stack

Löscht einen AWS CloudFormation-Stack.

Eingabe

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

Ein eindeutiger Bezeichner für diese `DeleteStack`-Anfrage. Geben Sie dieses Token an, wenn Sie planen, Anfragen zu wiederholen, damit CloudFormation weiß, dass Sie nicht versuchen, einen Stack mit demselben Namen zu löschen. Sie können `DeleteStack`-Anfragen wiederholen, um zu verifizieren, ob CloudFormation sie empfangen hat.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Muster: [a-zA-Z][-a-zA-Z0-9]*

Required: No

RetainResources.member.N

Diese Eingabe gilt nur für Stacks im Status `DELETE_FAILED`. Eine Liste der logischen Ressourcen-IDs für die Ressourcen, die Sie beibehalten möchten. Während des Löschvorgangs löscht CloudFormation den Stack, löscht jedoch die aufbewahrten Ressourcen nicht.

Das Aufbewahren der Ressourcen ist nützlich, wenn Sie eine Ressource nicht löschen können, wie etwa einen nicht leeren S3-Bucket, Sie aber den Stack löschen möchten.

Typ: Zeichenfolge-Array

Required: No

RoleARN

Der Amazon-Ressourcenname (ARN) für eine AWS Identity and Access Management-(IAM)-Rolle, die CloudFormation zum Erstellen des Stacks annimmt. CloudFormation verwendet die Anmeldeinformationen der Rolle, um Anrufe in Ihrem Auftrag zu tätigen. CloudFormation verwendet diese Rolle stets für alle künftigen Vorgänge am Stack. Wenn Benutzer die Berechtigung für Vorgänge am Stack besitzen, verwendet CloudFormation diese Rolle auch dann, wenn die Benutzer nicht über die Berechtigung zur Weitergabe verfügen. Stellen Sie sicher, dass die Rolle die geringstmögliche Menge an Berechtigungen gewährt.

Wenn Sie keinen Wert angeben, verwendet CloudFormation die Rolle, die dem Stack vorher zugeordnet war. Wenn keine Rolle verfügbar ist, verwendet CloudFormation eine temporäre Sitzung, die anhand Ihrer Benutzeranmeldeinformationen generiert wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 20. Maximale Länge beträgt 2048 Zeichen.

Required: No

StackName

Der Name oder die eindeutige Stack-ID, die dem Stack zugeordnet ist.

Typ: Zeichenfolge

Erforderlich: Ja

Sicherheitsüberlegungen

Bevor Sie die Aktion `aws:deleteStack` verwenden können, müssen Sie folgende Richtlinie der IAM-Automation-Assume-Role zuweisen. Weitere Informationen über die Übernahmerolle finden Sie unter Aufgabe 1: Erstellen einer Servicerolle für Automation.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "sqs:*",
            "cloudformation:DeleteStack",
            "cloudformation:DescribeStacks"
         ],
         "Resource":"*"
      }
   ]
}
```

`aws:executeAutomation` - Führen Sie eine weitere Automatisierung durch

Führt eine sekundäre Automatisierung durch Aufrufen eines sekundären Runbooks aus. Mit dieser Aktion können Sie Runbooks für die gängigsten Vorgänge erstellen und während einer Automatisierung auf diese Runbooks verweisen. Mit dieser Aktion können Sie Ihre Runbooks vereinfachen, indem Sie die Notwendigkeit für wiederholte Schritte bei ähnlichen Runboks entfernen.

Die sekundäre Automatisierung wird im Kontext des Benutzers ausgeführt, der die primäre Automatisierung gestartet hat. Dies bedeutet, dass die sekundäre Automatisierung dieselbe AWS Identity and Access Management (IAM)-Rolle oder denselben Benutzer verwendet wie der Benutzer, der die erste Automatisierung gestartet hat.

> **Important**
>
> Wenn Sie Parameter in einer sekundären Automatisierung festlegen, die eine Übernahmerolle verwenden (eine Rolle, die die iam:passRole-Richtlinie verwendet), muss der Benutzer oder die Rolle, der/die die primäre Automatisierung gestartet hat, über die

Berechtigung zur Weitergabe der Übernahmerolle an die sekundäre Automatisierung verfügen. Weitere Informationen zum Einrichten einer Übernahmerolle für Automation finden Sie unter Methode 2: Konfigurieren von Automation-Rollen mit IAM.

Eingabe

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

Der Name des sekundären Runbooks, das während des Schritts ausgeführt werden soll. Geben Sie für Runbooks im gleichen AWS-Konto den Namen des Runbooks an. Geben Sie für Runbooks, die von einem anderen AWS-Konto geteilt wurden den Amazon-Ressourcennamen (ARN) des Runbooks an. Weitere Informationen zur Verwendung von freigegebenen Runbooks finden Sie unter Verwenden von freigegebenen SSM-Dokumenten.

Typ: Zeichenfolge

Erforderlich: Ja

DocumentVersion

Die Version des sekundären Runbooks, das ausgeführt werden soll. Falls nicht festgelegt, führt Automation die Standardrunbookversion aus.

Typ: Zeichenfolge

Required: No

MaxConcurrency

Die maximale Anzahl von Zielen, für die diese Aufgabe parallel ausgeführt werden dürfen. Sie können eine Zahl, z. B. 10, oder einen Prozentsatz, z. B. 10 %, angeben.

Typ: Zeichenfolge

Required: No

MaxErrors

Die Anzahl der Fehler, die zulässig sind, bevor das System die Automatisierung auf zusätzlichen Zielen stoppt. Sie können entweder eine absolute Anzahl an Fehlern, z. B. 10, oder einen Prozentsatz des festgelegten Ziels, beispielsweise 10 % festlegen. Wenn Sie z. B. 3 angeben, führt das System keine Automatisierung mehr aus, wenn der vierte Fehler empfangen wird. Wenn Sie 0 angeben, führt das System keine weitere Automatisierung auf zusätzlichen Zielen aus, nachdem das erste Fehlerergebnis zurückgegeben wird. Wenn Sie eine Automatisierung auf 50 Ressourcen ausführen und `MaxErrors` auf 10 % setzen, hört das System auf, dass die Automatisierung auf zusätzlichen Zielen auszuführen, sobald der sechste Fehler empfangen wurde.

Automatisierung, die bereits eine ausgeführt werden, wenn der `MaxErrors`-Fehlerschwellenwert erreicht wird, können abgeschlossen werden, einige dieser Automatisierungen können jedoch

dennoch fehlschlagen. Wenn Sie sicherstellen müssen, dass es nicht mehr fehlgeschlagene Automatisierungen als die angegebenen `MaxErrors` geben wird, setzen Sie `MaxConcurrency` auf 1, sodass die Automatisierungen nacheinander ausgeführt werden.

Typ: Zeichenfolge

Required: No

RuntimeParameters

Erforderliche Parameter für das sekundäre Runbook. Das Mapping verwendet das folgende Format: {"parameter1" : "value1", "parameter2" : "value2" }

Typ: Zuordnung

Required: No

Tags (Markierungen)

Optionale Metadaten, die Sie einer Ressource zuweisen. Sie können maximal fünf Tags für eine Automatisierung festlegen.

Typ: MapList

Required: No

TargetLocations

Ein Standort ist eine Kombination aus AWS-Regionen und/oder AWS-Konten, wo Sie die Automatisierung ausführen möchten. Es muss eine Mindestanzahl von 1 Element angegeben werden und eine maximale Anzahl von 100 Elementen kann angegeben werden.

Typ: MapList

Required: No

TargetMaps

Eine Liste von Schlüssel-Wert-Zuweisungen von Dokumentparametern zu Zielressourcen. Sowohl `Targets` als auch `TargetMaps` kann nicht zusammen angegeben werden.

Typ: MapList

Required: No

TargetParameterName

Der Name des Parameters, der als Zielressource für die ratengesteuerte Automatisierung verwendet wird. Erforderlich, wenn Sie `Targets` angeben.

Typ: Zeichenfolge

Required: No

Targets (Ziele)

Eine Liste von Schlüssel-Wert-Zuordnungen zu Zielressourcen. Erforderlich, wenn Sie `TargetParameterName` angeben.

Typ: MapList

Required: No

Ausgabe

Ausgabe

Die von der sekundären Automatisierung generierte Ausgabe. Sie können anhand des folgenden Formats auf die Ausgabe verweisen: *`Secondary_Automation_Step_Name`*.Output

Typ: StringList

Ein Beispiel:

```
- 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

Die ID der sekundären Automatisierung.

Typ: Zeichenfolge

Status

Der Status der sekundären Automatisierung.

Typ: Zeichenfolge

`aws:executeAwsApi` – Aufrufen und Ausführen von AWS API-Operationen

Ruft AWS -API-Operationen auf und führt sie aus. Die meisten API-Operationen werden unterstützt, es wurden jedoch nicht alle API-Operationen getestet. Streaming-API-Operationen, z. B. die -GetObjectOperation, werden nicht unterstützt. Wenn Sie sich nicht sicher sind, ob ein API-Vorgang, den Sie verwenden möchten, ein Streaming-Vorgang ist, lesen Sie in der Boto3-Dokumentation für den Service nach, ob eine API Streaming-Eingaben oder -Ausgaben erfordert. Wir aktualisieren regelmäßig die von dieser Aktion verwendete Boto3-Version. Nach der Veröffentlichung einer neuen Boto3-Version kann es jedoch bis zu mehreren Wochen dauern, bis sich die Änderungen in dieser Aktion niederschlagen. Jede `aws:executeAwsApi`-Aktion kann bis zu einer maximalen Dauer von

25 Sekunden dauern. Weitere Beispiele zur Verwendung dieser Aktion finden Sie unter Weitere Runbook-Beispiele.

Eingaben

Eingaben werden von der ausgewählten API-Operation bestimmt.

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"
      }
   ]
}
```

Service

Der AWS-Service Namespace, der die API-Operation enthält, die Sie ausführen möchten. Sie können eine Liste der unterstützten AWS-Service Namespaces unter Verfügbare Services der

anzeigen AWS SDK for Python (Boto3). Der Namespace befindet sich im Abschnitt Client . Beispielsweise lautet der Namespace für Systems Manager `ssm`. Der Namespace für Amazon Elastic Compute Cloud (Amazon EC2) ist `ec2`.

Typ: Zeichenfolge

Erforderlich: Ja

Api

Der Name der API-Operation, die Sie ausführen möchten. Sie können die API-Operationen (auch als Methoden bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise werden alle API-Vorgänge (Methoden) für Amazon Relational Database Service (Amazon RDS) auf der folgenden Seite aufgelistet: Amazon RDS-Methoden.

Typ: Zeichenfolge

Erforderlich: Ja

API-Operation-Eingaben

Eine oder mehrere API-Eingaben. Sie können die verfügbaren Eingaben (auch als Parameter bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances aus und scrollen Sie nach unten, um die verfügbaren Parameter anzuzeigen, z. B. DB InstanceIdentifier, Name und Werte .

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"
}
```

Typ: Abhängig von der gewählten API-Operation

Erforderlich: Ja

Outputs

Die Ausgaben werden vom Benutzer basierend auf der Antwort des ausgewählten API-Vorgangs angegeben.

Name

Ein Name für die Ausgabe.

Typ: Zeichenfolge

Erforderlich: Ja

Selector

Der JSONPath zu einem bestimmten Attribut im Antwortobjekt. Sie können die Antwortobjekte anzeigen indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie abwärts zum Abschnitt Response Structure (Antwortstruktur). DBInstances wird als Antwortobjekt aufgeführt.

Typ: Ganzzahl, Boolean StringList, Zeichenfolge StringMapoder MapList

Erforderlich: Ja

Typ

Der Datentyp für das Antwortelement.

Typ: Unterschiedlich

Erforderlich: Ja

`aws:executeScript` - Führen Sie ein Skript aus

Führt das bereitgestellte Python- oder PowerShell Skript mit der angegebenen Laufzeit und dem angegebenen Handler aus. Jede `aws:executeScript`-Aktion kann bis zu einer maximalen Dauer von 600 Sekunden (10 Minuten) laufen. Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:executeScript`-Schritt limitieren.

Verwenden Sie Rückgabe-Anweisungen in Ihrer Funktion, um Ihrer Ausgabenutzlast Ausgaben hinzuzufügen. Für Beispiele zum Definieren von Ausgaben für Ihre `aws:executeScript`-Aktion, siehe Beispiel 2: Skriptbasiertes Runbook. Sie können auch die Ausgabe von `aws:executeScript` Aktionen in Ihren Runbooks an die von Ihnen angegebene Amazon CloudWatch Logs-Protokollgruppe senden. Weitere Informationen finden Sie unter Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs.

Wenn Sie die Ausgabe von `aws:executeScript` Aktionen an CloudWatch Logs senden möchten oder wenn die Skripts, die Sie für `aws:executeScript` Aktionen angeben, AWS API-Operationen aufrufen, ist für die Ausführung des Runbooks immer eine AWS Identity and Access Management (IAM-) Service-Rolle (oder Übernahme einer Rolle) erforderlich.

Die `aws:executeScript` Aktion enthält die folgenden vorinstallierten PowerShell Core-Module:

- Microsoft. PowerShell. Gastgeber
- Microsoft. PowerShell. Verwaltung
- Microsoft. PowerShell. Sicherheit
- Microsoft. PowerShell. Hilfsprogramm
- PackageManagement
- PowerShellGet

Um PowerShell Core-Module zu verwenden, die nicht vorinstalliert sind, muss Ihr Skript das Modul mit der `-Force` Markierung installieren, wie im folgenden Befehl gezeigt. Das `AWSPowerShell.NetCore`-Modul wird nicht unterstützt. *`ModuleName`*Ersetzen Sie es durch das Modul, das Sie installieren möchten.

```
Install-Module ModuleName -Force
```

Um PowerShell Core-Cmdlets in Ihrem Skript zu verwenden, empfehlen wir die Verwendung der `AWS.Tools` Module, wie in den folgenden Befehlen gezeigt. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

- Amazon S3 Cmdlets.

```
Install-Module AWS.Tools.S3 -Force
Get-S3Bucket -BucketName bucketname
```

- Amazon EC2 Cmdlets.

```
Install-Module AWS.Tools.EC2 -Force
Get-EC2InstanceStatus -InstanceId instanceId
```

- Allgemeine oder dienstunabhängige AWS Tools for Windows PowerShell Cmdlets.

```
Install-Module AWS.Tools.Common -Force
Get-AWSRegion
```

Wenn Ihr Skript zusätzlich zur Verwendung von PowerShell Core-Cmdlets neue Objekte initialisiert, müssen Sie das Modul auch importieren, wie im folgenden Befehl gezeigt.

```
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
```

Beispiele für die Installation und den Import von `AWS.Tools` Modulen und die Verwendung von PowerShell Core-Cmdlets in Runbooks finden Sie unter. Verwenden von Document Builder zur Erstellung von Runbooks

Eingabe

Geben Sie die zum Ausführen Ihres Skripts erforderlichen Informationen an. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

> **Note**
>
> Der Anhang für ein Python-Skript kann eine .py-Datei oder eine .zip-Datei sein, die das Skript enthält. PowerShell Skripten müssen in ZIP-Dateien gespeichert werden.

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"
    }
}
```

Laufzeit

Die Laufzeitsprache, die für die Ausführung des bereitgestellten Skripts verwendet werden soll. `aws:executeScript`unterstützt die Skripte 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) Core 6.0 (dotnetcore2.1) und 7.0 (dotnetcore3.1). PowerShell PowerShell

python3.7python3.8python3.9python3.10python3.11PowerShell Core 6.0 Unterstützte Werte: | | | | | **PowerShell 7.0**

Typ: Zeichenfolge

Erforderlich: Ja

Handler

Der Name Ihrer Funktion. Sie müssen sicherstellen, dass die im Handler definierte Funktion über zwei Parameter verfügt: `events` und `context`. Die PowerShell Laufzeit unterstützt diesen Parameter nicht.

Typ: Zeichenfolge

Erforderlich: Ja (Python) | Nicht unterstützt (PowerShell)

InputPayload

Ein JSON- oder YAML-Objekt, das an den ersten Parameter des Handlers übergeben wird. Dies kann verwendet werden, um Eingabedaten an das Skript zu übergeben.

Typ: Zeichenfolge

Erforderlich: Nein

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: "python3.8"
        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": "Example"
            }
            ec2.create_tags(
                Resources=[instanceId],
                Tags=[tag]
            )
```

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'
    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 = "Example"

            New-EC2Tag -Resource $input.instanceId -Tag $tag
```

Script

Ein eingebettetes Skript, das während der Automatisierung ausgeführt werden soll.

Typ: Zeichenfolge

Erforderlich: Nein (Python) | Ja (PowerShell)

Attachment

Der Name einer eigenständigen Skriptdatei oder einer ZIP-Datei, die von der Aktion aufgerufen werden kann. Geben Sie denselben Wert wie den `Name` der Dokument-Anhangsdatei an, den Sie im Anforderungsparameter `Attachments` angeben. Weitere Informationen finden Sie unter Anhänge in der API-Referenz für AWS Systems Manager . Wenn Sie ein Skript mithilfe einer Anlage bereitstellen, müssen Sie auch einen `files`-Abschnitt in den Elementen der obersten Ebene Ihres Runbooks definieren. Weitere Informationen finden Sie unter Schema der Version 0.3.

Um eine Datei für Python aufzurufen, verwenden Sie das `filename.method_name`-Format in `Handler`.

> **Note**
>
> Der Anhang für ein Python-Skript kann eine .py-Datei oder eine .zip-Datei sein, die das Skript enthält. PowerShell Skripten müssen in ZIP-Dateien gespeichert werden.

Wenn Sie Python-Bibliotheken in Ihren Anhang einfügen, empfehlen wir, eine leere `__init__.py`-Datei in jedem Modulverzeichnis hinzuzufügen. Auf diese Weise können Sie die Module aus der Bibliothek in Ihrem Anhang innerhalb Ihres Skriptinhalts importieren. Zum Beispiel: `from library import module`

Typ: Zeichenfolge

Erforderlich: Nein

Output

Nutzlast

Die JSON-Darstellung des Objekts, das von Ihrer Funktion zurückgegeben wird. Bis zu 100 KB werden zurückgegeben. Wenn Sie eine Liste ausgeben, werden maximal 100 Elemente zurückgegeben.

`aws:executeStateMachine` – Führen Sie eine AWS Step Functions-State Machine aus.

Führt eine AWS Step Functions-State Machine aus.

Eingabe

Diese Aktion unterstützt die meisten Parameter für die API-Operation StartExecution.

Erforderliche AWS Identity and Access Management (IAM)-Berechtigungen

- `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

Der Amazon-Ressourcenname (ARN) der Step Functions State-Machine.

Typ: Zeichenfolge

Erforderlich: Ja

Name

Der Name der Ausführung.

Typ: Zeichenfolge

Required: No

input

Eine Zeichenfolge, die die JSON-Eingabedaten für die Ausführung enthält.

Typ: Zeichenfolge

Required: No

Outputs

Die folgenden Ausgaben sind für diese Aktion vordefiniert.

executionArn

Der ARN der Ausführung.

Typ: Zeichenfolge

input

Die Zeichenfolge, die die JSON-Eingabedaten der Ausführung enthält. Längenbeschränkungen gelten für die Nutzlastgröße und werden als Bytes in UTF-8-Codierung ausgedrückt.

Typ: Zeichenfolge

Name

Der Name der Ausführung.

Typ: Zeichenfolge

output

Die JSON-Ausgabedaten der Ausführung. Längenbeschränkungen gelten für die Nutzlastgröße und werden als Bytes in UTF-8-Codierung ausgedrückt.

Typ: Zeichenfolge

startDate

Das Datum, an dem die Ausführung gestartet wird.

Typ: Zeichenfolge

stateMachineArn

Der ARN des ausgeführten angegebenen Computers.

Typ: Zeichenfolge

status

Der aktuelle Status der Ausführung.

Typ: Zeichenfolge

stopDate

Wenn die Ausführung bereits beendet wurde, das Datum, an dem die Ausführung beendet wurde.

Typ: Zeichenfolge

`aws:invokeWebhook` – Automation-Webhook-Integration aufrufen

Ruft die angegebene Automation-Webhook-Integration auf. Weitere Informationen zum Erstellen von Automation-Integrationen finden Sie unter Erstellen von Webhook-Integrationen für Automation.

> **Note**
>
> Um die `aws:invokeWebhook`-Aktion zu verwenden, muss Ihre Benutzer- oder Servicerolle die folgenden Aktionen zulassen:
>
> - ssm:GetParameter
> - kms:Decrypt
>
> Die Berechtigung für den `Decrypt`-Vorgang von AWS Key Management Service (AWS KMS) ist nur erforderlich, wenn Sie den Parameter für Ihre Integration mit einem kundenverwalteten Schlüssel verschlüsseln.

Eingabe

Geben Sie die Informationen für die aufzurufende Automation-Integration an.

YAML

```
action: "aws:invokeWebhook"
inputs:
 IntegrationName: "exampleIntegration"
 Body: "Request body"
```

JSON

```
{
    "action": "aws:invokeWebhook",
    "inputs": {
        "IntegrationName": "exampleIntegration",
        "Body": "Request body"
    }
}
```

IntegrationName

Der Name der Automation-Integration. Zum Beispiel `exampleIntegration`. Die von Ihnen angegebene Integration muss bereits vorhanden sein.

Typ: Zeichenfolge

Erforderlich: Ja

Fließtext

Die Nutzlast, die Sie beim Aufrufen der Webhook-Integration senden möchten.

Typ: Zeichenfolge

Required: No

Ausgabe

Antwort

Der Text aus der Antwort des Webhook-Anbieters.

ResponseCode

Der HTTP-Statuscode aus der Antwort des Webhook-Anbieters.

`aws:invokeLambdaFunction` – Aufrufen einer AWS Lambda-Funktion

Ruft die angegebene AWS Lambda-Funktion auf.

> **Note**
>
> Jede `aws:invokeLambdaFunction`-Aktion kann bis zu einer maximalen Dauer von 300 Sekunden (5 Minuten) laufen. Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:invokeLambdaFunction`-Schritt limitieren.

Eingabe

Diese Aktion unterstützt die meisten aufgerufenen Parameter für den Lambda-Service. Weitere Informationen finden Sie unter Aufrufen.

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

Der Name der Lambda-Funktion. Diese Funktion muss vorhanden sein.

Typ: Zeichenfolge

Erforderlich: Ja

Qualifier

Die Version oder der Aliasname der Funktion.

Typ: Zeichenfolge

Required: No

InvocationType

Der Aufruftyp. Der Standardwert ist `RequestResponse`.

Typ: Zeichenfolge

Zulässige Werte: `Event` | `RequestResponse` | `DryRun`

Required: No

LogType

Wenn der Standardwert `Tail` ist, muss der Aufruftyp `RequestResponse` sein. Lambda gibt die letzten 4 KB von Protokolldaten mit base64 verschlüsselt zurück, die von Ihrer Lambda-Funktion vorliegen.

Typ: Zeichenfolge

Zulässige Werte: `None` | `Tail`

Required: No

ClientContext

Die Client-spezifischen Informationen.

Required: No

InputPayload

Ein YAML- oder JSON-Objekt, das an den ersten Parameter des Handlers übergeben wird. Sie können diese Eingabe verwenden, um Daten an die Funktion zu übergeben. Diese Eingabe bietet mehr Flexibilität und Unterstützung als die Legacy-`Payload`-Eingabe. Wenn Sie sowohl `InputPayload` als auch `Payload` für die Aktion definieren, hat `InputPayload` Vorrang, und der `Payload`-Wert wird nicht verwendet.

Typ: StringMap

Required: No

Nutzlast

Eine JSON-Zeichenfolge, die an den ersten Parameter des Handlers übergeben wird. Dies kann verwendet werden, um Eingabedaten an die Funktion zu übergeben. Wir empfehlen die Verwendung der `InputPayload`-Eingabe für zusätzliche Funktionen.

Typ: Zeichenfolge

Required: No

Ausgabe

StatusCode

Der HTTP-Statuscodes.

FunctionError

Falls vorhanden, weist es darauf hin, dass währen der Ausführung der Funktion ein Fehler aufgetreten ist. Fehlerdetails sind in der Antwortnutzlast enthalten.

LogResult

Die mit base64 verschlüsselten Protokolle zum Aufrufen der Lambda-Funktion. Protokolle sind nur dann vorhanden, wenn der Aufrufen-Typ `RequestResponse` ist und die Protokolle angefragt wurden.

Nutzlast

Die JSON-Darstellung des Objekts, das von der Lambda-Funktion zurückgegeben wird. Die Nutzlast ist nur vorhanden, wenn der Aufrufen-Typ `RequestResponse` ist. Bis zu 200 KB werden zurückgegeben.

Das Folgende ist ein Teil des `AWS-PatchInstanceWithRollback`-Runbooks, der zeigt, wie auf Ausgaben der `aws:invokeLambdaFunction`-Aktion verwiesen wird.

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 – Über Schritte in einer Automatisierung iterieren

Diese Aktion wiederholt sich über eine Teilmenge von Schritten in einem Automation-Runbook. Sie können einen Schleifenstil `do while` oder `for each` eine Schleife wählen. Verwenden Sie den `LoopCondition`-Eingabeparameter, um eine `do while`-Schleife zu erstellen. Verwenden Sie die Eingabeparameter `Iterators` und `IteratorDataType`, um eine `for each`-Schleife zu erstellen. Wenn Sie eine `aws:loop`-Aktion verwenden, geben Sie nur entweder den Eingabeparameter `Iterators` oder `LoopCondition` an. Die maximale Anzahl von Iterationen beträgt 100.

Die `onCancel`-Eigenschaft kann nur für Schritte definiert werden, die innerhalb einer Schleife definiert sind. Die `onCancel`-Eigenschaft wird für die `aws:loop`-Aktion nicht unterstützt.

Beispiele

Im Folgenden finden Sie Beispiele für die Erstellung der verschiedenen Typen von Loop-Aktionen.

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
```

Eingabe

Die Eingabe ist wie folgt.

Iteratoren

Die Liste der Elemente, über die die Schritte iteriert werden sollen. Die maximale Anzahl von Iteratoren beträgt 100.

Typ: StringList

Erforderlich: Nein

IteratorDataTyp

Ein optionaler Parameter zur Angabe des Datentyps von `Iterators`. Ein Wert für diesen Parameter kann zusammen mit dem `Iterators`-Eingabeparameter angegeben werden. Wenn Sie keinen Wert für diesen Parameter und `Iterators` angeben, müssen Sie einen Wert für den `LoopCondition`-Parameter angeben.

Typ: Zeichenfolge

Gültige Werte: Boolean | Integer | String | StringMap

Standard: Zeichenfolge

Erforderlich: Nein

LoopCondition

Besteht aus `Variable` und einer auszuwertenden Operatorbedingung. Wenn Sie keinen Wert für diesen Parameter angeben, müssen Sie einen Wert für die `Iterators`- und `IteratorDataType`-Parameter angeben. Sie können komplexe Operatorauswertungen verwenden, indem Sie eine Kombination aus Operatoren `And`, `Not` und `Or` verwenden. Die Bedingung wird bewertet, nachdem die Schritte in der Schleife abgeschlossen sind. Wenn die Bedingung `true` ist und der `MaxIterations`-Wert nicht erreicht wurde, werden die Schritte in der Schleife erneut ausgeführt. Die Bedingungen für den Operator lauten wie folgt:

Zeichenfolgenoperationen

- StringEquals
- EqualsIgnoreFall
- StartsWith
- EndsWith
- Enthält

Numerische Operationen

- NumericEquals
- NumericGreater
- NumericLesser
- NumericGreaterOrEquals

- NumericLesser
- NumericLesserOrEquals

Boolesche Operation

- BooleanEquals

Typ: StringMap

Erforderlich: Nein

MaxIterations

Gibt an, wie oft die Schritte in der Schleife maximal ausgeführt werden. Sobald der für diese Eingabe angegebene Wert erreicht ist, stoppt die Schleife, auch wenn `LoopCondition` immer noch `true` ist oder im `Iterators`-Parameter verbleibende Objekte vorhanden sind.

Typ: Ganzzahl

Zulässige Werte: 1–100

Erforderlich: Nein

Schritte

Die Liste der auszuführenden Schritte. Diese funktionieren wie ein verschachteltes Runbook. In diesen Schritten können Sie mithilfe der `{{loopStepName.CurrentIteratorValue}}`-Syntax auf den aktuellen Iteratorwert für eine `for each`-Schleife zugreifen. Sie können mithilfe der `{{loopStepName.CurrentIteration}}`-Syntax auch auf einen Integer-Wert der aktuellen Iteration für beide Schleifentypen zugreifen.

Typ: Liste der Schritte

Erforderlich: Ja

Output

CurrentIteration

Die aktuelle Schleifeniteration als Ganzzahl. Iterationswerte beginnen bei 1.

Typ: Ganzzahl

CurrentIteratorWert

Der Wert des aktuellen Iterators als Zeichenfolge. Diese Ausgabe ist nur in `for each`-Schleifen vorhanden.

Typ: Zeichenfolge

`aws:pause` - Pausieren einer Automatisierung

Mit dieser Aktion wird die Ausführung der Automatisierung unterbrochen. Nach der Unterbrechung lautet der Automation-Status Waiting. Um die Automatisierung fortzusetzen, verwenden Sie die API-Operation SendAutomationSignal mit dem Signaltyp `Resume`. Wir empfehlen die Verwendung von der `aws:sleep`- oder `aws:approve`-Aktion zur genaueren Kontrolle Ihrer Workflows.

Eingabe

Die Eingabe ist wie folgt.

YAML

```
name: pauseThis
action: aws:pause
inputs: {}
```

JSON

```
{
    "name": "pauseThis",
    "action": "aws:pause",
    "inputs": {}
}
```

Ausgabe

Keine

`aws:runCommand` - Führt einen Befehl auf einer verwalteten Instance aus

Führt die angegebenen Befehle aus.

> **Note**
>
> Automation unterstützt nur die Ausgabe einer AWS Systems Manager Run Command-Aktion. Ein Runbook kann mehrere Run Command-Aktionen und enthalten, die Ausgabe wird allerdings nur für je eine Aktion unterstützt.

Eingabe

Diese Aktion unterstützt die meisten Befehlsendeparameter. Weitere Informationen finden Sie unter 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

Wenn das Dokument vom Typ Command Ihnen gehört AWS, oder geben Sie den Namen des Dokuments an. Geben Sie den Amazon-Ressourcennamen (ARN) des Dokuments an, wenn Sie ein Dokument verwenden, das von einem anderen AWS-Konto mit Ihnen geteilt wird. Weitere Informationen zur Verwendung von geteilten Dokumenten finden Sie unter Verwenden von freigegebenen SSM-Dokumenten.

Typ: Zeichenfolge

Erforderlich: Ja

InstanceIds

Die Instance-IDs, auf denen Sie den Befehl ausführen möchten. Sie können maximal 50 IDs angeben.

Sie können auch Pseudoparameter `{{RESOURCE_ID}}` anstelle von Instance-IDs verwenden, um den Befehl auf allen Instances in der Zielgruppe auszuführen. Weitere Informationen zu Pseudoparametern finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben.

Alternativ können Sie Befehle mit dem Parameter `Targets` an eine Instance-Flotte senden. Die `Targets`-Parameter akzeptiert Amazon Elastic Compute Cloud (Amazon EC2)-Tags. Weitere Informationen zur Verwendung des Parameters `Targets` finden Sie unter Ausführen von Befehlen in großem Maßstab.

Typ: StringList

Erforderlich: Nein (Wenn Sie den `{{RESOURCE_ID}}` Pseudo-Parameter nicht angeben InstanceIds oder verwenden, müssen Sie den `Targets` Parameter angeben.)

Targets (Ziele)

Ein Array von Suchkriterien, die Instances mit einer Schlüssel-Wert-Kombination adressieren, die Sie angeben. Der Wert `Targets` ist erforderlich, wenn Sie nicht eine oder mehrere Instance-IDs im Aufruf angeben. Weitere Informationen zur Verwendung des Parameters `Targets` finden Sie unter Ausführen von Befehlen in großem Maßstab.

Typ: MapList (Das Schema der Map in der Liste muss mit dem Objekt übereinstimmen.) Informationen finden Sie unter Target in der AWS Systems Manager -API-Referenz.

Erforderlich: Nein (Wenn Sie nichts angeben`Targets`, müssen Sie den `{{RESOURCE_ID}}` Pseudo-Parameter angeben InstanceIds oder verwenden.)

Im Folgenden sehen Sie ein Beispiel.

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"
            ]
        }
    }
}
```

Parameter

Die erforderlichen und optionalen Parameter, die im Dokument angegeben sind.

Typ: Zuordnung

Erforderlich: Nein

CloudWatchOutputConfig

Konfigurationsoptionen für das Senden von Befehlsausgaben an Amazon CloudWatch Logs. Weitere Informationen zum Senden von Befehlsausgaben an CloudWatch Logs finden Sie unterKonfiguration von Amazon CloudWatch Logs für Run Command.

Typ: StringMap (Das Schema der Map muss mit dem Objekt übereinstimmen. Weitere Informationen finden Sie CloudWatchOutputConfigin der AWS Systems Manager API-Referenz).

Erforderlich: Nein

Im Folgenden sehen Sie ein Beispiel.

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
        }
    }
}
```

Kommentar

Benutzerdefinierte Informationen über den Befehl.

Typ: Zeichenfolge

Erforderlich: Nein

DocumentHash

Der Hash für das Dokument.

Typ: Zeichenfolge

Erforderlich: Nein

DocumentHashGeben Sie ein

Der Typ des Hash.

Typ: Zeichenfolge

Zulässige Werte: `Sha256` | `Sha1`

Erforderlich: Nein

NotificationConfig

Die Konfigurationen für das Senden von Benachrichtigungen.

Erforderlich: Nein

Ausgänge 3 BucketName

Der Name des S3-Buckets für Befehlsausgabeantworten.

Typ: Zeichenfolge

Erforderlich: Nein

Ausgänge3 KeyPrefix

Das Präfix.

Typ: Zeichenfolge

Erforderlich: Nein

ServiceRoleArn

Der ARN der AWS Identity and Access Management (IAM-) Rolle.

Typ: Zeichenfolge

Erforderlich: Nein

TimeoutSeconds

Die Wartezeit in Sekunden, bis ein Befehl AWS Systems Manager SSM Agent an die Instance übermittelt wird. Wenn der Befehl vom SSM Agent auf der Instance vor dem Erreichen des angegebenen Werts nicht empfangen wurde, ändert sich der Status des Befehls in `Delivery Timed Out`.

Typ: Ganzzahl

Erforderlich: Nein

Gültige Werte: 30-2592000

Output

CommandId

Die ID des Befehls.

Status

Der Status des Befehls.

ResponseCode

Der Antwortcode des Befehls. Wenn das Dokument, das Sie ausführen, mehr als einen Schritt umfasst, wird für diese Ausgabe kein Wert zurückgegeben.

Output

Die Ausgabe des Befehls. Wenn Sie mit Ihrem Befehl auf ein Tag oder mehrere Instanzen abzielen, wird kein Ausgabewert zurückgegeben. Sie können die `ListCommandInvocations` API-Operationen `GetCommandInvocation` und verwenden, um Ausgaben für einzelne Instanzen abzurufen.

`aws:runInstances` - So starten Sie eine Amazon-EC2-Instance

Startet eine neue Amazon Elastic Compute Cloud (Amazon EC2)-Instance.

Eingabe

Die Aktion unterstützt die meisten API-Parameter. Weitere Informationen finden Sie in der RunInstances-API-Dokumentation.

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

Reserved Instances.

Typ: Zeichenfolge

Required: No

BlockDeviceMappings

Die Blockgeräte für die Instance.

Typ: MapList

Required: No

ClientToken

Der Bezeichner, um die Idempotenz der Anfrage sicherzustellen.

Typ: Zeichenfolge

Required: No

DisableApiTermination

Aktiviert oder deaktiviert die Instance-API-Beendigung.

Typ: Boolesch

Required: No

EbsOptimized

Aktiviert oder deaktiviert die Amazon Elastic Block Store (Amazon EBS)-Optimierung.

Typ: Boolesch

Required: No

IamInstanceProfileArn

Der Amazon-Ressourcenname (ARN) des AWS Identity and Access Management-(IAM)-Instance-Profils für die Instance.

Typ: Zeichenfolge

Required: No

IamInstanceProfileName

Der Name des IAM-Instance-Profils für die Instance.

Typ: Zeichenfolge

Required: No

ImageId

Die ID des Amazon Machine Image (AMI).

Typ: Zeichenfolge

Erforderlich: Ja

InstanceInitiatedShutdownBehavior

Gibt an, ob die Instance beim Herunterfahren des Systems angehalten oder beendet wird.

Typ: Zeichenfolge

Required: No

InstanceType

Der Instance-Typ.

> **Note**
>
> Wenn kein Wert für den Instance-Typ angegeben wird, wird der Instance-Typ m1.small verwendet.

Typ: Zeichenfolge

Required: No

KernelId

Die ID des Kernels.

Typ: Zeichenfolge

Required: No

KeyName

Der Name des Schlüsselpaars.

Typ: Zeichenfolge

Required: No

MaxInstanceCount

Die Höchstanzahl zu startender Instances.

Typ: Zeichenfolge

Required: No

MetadataOptions

Die Metadatenoptionen für die Instance. Weitere Informationen finden Sie unter InstanceMetadataOptionsRequest.

Typ: StringMap

Required: No

MinInstanceCount

Die Mindestanzahl zu startender Instances.

Typ: Zeichenfolge

Required: No

Überwachung

Aktiviert oder deaktiviert die detaillierte Überwachung.g

Typ: Boolesch

Required: No

NetworkInterfaces

Die Netzwerkschnittstellen.

Typ: MapList

Required: No

Placement

Die Platzierung für die Instance.

Typ: StringMap

Required: No

PrivateIpAddress

Die primäre IPv4-Adresse.

Typ: Zeichenfolge

Required: No

RamdiskId

Die ID des RAM-Datenträgers.

Typ: Zeichenfolge

Required: No

SecurityGroupIds

Die IDs der Sicherheitsgruppen für die Instance.

Typ: StringList

Required: No

SecurityGroups

Die Namen der Sicherheitsgruppen für die Instance.

Typ: StringList

Required: No

SubnetId

Die Subnetz-ID.

Typ: Zeichenfolge

Required: No

TagSpecifications

Die Tags, die beim Start auf die Ressourcen angewendet werden. Instances und Volumes können nur beim Start mit Tags versehen werden. Die angegebenen Tags werden auf alle Instances bzw. Volumes angewendet, die beim Start erstellt werden. Um eine Instance nach dem Start mit Tags zu versehen, verwenden Sie die Aktion `aws:createTags` - Erstellen von Tags für AWS-Ressourcen.

Type: MapList (Weitere Informationen finden Sie unter TagSpecification.)

Required: No

UserData

Ein Skript, das als Zeichenfolgenliteralwert bereitgestellt wird. Wenn ein Literalwert eingegeben wird, muss er Base64-kodiert sein.

Typ: Zeichenfolge

Required: No

Ausgabe

InstanceIds

Die IDs der Instances.

InstanceStates

Der Status der Instance.

`aws:sleep` - Verzögerung einer Automatisierung

Verzögert eine Automatisierung um eine bestimmte Zeit. Diese Aktion verwendet das Datums- und Uhrzeitformat der International Organization for Standardization (ISO) 8601. Weitere Informationen zu diesem Datums- und Uhrzeitformat finden Sie unter ISO 8601.

Eingabe

Sie können eine Automatisierung um eine festgelegte Dauer verzögern.

YAML

```
name: sleep
action: aws:sleep
inputs:
  Duration: PT10M
```

JSON

```
{
   "name":"sleep",
   "action":"aws:sleep",
   "inputs":{
      "Duration":"PT10M"
   }
}
```

Sie können eine Automatisierung auch bis zu einem festgelegten Zeitpunkt verzögern. Wenn das Datum und die Uhrzeit verstrichen sind, erfolgt die Aktion unmittelbar.

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**
>
> Automation unterstützt eine maximale Verzögerung von 604799 Sekunden (7 Tage).

Dauer

Ein ISO 8601-Dauer. Sie können keine negative Dauer angeben.

Typ: Zeichenfolge

Required: No

Zeitstempel

Ein ISO 8601-Zeitstempel. Wenn Sie keinen Wert für diesen Parameter angeben, müssen Sie einen Wert für den `Duration`-Parameter angeben.

Typ: Zeichenfolge

Required: No

Ausgabe

Keine

`aws:updateVariable` – Aktualisiert einen Wert für eine Runbook-Variable

Diese Aktion aktualisiert einen Wert für eine Runbook-Variable. Der Datentyp des Werts muss dem Datentyp der Variable entsprechen, die Sie aktualisieren möchten. Datentypkonvertierungen werden nicht unterstützt. Die `onCancel`-Eigenschaft wird für die `aws:updateVariable`-Aktion nicht unterstützt.

Eingabe

Die Eingabe ist wie folgt.

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"]
    }
}
```

Name

Der Name der Variable, deren Wert Sie aktualisieren möchten. Sie müssen das Format `variable:variable name` verwenden

Typ: Zeichenfolge

Erforderlich: Ja

Wert

Der neue Wert, der der Variable zugewiesen werden soll. Der Wert muss mit dem Datentyp der Variable übereinstimmen. Datentypkonvertierungen werden nicht unterstützt.

Typ: Boolean | Ganzzahl | MapList | Zeichenfolge | StringList | StringMap

Erforderlich: Ja

Einschränkungen:

- MapList kann eine maximale Anzahl von 200 Elementen enthalten.
- Schlüssellängen können eine Mindestlänge von 1 und eine Maximallänge von 50 haben.
- StringList kann eine Mindestanzahl von 0 Elementen und eine maximale Anzahl von 50 Elementen sein.

- Die Länge einer Zeichenfolge kann eine Mindestlänge von 1 und eine Maximallänge von 512 haben.

Output

None

`aws:waitForAwsResourceProperty` - Warten Sie auf eine AWS-Ressourceneigenschaft

Die `aws:waitForAwsResourceProperty`-Aktion erlaubt Ihrer Automatisierung auf einen bestimmten Ressourcenstatus oder Ereignisstatus zu warten, bevor Sie die Automatisierung fortsetzen. Weitere Beispiele zur Verwendung dieser Aktion finden Sie unter Weitere Runbook-Beispiele.

> **Note**
>
> Der Standardwert für die Zeitüberschreitung für diese Aktion beträgt 3 600 Sekunden (eine Stunde). Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:waitForAwsResourceProperty`-Schritt anpassen. Weitere Informationen und Beispiele zur Verwendung dieser Aktion finden Sie unter Behandeln von Timeouts in Runbooks.

Eingabe

Eingaben werden von der ausgewählten API-Operation bestimmt.

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"
    ]
  }
}
```

Service

Der AWS-Service-Namespace, der die API-Operation enthält, die Sie ausführen möchten. Beispielsweise ist der Namespace für AWS Systems Manager `ssm`. Der Namespace für Amazon Elastic Compute Cloud (Amazon EC2) ist `ec2`. Sie finden eine Liste der unterstützten AWS-Service-Namespaces im Abschnitt Verfügbare Services der AWS CLI-Befehlsreferenz.

Typ: Zeichenfolge

Erforderlich: Ja

Api

Der Name der API-Operation, die Sie ausführen möchten. Sie können die API-Operationen (auch als Methoden bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise werden alle API-Vorgänge (Methoden) für Amazon Relational Database Service (Amazon RDS) auf der folgenden Seite aufgelistet: Amazon RDS-Methoden.

Typ: Zeichenfolge

Erforderlich: Ja

API-Operation-Eingaben

Eine oder mehrere API-Eingaben. Sie können die verfügbaren Eingaben (auch als Parameter bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden

Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie abwärts, um die verfügbaren Parameter zu sehen, wie etwa DBInstanceIdentifier, Name und Values (Werte).

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"
}
```

Typ: Abhängig von der gewählten API-Operation

Erforderlich: Ja

PropertySelector

Der JSONPath zu einem bestimmten Attribut im Antwortobjekt. Sie können die Antwortobjekte anzeigen indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie abwärts zum Abschnitt Response Structure (Antwortstruktur). DBInstances wird als Antwortobjekt aufgeführt.

Typ: Zeichenfolge

Erforderlich: Ja

DesiredValues

Die erwartete Status oder Zustand, bei dem die Automatisierung fortgesetzt werden soll.

Typ: MapList, StringList

Erforderlich: Ja

Systemvariablen für Automation

AWS Systems Manager-Automation-Runbooks verwenden die folgenden Variablen. Ein Beispiel für die Verwendung dieser Variablen erhalten Sie, wenn Sie die JSON-Quelle des `AWS-UpdateWindowsAmi`-Runbooks anzeigen.

So zeigen Sie die JSON-Quelle des **`AWS-UpdateWindowsAmi`**-Runbooks an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie in der Dokumentliste entweder über die Suchleiste oder die Zahlen rechts neben der Suchleiste das Runbook **`AWS-UpdateWindowsAmi`** aus.
4. Wählen Sie die Registerkarte Content aus.

Systemvariablen

Automation-Runbooks unterstützen die folgenden Variablen.

Variable	Details
`global:ACCOUNT_ID`	Die AWS-Konto-ID des Benutzers oder der Rolle, in dem die Automatisierung ausgeführt wird.
`global:DATE`	Das Datum (zur Ausführungszeit) im Format yyyy-MM-dd.
`global:DATE_TIME`	Das Datum und die Uhrzeit (zur Ausführun gszeit) im Format yyyy-MM-dd_HH.mm.ss.

Variable	Details
`global:AWS_PARTITION`	Die Partition, in der sich die Ressource befindet. Für Standard-AWS-Regionen lautet die Partition `aws`. Für Ressourcen in anderen Partitionen lautet die Partition `aws-`*`partition name`* . Beispielsweise ist die Partition für Ressourcen in der AWS-GovCloud (US-West)-Region `aws-us-gov` .
`global:REGION`	Die Region, in der das Runbook ausgeführt wird. Beispiel: us-east-2.

Variablen für Automation

Automation-Runbooks unterstützen die folgenden Automatisierungsvariablen.

Variable	Details
`automation:EXECUTION_ID`	Die eindeutige ID, die der aktuellen Automatisierung zugewiesen ist. Zum Beispiel `1a2b3c-1a2b3c-1a2b3c-1a2b3c-1a2b3c1a2b3c` .

Themen

- Terminologie
- Unterstützte Szenarien
- Nicht unterstützte Szenarien

Terminologie

Die folgenden Bedingungen beschreiben, wie Variablen und Parameter gelöst werden.

Begriff	Definition	Beispiel
Konstanter ARN	Ein gültiger Amazon-Ressourcenname (ARN) ohne Variablen.	`arn:aws:iam::123456789012:role/roleName`
Runbook-Parameter	Ein auf der Runbook-Ebene definierter Parameter (z. B. `instanceId`). Der Parameter wird in einer grundlegenden Zeichenfolgenersetzung verwendet. Sein Wert wird zur Startausführungszeit bereitgestellt.	`{ "description": "Create Image Demo", "version": "0.3", "assumeRole": "Your_Automation_Assume_Role_ARN ", "parameters":{ "instanceId": { "type": "String", "description": "Instance to create image from" } }`
Systemvariable	Eine allgemeine Variable, die in das Runbook eingefügt wird, wenn ein beliebiger Teil des Runbooks bewertet wird.	`"activities": [ { "id": "copyImage", "activityType": "AWS-CopyImage", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "ImageName": "{{imageName}}", "SourceImageId": "{{sourceImageId}}", "SourceRegion": "{{sourceRegion}}",`

Begriff	Definition	Beispiel
		"Encrypted": true, "ImageDescription": "Test CopyImage Description created on **{{global:DATE}}** " } }]

Begriff	Definition	Beispiel
Variable für Automation	Eine Variable, die sich auf die Automatisierung bezieht, die in das Runbook eingefügt wird, wenn ein Teil des Runbooks bewertet wird.	{ "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}}** ”] } } }

Begriff	Definition	Beispiel
Systems Manager-Parameter	Eine in AWS Systems Manager Parameter Store definierte Variable. Eine direkte Referenzierung in der Schritteingabe ist nicht möglich. Eventuell sind für den Zugriff auf den Parameter Berechtigungen erforderlich.	<pre>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</pre>

Begriff	Definition	Beispiel
		timeoutSeconds: 1200 onFailure: Abort inputs: ImageId: '{{LatestAmi}}' ...

Unterstützte Szenarien

Szenario	Kommentare	Beispiel
Konstanter ARN `assumeRole` beim Erstellen.	Es wird eine Autorisierungsprüfung durchgeführt, um zu bestätigen, dass der aufrufende Benutzer über die Berechtigung zum Übergeben der Rolle `assumeRole` verfügt.	{ "description": "Test all Automation resolvable parameters", "schemaVersion": "0.3", "assumeRole": ***"arn:aws:iam::123456789012:role/roleName"*** , "parameters": { ...
Der Runbook-Parameter wird für `AssumeRole` bereitgestellt, wenn die Automatisierung gestartet wird.	Muss in der Parameterliste des Runbooks definiert werden.	{ "description": "Test all Automation resolvable parameters", "schemaVersion": "0.3", "assumeRole": **"{{dynamicARN}}"** , "parameters": { ...

Szenario	Kommentare	Beispiel
Für Runbookparameter beim Start bereitgestellter Wert.	Der Kunde stellt den für einen Parameter zu verwendenden Wert bereit. Alle zur bereitgestellten Eingaben müssen in der Parameterliste des Runbooks definiert sein.	<pre>... "parameters": { "amiId": { "type": "String", "default": "ami-12345678 ", "description": "list of commands to run as part of first step" }, ...</pre> Eingaben zum Start der Automation-Ausführung umfassen : `{"amiId" : ["ami-12345678 "] }`

Szenario	Kommentare	Beispiel
Systems Manager Parameter , auf den im Runbook-Inhalt verwiesen wird.	Die Variable existiert im Kundenkonto oder ist ein öffentlich zugänglic her Parameter und die `AssumeRole` für das Runbook hat Zugriff auf die Variable. Beim Erstellen wird eine Überprüfung durchgefü hrt, um zu bestätigen, dass `AssumeRole` Zugriff hat. Der Parameter kann nicht direkt in der Schritteingabe referenziert werden.	``` ... 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}}' ... ```

Szenario	Kommentare	Beispiel
Die Systemvariable, auf die in der Definition des Schritts verwiesen wird	Eine Systemvariable wird beim Start der Automatisierung in das Runbook eingefügt. Der in das Runbook eingefügte Wert steht in Relation zum Zeitpunkt des Einfügens. Das bedeutet, dass der Wert einer Zeitvariable, die in Schritt 1 eingefügt wurde, aufgrund der erforderlichen Zeit für die Ausführung der Schritte vom in Schritt 3 eingefügten Wert abweicht. Systemvariablen müssen nicht in der Parameter liste des Runbooks festgelegt werden.	``` ... "mainSteps": [{ "name": "RunSomeC ommands", "action": "aws:runCommand", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "DocumentName": "AWS:RunPowerShell", "InstanceIds": ["{{LaunchInstance .InstanceIds}}"], "Parameters": { "commands " : ["echo {The time is now {{global:DATE_TIME }}}"] } } }, ... ```

Szenario	Kommentare	Beispiel
Die Automation-Variable, auf die in der Definition des Schritts verwiesen wird.	Automation-Variablen müssen nicht in der Parameterliste des Runbooks festgelegt werden. Die einzige unterstützte AAutomation-Variable ist automation:EXECUTION_ID.	``` ... "mainSteps": [{ "name": "invokeLambdaFunction", "action": "aws:invokeLambdaFunction", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "FunctionName": "Hello-World-LambdaFunction", "Payload" : "{ "executionId" : "{{automation:EXECUTION_ID}}" }" } } ... ```

Szenario	Kommentare	Beispiel
Weitere Informationen finden Sie in der Ausgabe des vorherigen Schritts in der Definition des nächsten Schritts.	Dies ist die Parameter umleitung. Mithilfe der Syntax `{{stepName.OutputName}}` wird auf die Ausgabe eines vorherigen Schritts verwiesen. Diese Syntax kann vom Kunden nicht für Runbookparameter verwendet werden. Dies wird behoben, wenn der verweisende Schritt ausgeführt wird. Der Parameter ist nicht in der Liste der Parameter des Runbooks aufgeführt.	<pre>... "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" } } ...</pre>

Nicht unterstützte Szenarien

Szenario	Kommentar	Beispiel
Systems Manager Parameter bereitgestellt für `assumeRole` beim Erstellen	Nicht unterstützt	`...` `{` `  "description": "Test all Automation resolvable parameters",` `  "schemaVersion": "0.3",` `  "assumeRole": "{{ssm:administratorRoleARN}}  ",` `  "parameters": {` `...`
System Manager-Parameter, der direkt in der Schritteingabe referenziert wird.	Gibt eine `InvalidDocumentContent` - Ausnahme zur Erstellungszeit zurück.	`...` `mainSteps:` `  - name: launchInstance` `    action: 'aws:runInstances'` `    maxAttempts: 3` `    timeoutSeconds: 1200` `    onFailure: Abort` `    inputs:` `      ImageId: '{{ssm:/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-Base}}'` `...`

Szenario	Kommentar	Beispiel
Variablenschrittdefinition	Die Definition eines Schritts im Runbook wird anhand von Variablen zusammengestellt.	... "mainSteps": [{ "name": "LaunchInstance", "action": "aws:runInstances", "**{{attemptModel}}** ": 1, "onFailure": "Continue", "inputs": { "ImageId": "*ami-12345678* ", "MinInstanceCount": 1, "MaxInstanceCount": 2 } ... User supplies input : { "attemptModel" : "**minAttempts** " }

Szenario	Kommentar	Beispiel
Querverweise auf Runbook-Parameter	Der Benutzer liefert zur Startzeit einen Eingabeparameter, der ein Verweis auf einen anderen Parameter im Runbook ist.	... "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}}" }, ...

Szenario	Kommentar	Beispiel
Multi-Level-Expansion	Das Runbook definiert eine Variable, die den Namen einer Variablen ergibt. Dieser befindet sich in den Variablen trennzeichen (d. h. {{ }}) und wird auf den Wert dieser Variable/dieses Parameters erweitert.	... "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}}*** }}"] }

Szenario	Kommentar	Beispiel
		`` ... Note: The customer intention here would be to run a command of "echo {Hello world}" ```

Szenario	Kommentar	Beispiel
Verweis auf die Ausgabe aus einem Runbook-Schritt, bei dem es sich um einen anderen Variablentyp handelt	Der Benutzer verweist auf die Ausgabe eines vorherige n Runbook-Schritts in einem späteren Schritt. Die Ausgabe ist ein Variablentyp, der nicht den Anforderungen der Aktion des nachfolgenden Schritts erfüllt.	``` ... 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 ```

Szenario	Kommentar	Beispiel
		`outputs a "StringList" type variable.`

Erstellen Ihrer eigenen Runbooks

Ein Automatisierungs-Runbook definiert die Aktionen, die Systems Manager auf Ihren verwalteten Instanzen und anderen AWS Ressourcen ausführt, wenn eine Automatisierung ausgeführt wird. Automatisierung ist eine Fähigkeit von. AWS Systems Manager Ein Runbook enthält einen oder mehrere Schritte, die in sequenzieller Reihenfolge ausgeführt werden. Jeder Schritt basiert auf einer einzigen Aktion. Die Ausgabe von einem Schritt kann als Eingabe in einem späteren Schritt verwendet werden.

Der Prozess der Ausführung dieser Aktionen und ihrer Schritte wird als Automatisierung bezeichnet.

Mit den für Runbooks unterstützten Aktionstypen können Sie eine Vielzahl von Vorgängen in Ihrer AWS Umgebung automatisieren. Mithilfe des `executeScript` Aktionstyps können Sie beispielsweise eine Python oder ein PowerShell Skript direkt in Ihr Runbook einbetten. (Wenn Sie ein benutzerdefiniertes Runbook erstellen, können Sie Ihr Skript inline hinzufügen oder es von einem S3-Bucket oder von Ihrem lokalen Computer aus anhängen.) Sie können die Verwaltung Ihrer AWS CloudFormation Ressourcen automatisieren, indem Sie die `deleteStack` Aktionstypen `createStack` und verwenden. Darüber hinaus kann ein Schritt mithilfe des `executeAwsApi` Aktionstyps jede beliebige API-Operation ausführen AWS-Service, z. B. das Erstellen oder Löschen von AWS Ressourcen, das Starten anderer Prozesse, das Initiieren von Benachrichtigungen und vieles mehr.

Eine Liste aller 20 unterstützten Aktionstypen für Automation finden Sie unter Systems Manager Automation Aktionen-Referenz.

AWS Systems Manager Automation bietet mehrere Runbooks mit vordefinierten Schritten, mit denen Sie allgemeine Aufgaben wie den Neustart einer oder mehrerer Amazon Elastic Compute Cloud (Amazon EC2) -Instances oder das Erstellen einer () ausführen können. Amazon Machine Image AMI Sie können auch Ihre eigenen Runbooks erstellen und sie mit anderen teilen oder sie für AWS-Konten alle Automation-Benutzer veröffentlichen.

Runbooks werden mit YAML oder JSON geschrieben. Mit dem Document Builder in der Systems Manager-Automation-Konsole können Sie jedoch ein Runbook erstellen, ohne nativen JSON- oder YAML-Code erstellen zu müssen.

> **Important**
>
> Wenn Sie einen automatisierten Workflow ausführen, der andere Services mithilfe einer AWS Identity and Access Management -(IAM)-Servicerolle aufruft, muss die Servicerolle mit der Berechtigung zum Aufrufen dieser Services konfiguriert sein. Diese Anforderung gilt für alle AWS Automation-Runbooks (`AWS-*`-Runbooks), wie zum Beispiel `AWS-ConfigureS3BucketLogging`, `AWS-CreateDynamoDBBackup` und `AWS-RestartEC2Instance`-Runbooks, um nur einige zu nennen. Diese Anforderung gilt auch für alle benutzerdefinierten Automatisierungs-Runbooks, die Sie erstellen und die andere mithilfe AWS-Services von Aktionen aufrufen, die andere Dienste aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:createStack`- oder `aws:copyImage`-Aktionen verwenden, konfigurieren Sie die Dienstrolle mit der Berechtigung zum Aufrufen solcher Services. Sie können anderen Berechtigungen erteilen, AWS-Services indem Sie der Rolle eine IAM-Inline-Richtlinie hinzufügen. Weitere Informationen finden Sie unter (Optional) Fügen Sie eine Inline-Automatisierungsrichtlinie oder eine vom Kunden verwaltete Richtlinie hinzu, um andere aufzurufen AWS-Services.

Informationen zu den Aktionen, die Sie in einem Runbook angeben können, finden Sie unter Systems Manager Automation Aktionen-Referenz.

Informationen zur Verwendung von AWS Toolkit for Visual Studio Code zum Erstellen von Runbooks finden Sie unter Arbeiten mit Systems Manager Automation-Dokumenten im AWS Toolkit for Visual Studio Code Benutzerhandbuch.

Informationen zur Verwendung des Visual Designers zum Erstellen eines benutzerdefinierten Runbooks finden Sie unter. Visuelle Designerfahrung für Automation-Runbooks

Inhalt

- Visuelle Designerfahrung für Automation-Runbooks
  - Bevor Sie beginnen
  - Überblick über die Benutzeroberfläche für visuelle Designerfahrung
    - Aktionsbrowser

    - Leinwand
    - Formular
    - Tastenkombinationen
  - Die visuelle Designerfahrung nutzen
    - Einen Runbook-Workflow erstellen
    - Ein Runbook entwerfen
    - Ihr Runbook aktualisieren
    - Ihr Runbook exportieren
  - Konfigurieren von Eingaben und Ausgaben für Ihre Aktionen
    - Eingabedaten für eine Aktion angeben
    - Die Ausgabedaten für eine Aktion definieren
  - Fehlerbehandlung bei der visuellen Designerfahrung
    - Bei einem Fehler die Aktion erneut versuchen
    - Timeouts
    - Fehlgeschlagene Aktionen
    - Abgebrochene Aktionen
    - Kritische Aktionen
    - Aktionen beenden
  - Tutorial: Ein Runbook mithilfe der visuellen Designerfahrung erstellen
    - Schritt 1: Zur visuellen Designerfahrung navigieren
    - Schritt 2: Einen Workflow erstellen
    - Schritt 3: Den automatisch generierten Code überprüfen
    - Schritt 4: Ihr neues Runbook ausführen
    - Schritt 5: Bereinigen
- Erstellen von Automation-Runbooks
  - Identifizieren Sie Ihren Anwendungsfall
  - Einrichten Ihrer Entwicklungsumgebung
  - Entwickeln von Runbook-Inhalten
  - Beispiel 1: Erstellen von über- und untergeordneten Runbooks
    - Erstellen des untergeordneten Runbooks

        - Erstellen des übergeordneten Runbooks
    - Beispiel 2: Skriptbasiertes Runbook
    - Weitere Runbook-Beispiele
        - Bereitstellung der VPC-Architektur und der Microsoft Active Directory-Domänencontroller
        - Wiederherstellen eines Root-Volumes aus dem letzten Snapshot
        - Erstellen eines AMI und einer regionenübergreifenden Kopie
- Eingabeparameter erstellen, die Ressourcen auffüllen AWS
- Verwenden von Document Builder zur Erstellung von Runbooks
    - Erstellen eines Runbooks mithilfe von Document Builder
    - Erstellen eines Runbooks, das Skripte ausführt
- Verwenden von Skripten in Runbooks
    - Berechtigungen für die Verwendung von Runbooks
    - Hinzufügen von Skripts zu Runbooks
    - Skripteinschränkungen für Runbooks
- Verwendung bedingter Anweisungen in Runbooks
    - Arbeiten mit der aws:branch-Aktion
        - Erstellen eines aws:branch-Schritts in einem Runbook
            - Informationen zum Erstellen der Ausgabevariable
        - Beispiel aws:branch-Runbooks
        - Erstellen komplexer verzweigender Automatisierungen mit Operatoren
    - Beispiele für die Verwendung von bedingten Optionen
- Verwenden von Aktionsausgaben als Eingaben
    - Verwenden von JSONPath in Runbooks
- Erstellen von Webhook-Integrationen für Automation
    - Erstellen von Integrationen (Konsole)
    - Erstellen von Integrationen (Befehlszeile)
    - Erstellen von Webhooks für Integrationen
- Behandeln von Timeouts in Runbooks

Visuelle Designerfahrung für Automation-Runbooks

AWS Systems Manager Automation bietet eine visuelle Designerfahrung mit geringem Programmieraufwand, mit dem Sie Automation-Runbooks erstellen können. Die visuelle Designerfahrung bietet eine Drag-and-Drop-Oberfläche mit der Option, Ihren eigenen Code hinzuzufügen, sodass Sie Runbooks einfacher erstellen und bearbeiten können. Mit der visuellen Designerfahrung können Sie Folgendes tun:

- Bedingte Anweisungen steuern.
- Steuern Sie, wie Eingabe und Ausgabe für jede Aktion gefiltert oder transformiert werden.
- Konfigurieren Sie die Fehlerbehandlung.
- Erstellen Sie Prototypen für neue Runbooks.
- Verwenden Sie Ihre Prototyp-Runbooks als Ausgangspunkt für die lokale Entwicklung mit AWS Toolkit for Visual Studio Code.

Wenn Sie ein Runbook erstellen oder bearbeiten, können Sie über die Automation-Konsole auf die visuelle Designerfahrung zugreifen. Wenn Sie ein Runbook erstellen, überprüft die visuelle Designerfahrung Ihre Arbeit und generiert automatisch Code. Sie können den generierten Code überprüfen oder ihn für die lokale Entwicklung exportieren. Wenn Sie fertig sind, können Sie Ihr Runbook speichern, ausführen und die Ergebnisse in der Systems-Manager-Automation-Konsole überprüfen.

Bevor Sie beginnen

Um die visuelle Designerfahrung nutzen zu können, benötigen Sie AWS-Konto und Anmeldeinformationen, die die richtigen Berechtigungen für alle Ressourcen bereitstellen, die Sie verwenden möchten.

Bei der visuellen Designerfahrung ist Automation in Amazon CodeGuru Security integriert, sodass Sie Verstöße gegen Sicherheitsrichtlinien und Sicherheitslücken in Ihren Python-Skripten erkennen können. Um dieses Feature für `aws:executeScript`-Aktionen verwenden zu können, muss Ihre AWS Identity and Access Management (IAM)-Richtlinie die folgenden Berechtigungen enthalten:

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

```
            "Effect": "Allow",
            "Action": [
                "codeguru-security:CreateUploadUrl",
                "codeguru-security:CreateScan",
                "codeguru-security:GetScan",
                "codeguru-security:GetFindings"
            ]
        }
    ]
}
```

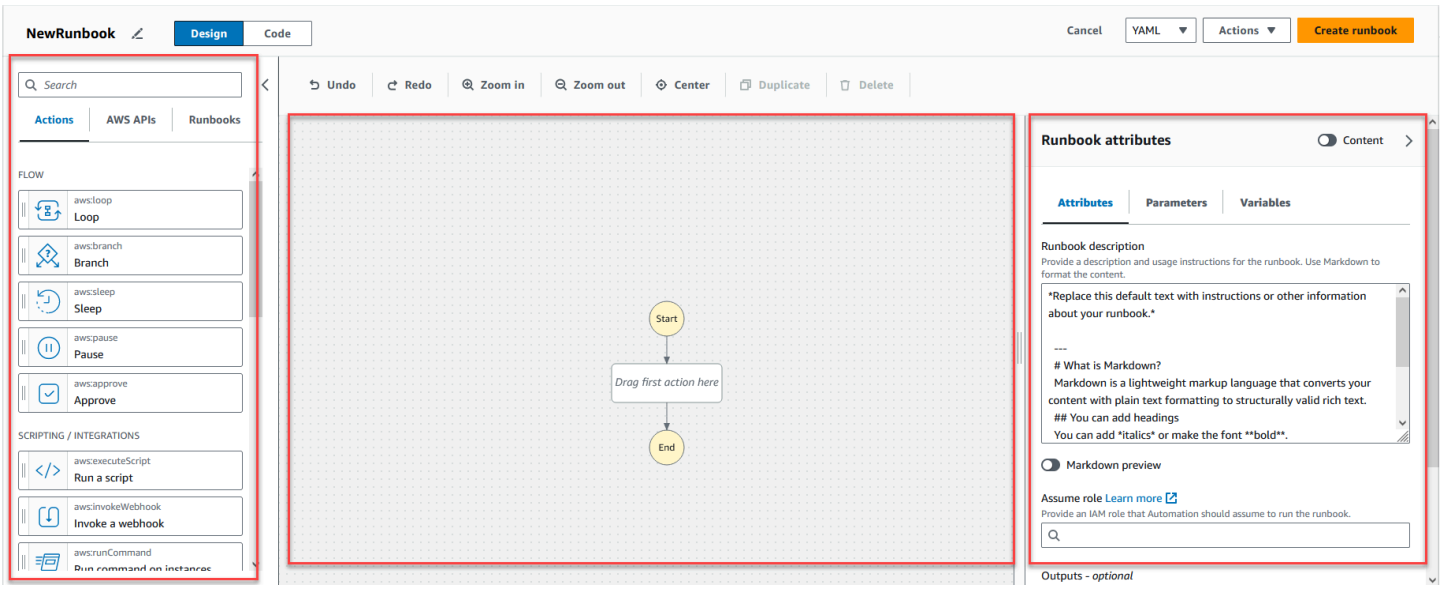
Themen

- Überblick über die Benutzeroberfläche für visuelle Designerfahrung
- Die visuelle Designerfahrung nutzen
- Konfigurieren von Eingaben und Ausgaben für Ihre Aktionen
- Fehlerbehandlung bei der visuellen Designerfahrung
- Tutorial: Ein Runbook mithilfe der visuellen Designerfahrung erstellen

Überblick über die Benutzeroberfläche für visuelle Designerfahrung

Die visuelle Designerfahrung für Systems Manager Automation ist ein visueller Workflow-Designer mit geringem Code-Aufwand, mit dem Sie Automation-Runbooks erstellen können.

Lernen Sie die visuelle Designerfahrung anhand eines Überblicks über die Komponenten der Benutzeroberfläche kennen:



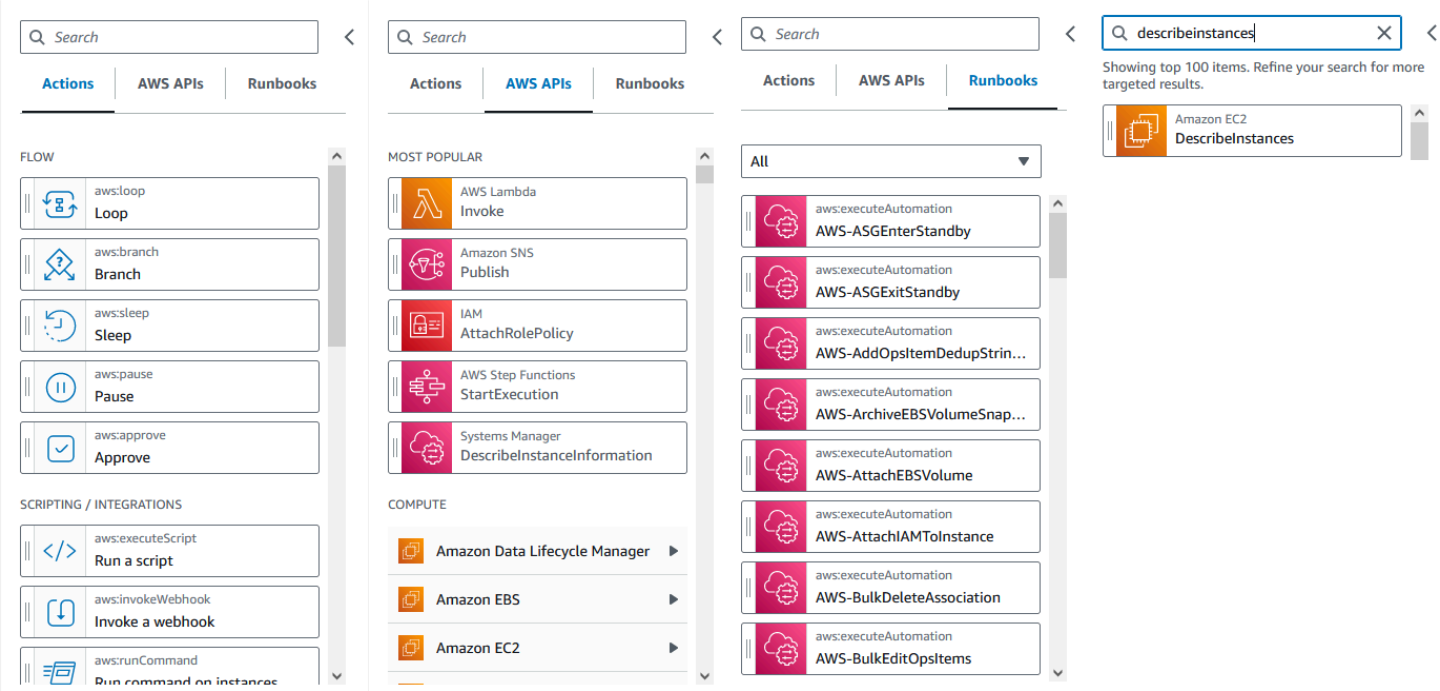
- Der Aktionsbrowser enthält die Registerkarten Aktionen, AWS -APIs und Runbooks.
- Auf der Arbeitsfläche können Sie Aktionen per Drag-and-Drop in Ihr Workflow-Diagramm ziehen, die Reihenfolge der Aktionen ändern und Aktionen auswählen, die konfiguriert oder angezeigt werden sollen.
- Im Formularfenster können Sie die Eigenschaften jeder Aktion, die Sie auf der Arbeitsfläche ausgewählt haben, anzeigen und bearbeiten. Wählen Sie den Schalter Inhalt, um die YAML- oder JSON-Daten für Ihr Runbook anzuzeigen, wobei die aktuell ausgewählte Aktion hervorgehoben ist.

Mit Informationslinks wird ein Fenster mit Kontextinformationen geöffnet, falls Sie Hilfe benötigen. Diese Bereiche enthalten auch Links zu verwandten Themen in der Systems-Manager-Automation-Dokumentation.

Aktionsbrowser

Im Aktionsbrowser können Sie Aktionen auswählen, die Sie per Drag-and-Drop in Ihr Workflow-Diagramm ziehen möchten. Mit dem Suchfeld oben im Aktionsbrowser können Sie nach allen Aktionen suchen. Der Aktionsbrowser enthält die folgenden Registerkarten:

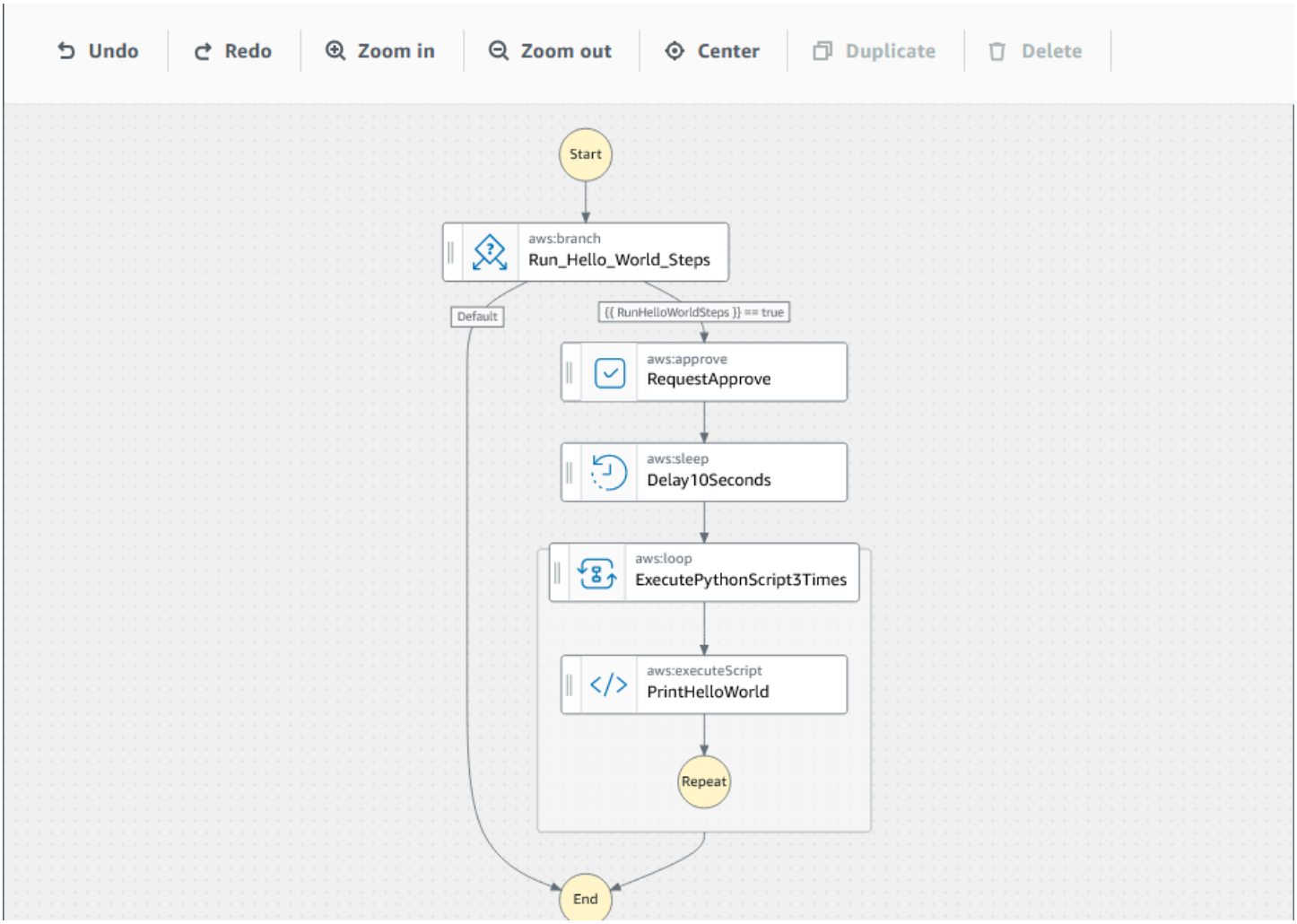
- Die Registerkarte Aktionen enthält eine Liste von Automatisierungs-Aktionen, die Sie per Drag-and-Drop in das Workflow-Diagramm Ihres Runbooks auf dem Workflow ziehen können.
- Auf der Registerkarte AWS APIs finden Sie eine Liste von AWS APIs, die Sie per Drag-and-Drop in das Workflow-Diagramm Ihres Runbooks im Zeichenbereich ziehen können.
- Die Registerkarte Runbooks enthält mehrere ready-to-usewiederverwendbare Runbooks als Bausteine, die Sie für eine Vielzahl von Anwendungsfällen verwenden können. Beispielsweise können Sie Runbooks verwenden, um allgemeine Behebungsaufgaben für Amazon-EC2-Instances in Ihrem Workflow durchzuführen, ohne dieselben Aktionen erneut erstellen zu müssen.



Leinwand

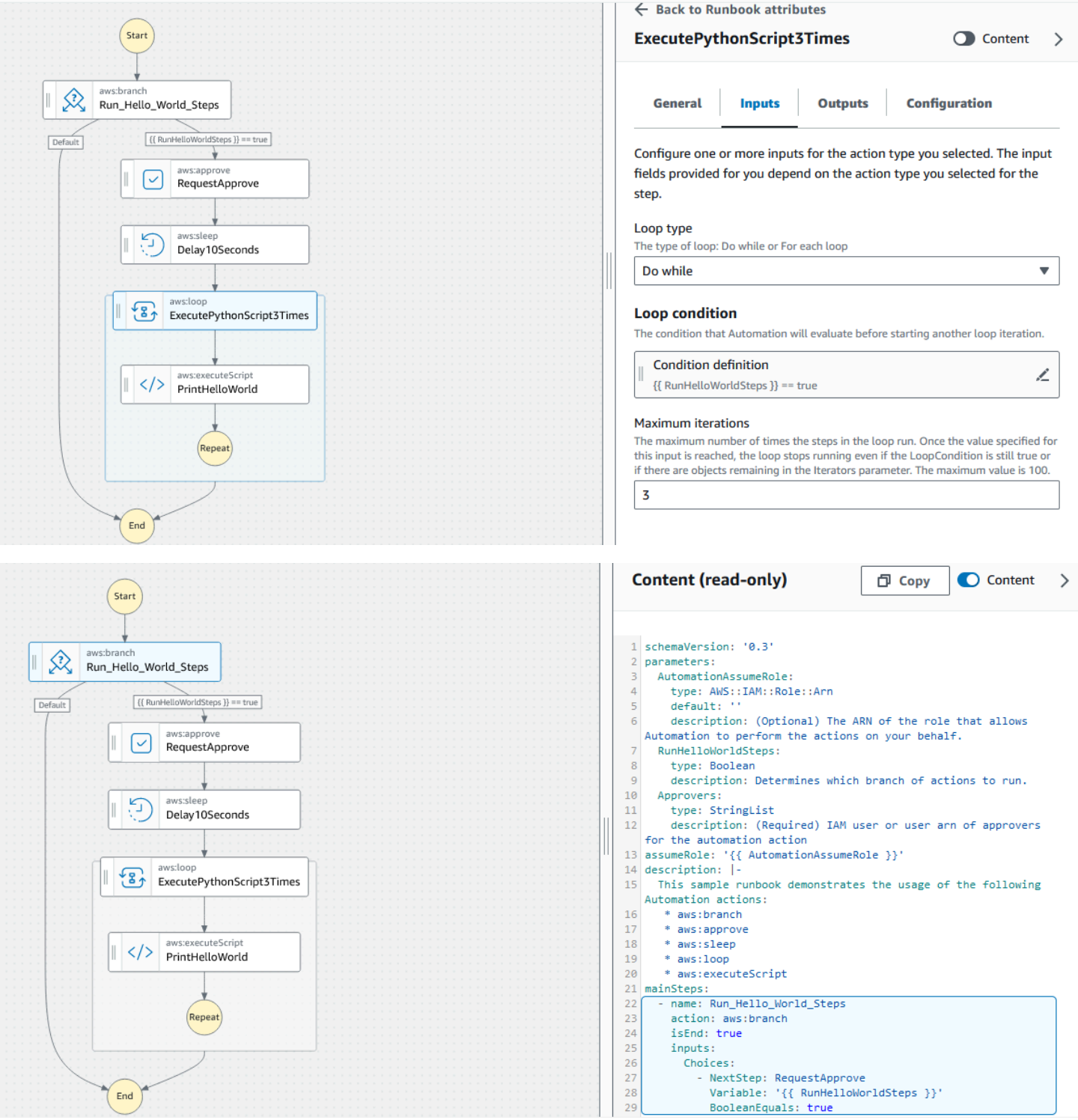
Nachdem Sie eine Aktion ausgewählt haben, die Sie zu Ihrer Automatisierung hinzufügen möchten, ziehen Sie sie auf den Workflow und legen Sie sie in Ihr Workflow-Diagramm ab. Sie können Aktionen auch per Drag-and-Drop an verschiedene Stellen im Workflow Ihres Runbooks verschieben. Wenn Ihr Workflow komplex ist, können Sie ihn möglicherweise nicht vollständig auf der Arbeitsfläche anzeigen. Verwenden Sie die Steuerelemente oben auf der Arbeitsfläche, um die Ansicht zu vergrößern oder zu verkleinern. Um verschiedene Teile eines Workflows anzuzeigen, können Sie das Workflow-Diagramm auf die Arbeitsfläche ziehen.

Ziehen Sie eine Aktion aus dem Browser Aktionen und legen Sie sie in das Workflow-Diagramm Ihres Runbooks ab. Eine Linie zeigt, wo sie in Ihrem Workflow platziert wird. Um die Reihenfolge einer Aktion zu ändern, können Sie sie an eine andere Stelle in Ihrem Workflow ziehen. Die neue Aktion wurde zu Ihrem Workflow hinzugefügt und ihr Code wird automatisch generiert.



Formular

Nachdem Sie Ihrem Runbook-Workflow eine Aktion hinzugefügt haben, können Sie sie so konfigurieren, dass sie Ihrem Anwendungsfall entspricht. Wählen Sie die Aktion aus, die Sie konfigurieren möchten, und die zugehörigen Parameter und Optionen werden im Formular-Bereich angezeigt. Sie können den YAML- oder JSON-Code auch sehen, indem Sie den Schalter Inhalt auswählen. Der Code, der von Ihnen ausgewählten Aktion zugeordnet ist, hervorgehoben.



Tastenkombinationen

Die visuelle Designerfahrung unterstützt die in der folgenden Tabelle aufgeführten Tastenkombinationen.

Tastaturkürzel	Verwenden Sie Tastaturkürzel, um den Workflow zu navigieren.
Rückgängig machen	Setzen Sie den letzten Vorgang rückgängig.
Wiederholen	Wiederholen Sie den letzten Vorgang.
Zentrieren	Zentrieren Sie den Workflow auf der Arbeitsfläche.
Auswahl aufheben	Heben Sie alle ausgewählten Zustände.
Löschen	Löschen Sie

Tastaturkürzel	alle ausgewählten Zustände.
Strg + D	Sie den ausgewählten Zustand.

Die visuelle Designerfahrung nutzen

Erfahren Sie, wie Sie Runbook-Workflows mithilfe der visuellen Designerfahrung erstellen, bearbeiten und ausführen. Sobald Ihr Workflow fertig ist, können Sie ihn speichern oder exportieren. Sie können die visuelle Designerfahrung auch für Rapid Prototyping nutzen.

Einen Runbook-Workflow erstellen

1. Melden Sie sich bei der Systems-Manager-Automation-Konsole an.
2. Wählen Sie Runbook erstellen.
3. Geben Sie im Feld Name einen Namen für Ihr Runbook ein, z. B. *MyNewRunbook*.
4. Wählen Sie neben der Option Design und Code das Stiftsymbol aus und geben Sie einen Namen für Ihr Runbook ein.

Sie können jetzt einen Workflow für Ihr neues Runbook entwerfen.

Ein Runbook entwerfen

Um einen Runbook-Workflow mithilfe der visuellen Designerfahrung zu entwerfen, ziehen Sie eine Automatisierungs-Aktion aus dem Browser Aktionen auf die Arbeitsfläche und platzieren sie an der gewünschten Stelle im Workflow Ihres Runbooks. Sie können Aktionen in Ihrem Workflow auch neu

anordnen, indem Sie sie an eine andere Position ziehen. Wenn Sie eine Aktion auf die Arbeitsfläche ziehen, wird an der Stelle, an der Sie die Aktion in Ihrem Workflow ablegen können, eine Linie angezeigt. Nachdem eine Aktion auf der Arbeitsfläche abgelegt wurde, wird ihr Code automatisch generiert und dem Inhalt Ihres Runbooks hinzugefügt.

Wenn Sie den Namen der Aktion kennen, die Sie hinzufügen möchten, verwenden Sie das Suchfeld oben im Browser Aktionen, um die Aktion zu finden.

Nachdem Sie eine Aktion auf der Arbeitsfläche abgelegt haben, konfigurieren Sie sie mithilfe des Fensters Formular auf der rechten Seite. Dieser Bereich enthält die Registerkarten Allgemein, Eingaben, Ausgaben und Konfiguration für jede Automatisierungs-Aktion oder API-Aktion, die Sie auf der Arbeitsfläche platzieren. Die Registerkarte Allgemein enthält beispielsweise die folgenden Abschnitte:

- Der Schrittname identifiziert den Schritt. Geben Sie einen eindeutigen Wert für den Schrittnamen an.
- Mithilfe der Beschreibung können Sie beschreiben, was die Aktion im Workflow Ihres Runbooks bewirkt.

Die Registerkarte Eingaben enthält Felder, die je nach Aktion variieren. Die `aws:executeScript`-Automatisierungs-Aktion enthält beispielsweise die folgenden Abschnitte:

- Die Laufzeit ist die Sprache, die zum Ausführen des bereitgestellten Skripts verwendet wird.
- Der Handler ist der Name Ihrer Funktion. Sie müssen sicherstellen, dass die im Handler definierte Funktion über zwei Parameter verfügt: `events` und `context`. Die PowerShell-Laufzeit unterstützt diesen Parameter nicht.
- Das Skript ist ein eingebettetes Skript, das während des Workflows ausgeführt werden soll.
- (Optional) Der Anhang ist für eigenständige Skripts oder ZIP-Dateien vorgesehen, die durch die Aktion aufgerufen werden können. Dieser Parameter muss für JSON-Runbooks angegeben werden.

Auf der Registerkarte Ausgaben können Sie die Werte angeben, die Sie aus einer Aktion ausgeben möchten. Sie können in späteren Aktionen Ihres Workflows auf Ausgabewerte verweisen oder zu Protokollierungszwecken Ausgaben aus Aktionen generieren. Nicht alle Aktionen verfügen über eine Registerkarte Ausgaben, da nicht alle Aktionen Ausgaben unterstützen. Die `aws:pause`-Aktion unterstützt beispielsweise keine Ausgaben. Für Aktionen, die Ausgaben unterstützen, besteht die Registerkarte Ausgaben aus den folgenden Abschnitten:

- Der Name ist der Name, der für den Ausgabewert verwendet werden soll. Sie können in späteren Aktionen Ihres Workflows auf Ausgaben verweisen.
- Der Selector ist eine JSONPath-Ausdruckszeichenfolge, die mit `"$."` beginnt und zum Auswählen einer oder mehrerer Komponenten innerhalb eines JSON-Elements verwendet wird.
- Der Typ ist der Datentyp für den Ausgabewert. Beispielsweise ein Datentyp `String` oder `Integer`.

Die Registerkarte Konfiguration enthält Eigenschaften und Optionen, die von allen Automatisierungs-Aktionen verwendet werden können. Die Aktion besteht aus folgenden Abschnitten:

- Die Eigenschaft Max. Versuche gibt an, wie oft eine Aktion wiederholt wird, wenn sie fehlschlägt.
- Die Eigenschaft Timeout in Sekunden gibt den Timeout-Wert für eine Aktion an.
- Die Eigenschaft Ist kritisch bestimmt, ob der Aktionsfehler die gesamte Automatisierung stoppt.
- Die Eigenschaft Nächster Schritt bestimmt, welche Aktion die Automatisierung als Nächstes im Runbook ausführt.
- Die Eigenschaft Schlägt fehl bestimmt, welche Aktion die Automatisierung im Runbook als Nächstes ausführt, falls die Aktion fehlschlägt.
- Die Eigenschaft Wird abgebrochen bestimmt, welche Aktion die Automatisierung als Nächstes im Runbook ausführt, wenn die Aktion von einem Benutzer abgebrochen wird.

Um eine Aktion zu löschen, können Sie die Rücktaste, die Werkzeugleiste über der Arbeitsfläche, verwenden oder mit der rechten Maustaste klicken und Aktion löschen wählen.

Wenn Ihr Workflow wächst, passt er möglicherweise nicht in die Arbeitsfläche. Um den Workflow an die Arbeitsfläche anzupassen, führen Sie eine der folgenden Optionen aus:

- Verwenden Sie die Steuerelemente an den Seitenbereichen, um die Größe der Bedienfelder zu ändern oder sie zu schließen.
- Verwenden Sie die Werkzeugleiste oben auf der Leinwand, um das Workflow-Diagramm zu vergrößern oder zu verkleinern.

Ihr Runbook aktualisieren

Sie können einen vorhandenen Runbook-Workflow aktualisieren, indem Sie eine neue Version Ihres Runbook erstellen. Aktualisierungen Ihrer Runbooks können mithilfe der visuellen Designerfahrung

oder durch direkte Bearbeitung des Codes vorgenommen werden. Um ein vorhandenes Runbook zu aktualisieren, gehen Sie wie folgt vor:

1. Melden Sie sich bei der Systems-Manager-Automation-Konsole an.
2. Wählen Sie das Runbooks, das Sie aktualisieren möchten.
3. Wählen Sie Create new version (Neue Version erstellen)aus.
4. Die visuelle Designerfahrung besteht aus zwei Bereichen: einem Codebereich und einem visuellen Workflow-Bereich. Wählen Sie im visuellen Workflow-Bereich die Option Design aus, um Ihren Workflow mit der visuellen Designerfahrung zu bearbeiten. Wenn Sie fertig sind, wählen Sie Neue Version erstellen aus, um Ihre Änderungen zu speichern und den Vorgang zu beenden.
5. (Optional) Verwenden Sie den Codebereich, um den Runbook-Inhalt in YAML oder JSON zu bearbeiten.

Ihr Runbook exportieren

Gehen Sie wie folgt vor, um den Workflow-YAML- oder JSON-Code Ihres Runbooks sowie ein Diagramm Ihres Workflows zu exportieren:

1. Wählen Sie Ihr Runbook in der Dokumentenkonsole aus.
2. Wählen Sie Create new version (Neue Version erstellen)aus.
3. Wählen Sie in der Dropdownliste Aktionen aus, ob Sie das Diagramm oder das Runbook exportieren möchten und welches Format Sie bevorzugen.

Konfigurieren von Eingaben und Ausgaben für Ihre Aktionen

Jede Automatisierungs-Aktion reagiert auf der Grundlage von Eingaben, die sie empfängt. In den meisten Fällen geben Sie die Ausgabe dann an die nachfolgenden Aktionen weiter. In der visuellen Designerfahrung können Sie die Eingabe- und Ausgabedaten einer Aktion auf den Registerkarten Eingaben und Ausgaben des Formularfensters konfigurieren.

Weitere Informationen zum Definieren und Verwenden von Ausgaben für Automatisierungs-Aktionen finden Sie unter Verwenden von Aktionsausgaben als Eingaben.

Eingabedaten für eine Aktion angeben

Jede Automatisierungs-Aktion hat eine oder mehrere Eingaben, für die Sie einen Wert angeben müssen. Der Wert, den Sie für die Eingabe einer Aktion angeben, wird durch den Datentyp und

das Format bestimmt, die von der Aktion akzeptiert werden. Für die `aws:sleep`-Aktionen ist beispielsweise ein Zeichenfolgenwert im ISO-8601-Format für die `Duration`-Eingabe erforderlich.

Im Allgemeinen verwenden Sie im Workflow Ihres Runbooks Aktionen, die Ausgaben zurückgeben, die Sie in nachfolgenden Aktionen verwenden möchten. Es ist wichtig, dass Sie sicherstellen, dass Ihre Eingabewerte korrekt sind, um Fehler im Workflow Ihres Runbooks zu vermeiden. Eingabewerte sind auch deshalb wichtig, weil sie bestimmen, ob die Aktion die erwartete Ausgabe zurückgibt. Wenn Sie die `aws:executeAwsApi`-Aktion verwenden, möchten Sie beispielsweise sicherstellen, dass Sie den richtigen Wert für den API-Vorgang angeben.

Die Ausgabedaten für eine Aktion definieren

Einige Automatisierungs-Aktionen geben eine Ausgabe zurück, nachdem sie ihre definierten Operationen ausgeführt haben. Aktionen, die Ausgaben zurückgeben, haben entweder vordefinierte Ausgaben oder ermöglichen es Ihnen, die Ausgaben selbst zu definieren. Die `aws:createImage`-Aktion hat beispielsweise vordefinierte Ausgaben, die `ImageId` und `ImageState` zurückgeben. Im Vergleich dazu können Sie mit der `aws:executeAwsApi`-Aktion die Ausgaben definieren, die Sie von der angegebenen API-Operation erwarten. Daher können Sie einen oder mehrere Werte aus einer einzelnen API-Operation zurückgeben, um sie in nachfolgenden Aktionen zu verwenden.

Um Ihre eigenen Ausgaben für eine Automatisierungs-Aktion zu definieren, müssen Sie einen Namen der Ausgabe, den Datentyp und den Ausgabewert angeben. Um die `aws:executeAwsApi`-Aktion weiterhin als Beispiel zu verwenden, nehmen wir an, Sie rufen den `DescribeInstances`-API-Vorgang von Amazon EC2 aus auf. In diesem Beispiel möchten Sie die Daten einer Amazon-EC2-Instance zurückgeben oder ausgeben und den `State`-Workflow Ihres Runbooks auf der Grundlage der Ausgabe verzweigen. Sie geben der Ausgabe **`InstanceState`** einen Namen und verwenden den **`String`**-Datentyp.

Das Verfahren zur Definition des tatsächlichen Werts der Ausgabe unterscheidet sich je nach Aktion. Wenn Sie beispielsweise die `aws:executeScript`-Aktion verwenden, müssen Sie `return`-Anweisungen in Ihren Funktionen verwenden, um Daten für Ihre Ausgaben bereitzustellen. Bei anderen Aktionen wie `aws:executeAwsApi`, `aws:waitForAwsResourceProperty`, und `aws:assertAwsResourceProperty` ist `Selector` erforderlich. `Selector` oder `PropertySelector`, wie sich einige Aktionen darauf beziehen, ist eine JSONPath-Zeichenfolge, die verwendet wird, um die JSON-Antwort aus einer API-Operation zu verarbeiten. Es ist wichtig zu verstehen, wie das JSON-Antwortobjekt aus einer API-Operation strukturiert ist, damit Sie den richtigen Wert für Ihre Ausgabe auswählen können. Sehen Sie sich das folgende Beispiel für eine JSON-Antwort an, indem Sie die zuvor erwähnte `DescribeInstances`-API-Operation verwenden:

```
{
  "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",
```

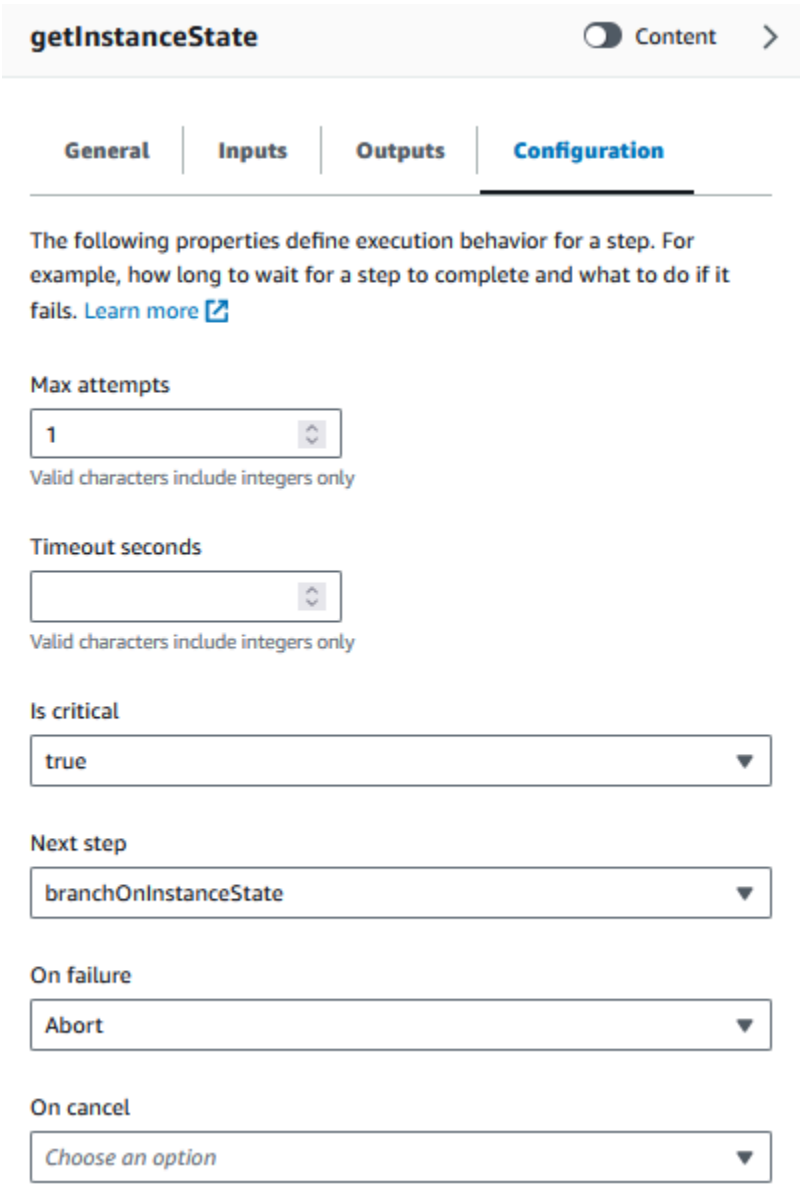
```
                "deleteOnTermination": true
              },
              "association": {
                "publicIp": "54.194.252.215",
                "publicDnsName": "ec2-54-194-252-215.eu-west-1.compute.amazonaws.com",
                "ipOwnerId": "amazon"
              },
              "privateIpAddressesSet": {
                "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
          }
        }
      }
    }
  }
}
```

Im JSON-Antwortobjekt `State` ist die Instance in einem `Instances`-Objekt verschachtelt, das im `Reservations`-Objekt verschachtelt ist. Um den Wert der Instance `State` zurückzugeben, verwenden Sie die folgende Zeichenfolge für `Selector`, damit der Wert in unserer Ausgabe verwendet werden kann: **`$.Reservations[0].Instances[0].State.Name`**.

Um in nachfolgenden Aktionen des Workflows Ihres Runbooks auf einen Ausgabewert zu verweisen, wird das folgende Format verwendet: `{{ StepName.NameOfOutput }}`. Zum Beispiel **`{{ GetInstanceState.InstanceState }}`**. In der visuellen Designerfahrung können Sie mithilfe der Dropdownliste für die Eingabe Ausgabewerte auswählen, die in nachfolgenden Aktionen verwendet werden sollen. Wenn Sie Ausgaben in nachfolgenden Aktionen verwenden, muss der Datentyp der Ausgabe mit dem Datentyp für die Eingabe übereinstimmen. In diesem Beispiel ist die `InstanceState`-Ausgabe `String`. Um den Wert in der Eingabe einer nachfolgenden Aktion zu verwenden, muss die Eingabe daher `String` akzeptieren.

Fehlerbehandlung bei der visuellen Designerfahrung

Wenn eine Aktion einen Fehler meldet, stoppt Automation standardmäßig den Workflow des Runbooks vollständig. Das liegt daran, dass der Standardwert für die `onFailure`-Eigenschaft für alle Aktionen `Abort` ist. Sie können konfigurieren, wie Automation mit Fehlern im Workflow Ihres Runbooks umgeht. Auch wenn Sie die Fehlerbehandlung konfiguriert haben, können einige Fehler dennoch dazu führen, dass eine Automatisierung fehlschlägt. Weitere Informationen finden Sie unter Fehlerbehebung für Systems Manager Automation.. In der visuellen Designerfahrung konfigurieren Sie die Fehlerbehandlung im Bereich Konfiguration.



Bei einem Fehler die Aktion erneut versuchen

Um eine Aktion im Falle eines Fehlers erneut zu versuchen, geben Sie einen Wert für die Eigenschaft Max. Versuche an. Der Standardwert lautet 1. Wenn Sie einen Wert größer als 1 angeben, gilt die Aktion erst dann als fehlgeschlagen, wenn alle Wiederholungsversuche fehlgeschlagen sind.

Timeouts

Sie können ein Timeout für Aktionen konfigurieren, um festzulegen, wie viele Sekunden Ihre Aktion maximal ausgeführt werden kann, bevor sie fehlschlägt. Um ein Timeout zu konfigurieren, geben Sie in der Eigenschaft Timeout-Sekunden die Anzahl der Sekunden ein, die Ihre Aktion warten soll, bis die Aktion fehlschlägt. Wenn das Timeout erreicht ist und die Aktion einen Wert von `Max attempts`

hat, der größer als 1 ist, gilt der Schritt erst dann als Timeout, wenn die Wiederholungsversuche abgeschlossen sind.

Fehlgeschlagene Aktionen

Wenn eine Aktion fehlschlägt, stoppt Automation standardmäßig den Workflow des Runbooks vollständig. Sie können dieses Verhalten ändern, indem Sie einen alternativen Wert für die Eigenschaft Bei einem Ausfall der Aktionen in Ihrem Runbook angeben. Wenn Sie möchten, dass der Workflow mit dem nächsten Schritt im Runbook fortfährt, wählen Sie Weiter aus. Wenn der Workflow zu einem anderen nachfolgenden Schritt im Runbook springen soll, wählen Sie Schritt aus und geben Sie dann den Namen des Schritts ein.

Abgebrochene Aktionen

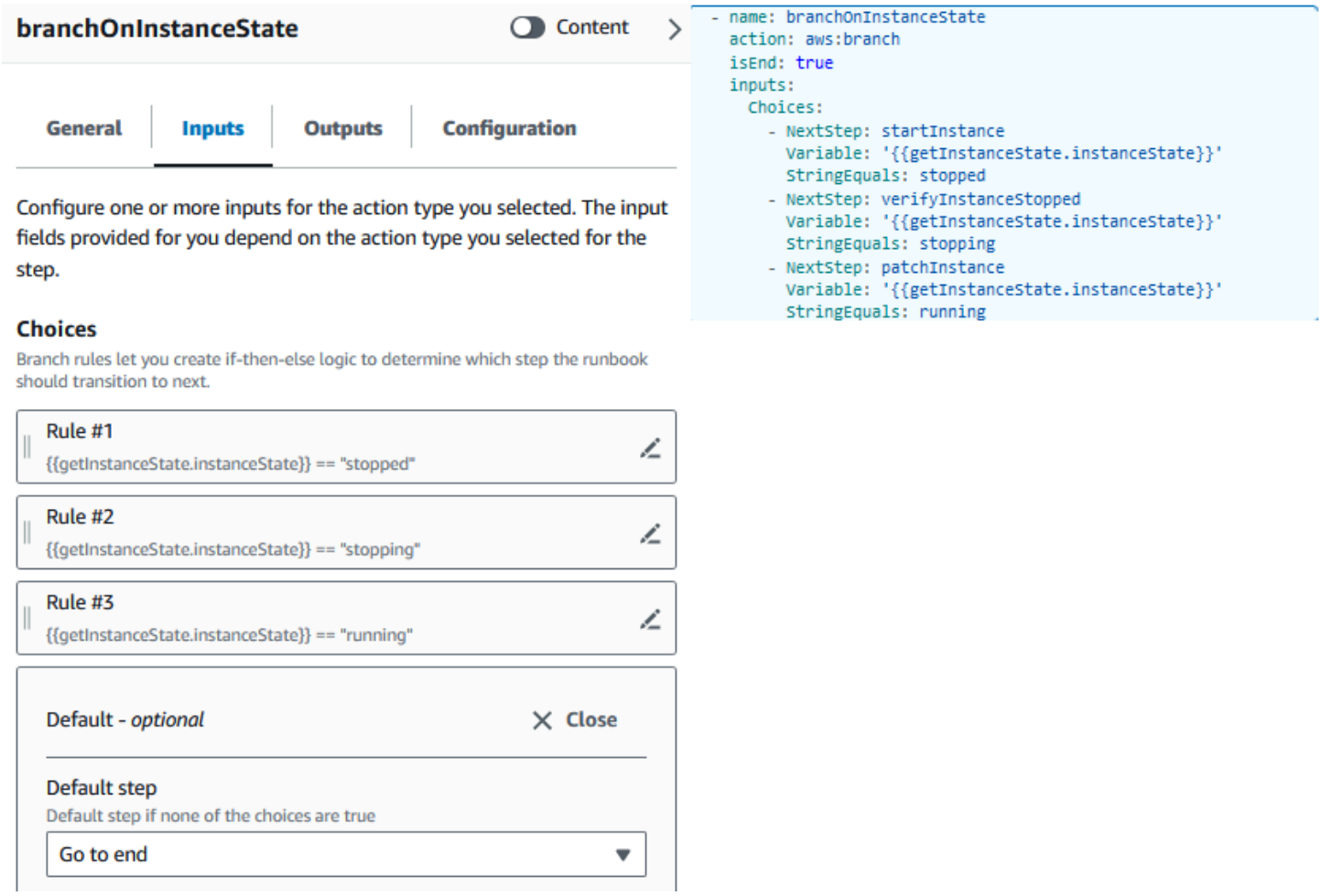
Wenn eine Aktion von einem Benutzer abgebrochen wird, stoppt Automation standardmäßig den Workflow des Runbooks vollständig. Sie können dieses Verhalten ändern, indem Sie einen alternativen Wert für die Eigenschaft Bei Abbruch der Aktionen in Ihrem Runbook angeben. Wenn der Workflow zu einem anderen nachfolgenden Schritt im Runbook springen soll, wählen Sie Schritt aus und geben Sie dann den Namen des Schritts ein.

Kritische Aktionen

Sie können eine Aktion als kritisch kennzeichnen, was bedeutet, dass sie den allgemeinen Berichtsstatus Ihrer Automatisierung bestimmt. Wenn ein Schritt mit dieser Bezeichnung fehlschlägt, meldet Automation den Endstatus als `Failed` unabhängig vom Erfolg anderer Aktionen. Um eine Aktion als kritisch zu konfigurieren, belassen Sie den Standardwert Richtig für die Eigenschaft Ist kritisch.

Aktionen beenden

Die Eigenschaft Ist am Ende stoppt eine Automatisierung am Ende der angegebenen Aktion. Der Standardwert dieser Eigenschaft ist `false`. Wenn Sie diese Eigenschaft für eine Aktion konfigurieren, stoppt die Automatisierung unabhängig davon, ob die Aktion erfolgreich ist oder fehlschlägt. Diese Eigenschaft wird am häufigsten bei `aws:branch`-Aktionen verwendet, um unerwartete oder undefinierte Eingabewerte zu verarbeiten. Das folgende Beispiel zeigt ein Runbook, das einen Instance-Status von entweder `running`, `stopping` oder `stopped` erwartet. Wenn sich eine Instance in einem anderen Status befindet, wird die Automatisierung beendet.



Tutorial: Ein Runbook mithilfe der visuellen Designerfahrung erstellen

In diesem Tutorial lernen Sie die Grundlagen für die Arbeit mit der visuellen Designerfahrung von Systems Manager Automation. In der visuellen Designerfahrung können Sie ein Runbook erstellen, das mehrere Aktionen verwendet. Sie verwenden das Drag-and-Drop-Feature, um Aktionen auf der Arbeitsfläche anzuordnen. Sie suchen auch nach diesen Aktionen, wählen sie aus und konfigurieren sie. Anschließend können Sie den automatisch generierten YAML-Code für den Workflow Ihres Runbooks anzeigen, die visuelle Designerfahrung beenden, das Runbook ausführen und die Ausführungsdetails überprüfen.

In diesem Tutorial erfahren Sie auch, wie Sie das Runbook aktualisieren und die neue Version anzeigen. Am Ende des Tutorials führen Sie einen Bereinigungsschritt durch und löschen Ihr Runbook.

Nachdem Sie dieses Tutorial abgeschlossen haben, wissen Sie, wie Sie mithilfe der visuellen Designerfahrung ein Runbook erstellen können. Sie werden auch wissen, wie Sie Ihr Runbook aktualisieren, ausführen und löschen.

> **Note**
>
> Bevor Sie mit diesem Tutorial beginnen, stellen Sie sicher, dass Sie Einrichten der Automatisierung abschließen.

Themen

- Schritt 1: Zur visuellen Designerfahrung navigieren
- Schritt 2: Einen Workflow erstellen
- Schritt 3: Den automatisch generierten Code überprüfen
- Schritt 4: Ihr neues Runbook ausführen
- Schritt 5: Bereinigen

Schritt 1: Zur visuellen Designerfahrung navigieren

1. Melden Sie sich bei der Systems-Manager-Automation-Konsole an.
2. Wählen Sie Automation-Runbook erstellen.

Schritt 2: Einen Workflow erstellen

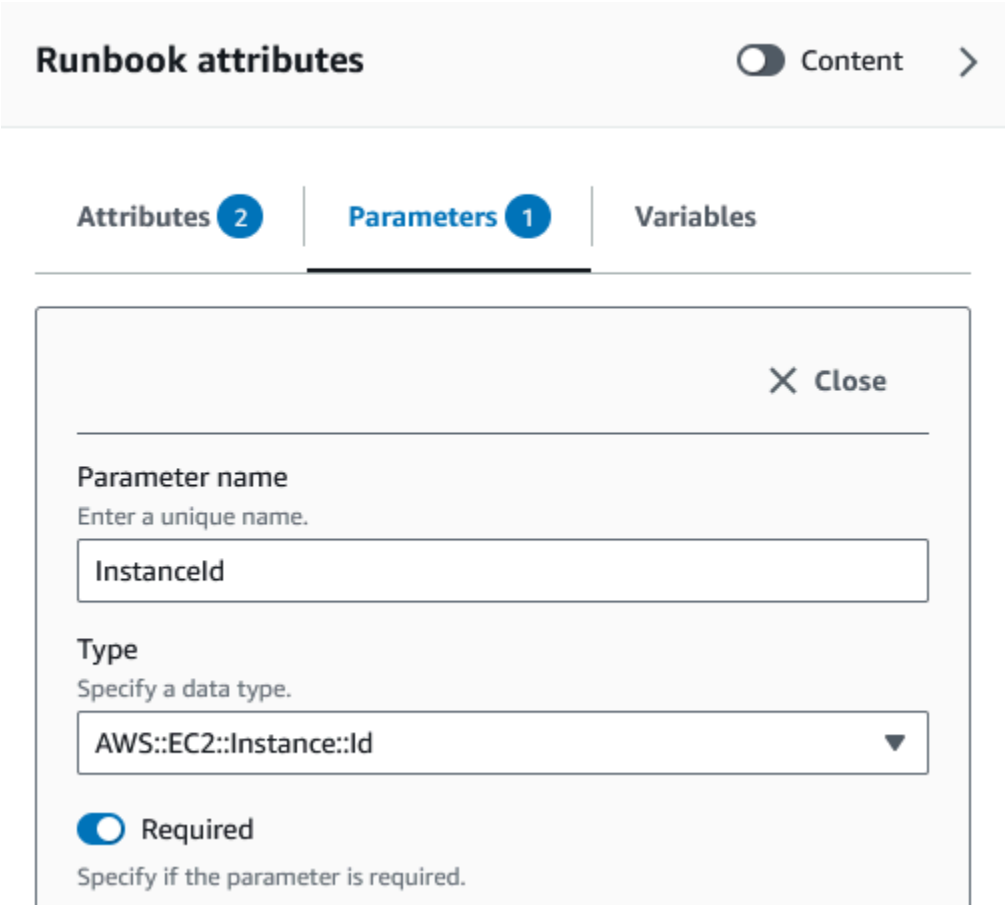
In der visuellen Designerfahrung ist ein Workflow eine grafische Darstellung Ihres Runbooks auf der Arbeitsfläche. Sie können die visuelle Designerfahrung verwenden, um die einzelnen Aktionen Ihres Runbooks zu definieren, zu konfigurieren und zu untersuchen.

So erstellen Sie ein Workflow

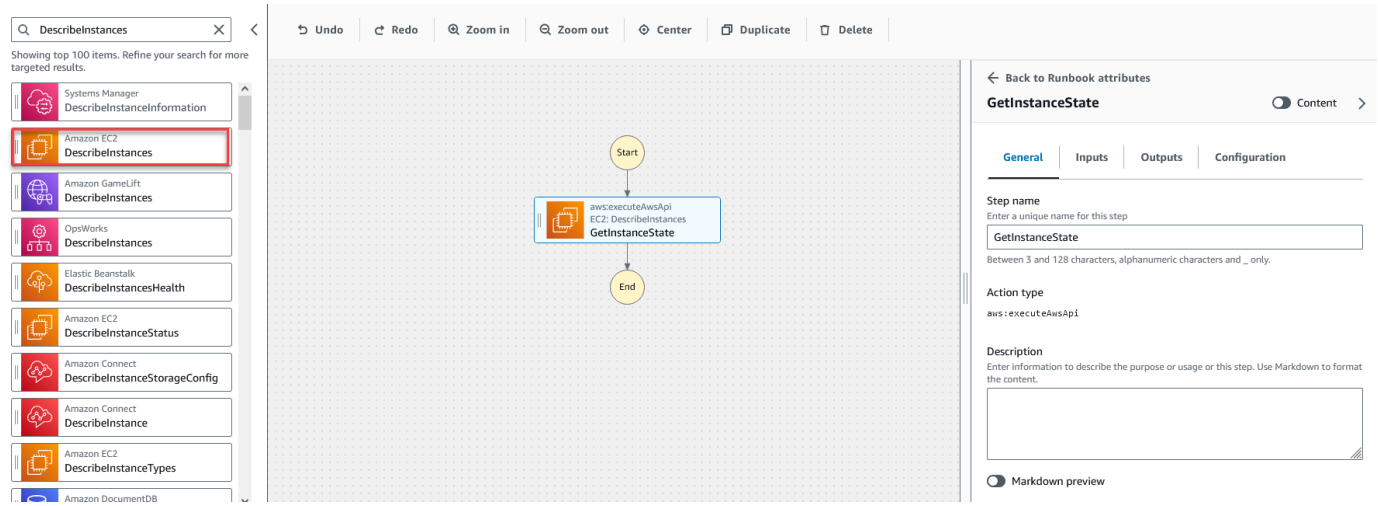
1. Wählen Sie neben der Option Design und Code das Stiftsymbol aus und geben Sie einen Namen für Ihr Runbook ein. Geben Sie für dieses Tutorial **`VisualDesignExperienceTutorial`** ein.

2. Erweitern Sie im Bereich Dokumentattribute des Bedienfelds Formular die Dropdownliste Eingabeparameter und wählen Sie Parameter hinzufügen aus.
   a. Geben Sie im Feld Parametername **`InstanceId`** ein.
   b. Wählen Sie in der Dropdownliste Typ die Option AWS::EC2::Instance.

c. Wählen Sie den Schalter Erforderlich aus.



3. Geben Sie im AWS -API-Browser **`DescribeInstances`** in die Suchleiste ein.
4. Ziehen Sie eine Amazon EC2 — DescribeInstances Aktion auf die leere Leinwand.
5. Geben Sie für Schrittname einen Wert ein. Verwenden Sie in diesem Tutorial **`GetInstanceState`** als Namen.



a. Erweitern Sie das Dropdown-Menü Zusätzliche Eingaben und geben Sie im Feld Eingabename **`InstanceIds`** ein.

b. Wählen Sie die Registerkarte Eingaben.

c. Wählen Sie im Feld Eingabewert die **`InstanceId`** Dokumenteingabe aus. Dies verweist auf den Wert des Eingabeparameters, den Sie zu Beginn des Verfahrens erstellt haben. Da die InstanceIdsEingabe für die `DescribeInstances` Aktion `StringList` Werte akzeptiert, müssen Sie die InstanceIdEingabe in eckige Klammern setzen. Das YAML für den Eingabewert sollte den folgenden Werten entsprechen:**`['{{ InstanceId }}']`**.

d. Wählen Sie auf der Registerkarte Ausgaben die Option Ausgabe hinzufügen aus und geben Sie **`InstanceState`** in das Feld Name ein.

e. Geben Sie **`$.Reservations[0].Instances[0].State.Name`** im Feld Auswahl ein.

f. Wählen Sie in der Dropdownliste Typ die Option Zeichenfolge aus.

6. Ziehen Sie eine Branch-Aktion aus dem Aktionsbrowser und legen Sie sie unter dem **`GetInstanceState`**-Schritt ab.

7. Geben Sie für Schrittname einen Wert ein. Verwenden Sie in diesem Tutorial den Namen **`BranchOnInstanceState`**.

   Um die Branch-Logik zu definieren, führen Sie die folgenden Schritte aus:

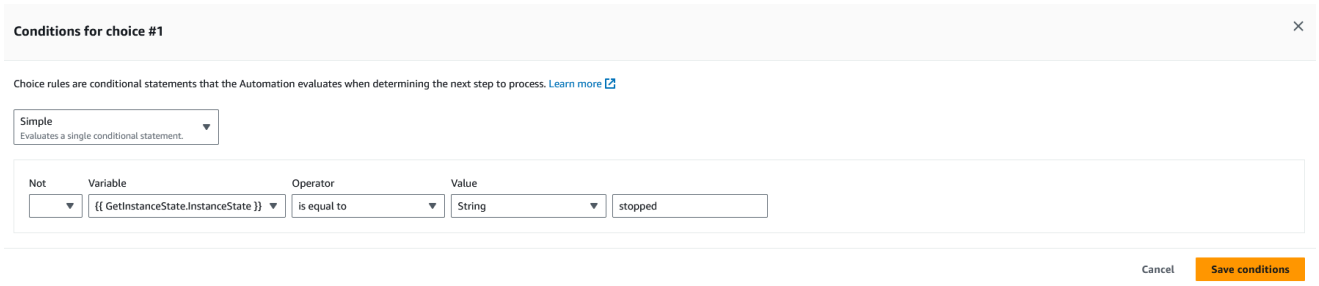
   a. Wählen Sie den **`Branch`**-Status auf der Arbeitsfläche aus. Wählen Sie dann unter Eingaben und Wahlmöglichkeiten das Stiftsymbol aus, um Regel #1 zu bearbeiten.

   b. Wählen Sie Bedingungen hinzufügen.

c. Wählen Sie im Dialogfeld Bedingungen für Regel #1 die **`GetInstanceState.InstanceState`**-Schrittausgabe aus der Dropdownliste Variable aus.

d. Wählen Sie für Operator die Option Ist gleich aus.

e. Wählen Sie als Wert Zeichenfolge aus der Dropdown-Liste aus. Geben Sie **`stopped`** ein.



f. Wählen Sie Bedingungen speichern aus.

g. Wählen Sie Neue Auswahlregel hinzufügen aus.

h. Wählen Sie Bedingungen hinzufügen für Regel #2.

i. Wählen Sie im Dialogfeld Bedingungen für Regel #2 die **`GetInstanceState.InstanceState`**-Schrittausgabe aus der Dropdownliste Variable aus.

j. Wählen Sie für Operator die Option Ist gleich aus.

k. Wählen Sie als Wert Zeichenfolge aus der Dropdown-Liste aus. Geben Sie **`stopping`** ein.

l. Wählen Sie Bedingungen speichern aus.

m. Wählen Sie Neue Auswahlregel hinzufügen aus.

n. Wählen Sie für Regel #3 Bedingungen hinzufügen.

o. Wählen Sie im Dialogfeld Bedingungen für Regel #3 die **`GetInstanceState.InstanceState`**-Schrittausgabe aus der Dropdownliste Variable aus.

p. Wählen Sie für Operator die Option Ist gleich aus.

q. Wählen Sie als Wert Zeichenfolge aus der Dropdown-Liste aus. Geben Sie **`running`** ein.

r. Wählen Sie Bedingungen speichern aus.

s. Wählen Sie in der Standardregel für den Standardschritt die Option Gehe zum Ende aus.

8. Ziehen Sie eine Aktion „Instanzstatus ändern“ in das leere Feld „Aktion hierher ziehen“ unter {{ GetInstanceState. InstanceState }} == Zustand „gestoppt“.

a. Geben Sie als Schrittnamen **`StartInstance`** ein.

b. Wählen Sie auf der Registerkarte „Eingaben“ unter „Instanz-IDs“ den Eingabewert für InstanceIddas Dokument aus der Dropdownliste aus.

c. Geben Sie für den gewünschten Status **`running`** an.

9. Ziehen Sie eine Aktion „Auf AWS Ressource warten“ in das leere Feld Aktion hierher ziehen unter dem Feld {{ GetInstanceState. InstanceState }} == Zustand „stoppt“.

10. Geben Sie für Schrittname einen Wert ein. Verwenden Sie in diesem Tutorial den Namen **`WaitForInstanceStop`**.

    a. Wählen Sie für das Feld Service Amazon EC2 aus.

    b. Wählen Sie für das API-Feld DescribeInstances.

    c. Geben Sie für das Feld Eigenschaftsauswahl den Wert **`$.Reservations[0].Instances[0].State.Name`** ein.

    d. Geben **`["stopped"]`** Sie für den Parameter Gewünschte Werte ein.

    e. Wählen Sie auf der Registerkarte „Konfiguration“ der WaitForInstanceStopAktion die Option „Nächster Schritt“ StartInstanceaus.

11. Ziehen Sie die Aktion „Befehl auf Instanzen ausführen“ in das leere Feld Aktion hierher ziehen unter {{ GetInstanceState. InstanceState }} == Zustand „läuft“.

12. Geben Sie als Schrittnamen **`SayHello`** ein.

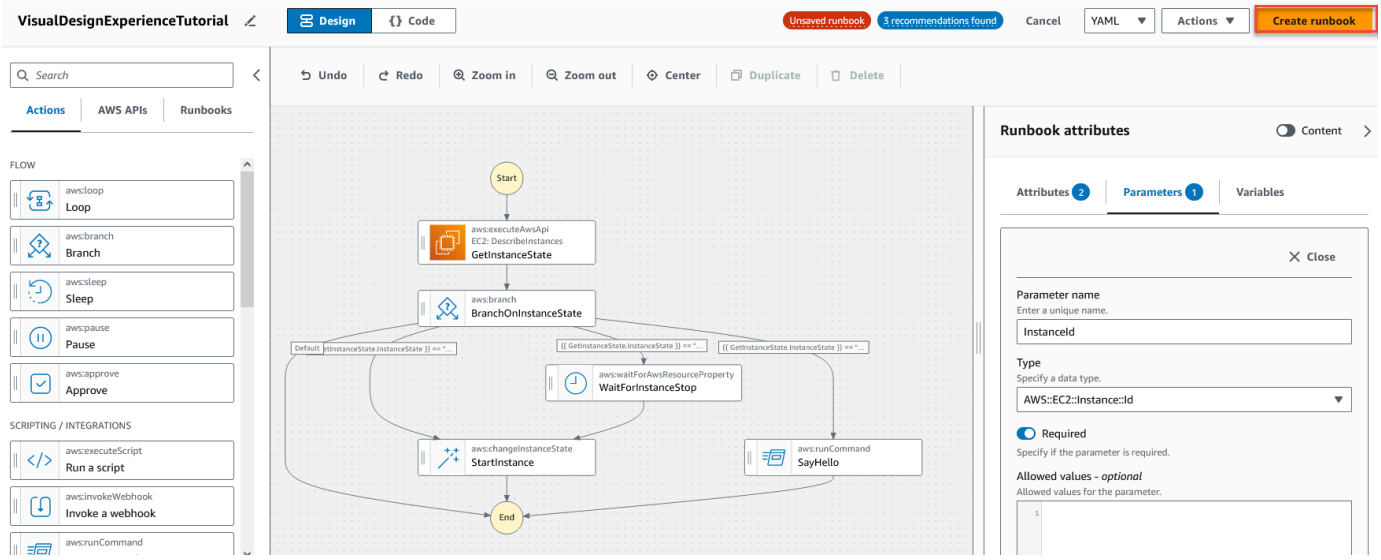
    a. Geben Sie auf der Registerkarte Eingaben den Wert **`AWS-RunShellScript`** für den Parameter Dokumentname ein.

    b. Wählen Sie für InstanceIdsden InstanceIdDokumenteingabewert aus der Dropdownliste aus.

    c. Erweitern Sie das Dropdownmenü Zusätzliche Eingaben und wählen Sie im Dropdownmenü Eingabename die Option Parameter aus.

    d. Geben Sie im Feld Eingabewert **`{"commands": "echo 'Hello World'"}`** ein.

13. Prüfen Sie das fertige Runbook auf der Arbeitsfläche und wählen Sie Runbook erstellen aus, um das Tutorial-Runbook zu speichern.



Schritt 3: Den automatisch generierten Code überprüfen

Wenn Sie Aktionen aus dem Browser Aktion auf die Arbeitsfläche ziehen und dort ablegen, erstellt die visuelle Designerfahrung automatisch den YAML- oder JSON-Inhalt Ihres Runbooks in Echtzeit. Sie können diesen Code anzeigen und bearbeiten. Um den automatisch generierten Code anzuzeigen, wählen Sie Code für die Umschalter Design und Code aus.

Schritt 4: Ihr neues Runbook ausführen

Nachdem Sie Ihr Runbook erstellt haben, können Sie die Automatisierung ausführen.

So führen Sie Ihr neues Automation-Runbook aus

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Automation (Automatisierung) und Execute automation (Automatisierung ausführen) aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Mit mir geteilt-Registerkarte. Um alle Runbooks anzuzeigen, wählen Sie die Alle Dokumente-Registerkarte.

> **Note**
>
> Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Document details (Dokument-Details), ob Document version (Dokumentversion) auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.

5. Wählen Sie Weiter aus.
6. Klicken Sie auf der Seite Automation-Runbook ausführen auf Einfache Ausführung.
7. Geben Sie im Abschnitt Input Parameters (Eingabeparameter) die erforderlichen Eingaben an. Optional können Sie eine IAM-Servicerolle aus der AutomationAssumeRoleListe auswählen.
8. (Optional) Wählen Sie einen CloudWatch Amazon-Alarm aus, der auf Ihre Automatisierung zur Überwachung angewendet werden soll. Um Ihrer Automatisierung einen CloudWatch Alarm zuzuweisen, muss der IAM-Principal, der die Automatisierung startet, über die Genehmigung für die `iam:createServiceLinkedRole` Aktion verfügen. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatchAmazon-Alarme verwenden. Wenn die Automatisierung gestoppt wird, wird Ihr Alarm aktiviert. Wenn Sie AWS CloudTrail verwenden, sehen Sie den API-Aufruf in Ihrem Trail.
9. Wählen Sie Execute (Ausführen).

Schritt 5: Bereinigen

So löschen Sie Ihr Runbook

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie die Registerkarte In meinem Besitz aus.
4. Suchen Sie das VisualDesignExperienceTutorialRunbook.
5. Wählen Sie die Schaltfläche auf der Dokumentkartenseite aus und wählen Sie dann in der Dropdownliste Aktionen die Option Dokument löschen aus.

Erstellen von Automation-Runbooks

Jedes Runbook in Automation, eine Fähigkeit von AWS Systems Manager, definiert eine Automatisierung. Automatisierungs-Runbooks definieren die Aktionen, die während einer Automatisierung ausgeführt werden. Im Runbook-Inhalt definieren Sie die Eingabeparameter, Ausgaben und Aktionen, die Systems Manager für Ihre verwalteten Instanzen und AWS Ressourcen ausführt.

Automatisierung umfasst mehrere vordefinierte Runbooks, die Sie verwenden können, um allgemeine Aufgaben wie das Neustarten einer oder mehrerer Amazon Elastic Compute Cloud (Amazon EC2)-Instances oder das Erstellen eines Amazon Machine Image (AMI) auszuführen. Ihre Anwendungsfälle können jedoch über die Funktionen der vordefinierten Runbooks hinausgehen. In diesem Fall können Sie eigene Runbooks erstellen und an Ihre Bedürfnisse anpassen.

Ein Runbook besteht aus Automatisierungsaktionen, Parametern für diese Aktionen und Eingabeparametern, die Sie angeben. Der Inhalt eines Runbooks wird entweder in YAML oder JSON geschrieben. Wenn Sie weder mit YAML noch mit JSON vertraut sind, empfehlen wir, den Visual Designer zu verwenden oder mehr über eine der beiden Auszeichnungssprachen zu lernen, bevor Sie versuchen, Ihr eigenes Runbook zu erstellen. Weitere Informationen zum Visual Designer finden Sie unter. Visuelle Designerfahrung für Automation-Runbooks

Die folgenden Abschnitten helfen Ihnen, Ihr erstes Runbook erstellen.

Identifizieren Sie Ihren Anwendungsfall

Der erste Schritt beim Erstellen eines Runbooks besteht darin, Ihren Anwendungsfall zu identifizieren. Beispielsweise haben Sie geplant, das `AWS-CreateImage`-Runbook täglich auf allen Ihren Amazon EC2 Produktions-Instances auszuführen.. Am Ende des Monats entscheiden Sie, dass Sie über mehr Images verfügen, als für Wiederherstellungspunkte erforderlich sind. Künftig möchten Sie automatisch die älteste AMI einer Amazon-EC2-Instance löschen, wenn eine neue AMI erstellt wird. Um dies zu erreichen, erstellen Sie ein neues Runbook, das folgende Funktionen erfüllt:

1. Führt die `aws:createImage`-Aktion aus und gibt die Instance-ID in der Image-Beschreibung an.
2. Führt die `aws:waitForAwsResourceProperty`-Aktion aus, um den Zustand des Images abzufragen, bis es `available` ist.
3. Nachdem der Image-Status `available` ist, führt die`aws:executeScript`-Aktion ein benutzerdefiniertes Python-Skript aus, das die IDs aller Images sammelt, die mit Ihrer Amazon-EC2-Instance verknüpft sind. Das Skript führt diese Filterung aus, indem es die Instance-ID in der Image-Beschreibung verwendet, die Sie bei der Erstellung angegeben haben. Anschließend sortiert das Skript die Liste der Bild-IDs basierend auf dem `creationDate` des Images und gibt die ID der ältesten AMI aus.
4. Zu guter Letzt wird die `aws:deleteImage`-Aktion ausgeführt, um die älteste AMI zu löschen, mithilfe der ID aus der Ausgabe des vorherigen Schritts.

In diesem Szenario haben Sie bereits das `AWS-CreateImage`-Runbook verwendet, haben aber festgestellt, dass Ihr Anwendungsfall eine größere Flexibilität erforderte. Das kommt häufig vor, da es Überschneidungen zwischen Runbooks und Automatisierungsaktionen geben kann. Daher müssen Sie möglicherweise anpassen, welche Runbooks oder Aktionen Sie verwenden, um Ihren Anwendungsfall zu adressieren.

Zum Beispiel ermöglichen die `aws:executeScript`- und die `aws:invokeLambdaFunction`-Aktion es Ihnen, benutzerdefinierte Skripts als Teil Ihrer Automatisierung auszuführen. Sie bevorzugen vielleicht `aws:invokeLambdaFunction` aufgrund der zusätzlichen unterstützten Laufzeitsprachen. Möglicherweise bevorzugen Sie jedoch `aws:executeScript`, da Sie damit Ihre Skriptinhalte direkt in YAML Runbooks erstellen und Skriptinhalte als Anhänge für JSON-Runbooks bereitstellen können. Sie könnten auch `aws:executeScript` als einfacher in Bezug auf AWS Identity and Access Management (IAM)-Einrichtung empfinden. Da es die in der bereitgestellten Berechtigungen verwendet`AutomationAssumeRole`, `aws:executeScript` ist keine zusätzliche AWS Lambda Funktionsausführungsrolle erforderlich.

In einem bestimmten Szenario kann eine Aktion mehr Flexibilität oder zusätzliche Funktionalität gegenüber einer anderen bieten. Daher empfiehlt es sich, die verfügbaren Eingabeparameter für das Runbook oder die Aktion zu überprüfen, die Sie verwenden möchten, um zu bestimmen, welche am besten zu Ihrem Anwendungsfall und Ihren Voreinstellungen passt.

Einrichten Ihrer Entwicklungsumgebung

Nachdem Sie Ihren Anwendungsfall und die vordefinierten Runbooks oder Automatisierungsaktionen identifiziert haben, die Sie in Ihrem Runbook verwenden möchten, müssen Sie Ihre

Entwicklungsumgebung für den Inhalt Ihres Runbooks einrichten. Für die Entwicklung Ihrer Runbook-Inhalte empfehlen wir die Verwendung der Systems Manager-Dokumentenkonsole AWS Toolkit for Visual Studio Code anstelle der Systems Manager Documents Console.

Das Toolkit for VS Code ist eine Open-Source-Erweiterung für Visual Studio Code (VS Code), die mehr Funktionen bietet als die Systems Manager Dokumentenkonsole. Zu den hilfreichen Funktionen gehören die Schemavalidierung für YAML und JSON, Snippets für Automatisierungsaktionstypen und die automatische Vervollständigung verschiedener Optionen in YAML und JSON.

Weitere Informationen zum Installieren des Toolkit for VS Code finden Sie unterInstallieren von AWS Toolkit for Visual Studio Code. Weitere Informationen zur Verwendung des Toolkit for VS Code zum Erstellen von Runbooks finden Sie unter Arbeiten mit Systems Manager Automation-Dokumenten im AWS Toolkit for Visual Studio Code -Benutzerhandbuch.

Entwickeln von Runbook-Inhalten

Nachdem Ihr Anwendungsfall identifiziert und die Umgebung eingerichtet ist, können Sie den Inhalt für Ihr Runbook entwickeln. Ihr Anwendungsfall und Ihre Einstellungen bestimmen weitgehend die Automatisierungsaktionen oder Runbooks, die Sie in Ihren Runbook-Inhalten verwenden. Einige Aktionen unterstützen nur eine Teilmenge von Eingabeparametern im Vergleich zu einer anderen Aktion, mit der Sie eine ähnliche Aufgabe ausführen können. Andere Aktionen haben spezifische Ausgaben, wie `aws:createImage`, wo einige Aktionen es Ihnen ermöglichen, eigene Ausgaben zu definieren, z. B. `aws:executeAwsApi`.

Wenn Sie sich nicht sicher sind, wie Sie eine bestimmte Aktion in Ihrem Runbook verwenden, empfehlen wir Ihnen, den entsprechenden Eintrag für die Aktion im Systems Manager Automation Aktionen-Referenz nachzulesen. Wir empfehlen auch, den Inhalt vordefinierter Runbooks zu überprüfen, um Beispiele für die Verwendung dieser Aktionen zu sehen. Weitere Beispiele für Anwendungen von Runbooks in der Praxis finden Sie unter Weitere Runbook-Beispiele.

Um die Unterschiede in Bezug auf Einfachheit und Flexibilität zu demonstrieren, die Runbook-Inhalte bieten, bieten die folgenden Tutorials ein Beispiel, wie Sie Gruppen von Amazon-EC2-Instances stufenweise patchen:

- the section called “Beispiel 1: Erstellen von über- und untergeordneten Runbooks” – In diesem Beispiel werden zwei Runbooks in einer Untergeordnet-Übergeordnet-Beziehung verwendet. Das übergeordnete Runbook initiiert eine Automatisierung der Ratensteuerung des untergeordneten Runbooks.

- the section called “Beispiel 2: Skriptbasiertes Runbook” – Dieses Beispiel zeigt, wie Sie die gleichen Aufgaben von Beispiel 1 ausführen können, indem Sie den Inhalt zu einem einzigen Runbook zusammenfassen und Skripte in Ihrem Runbook verwenden.

Beispiel 1: Erstellen von über- und untergeordneten Runbooks

Das folgende Beispiel erläutert, wie Sie zwei Runbooks erstellen, die getaggte Gruppen von Amazon Elastic Compute Cloud (Amazon EC2)-Instances stufenweise patchen. Diese Runbooks werden in einer Untergeordnet–Übergeordnet-Beziehung mit dem übergeordneten Runbook verwendet, das verwendet wird, um eine Kurssteuerungsautomatisierung des untergeordneten Runbooks zu initiieren. Weitere Informationen über die Ratenregelung-Automatisierungen finden Sie unter Ausführen von Automatisierungen im großen Maßstab. Weitere Informationen zu den hier verwendeten Automation-Aktionen finden Sie unter Systems Manager Automation Aktionen-Referenz.

Erstellen des untergeordneten Runbooks

In diesem Beispiel-Runbook wird das folgende Szenario behandelt. Emily ist Systemingenieurin bei AnyCompany Consultants, LLC. Sie muss Patches für Gruppen von Amazon Elastic Compute Cloud (Amazon EC2)-Instances konfigurieren, die primäre und sekundäre Datenbanken hosten. Anwendungen greifen 24 Stunden am Tag auf diese Datenbanken zu, sodass eine der Datenbankinstances immer verfügbar sein muss.

Sie entscheidet, dass das Patchen der Instances stufenweise der beste Ansatz ist. Die primäre Gruppe von Datenbankinstances wird zuerst gepatcht, gefolgt von der sekundären Gruppe von Datenbankinstances. Um zusätzliche Kosten zu vermeiden, indem Instances ausgeführt werden, die zuvor gestoppt wurden, möchte Emily außerdem, dass die gepatchten Instances in ihren ursprünglichen Zustand zurückversetzt werden, bevor das Patchen stattgefunden hat.

Emily identifiziert die primären und sekundären Gruppen von Datenbankinstances anhand der Tags, die den Instances zugeordnet sind. Sie beschließt, ein übergeordnetes Runbook zu erstellen, das eine Automatisierung der Ratenkontrolle eines untergeordneten Runbooks startet. Auf diese Weise kann sie die Tags ausrichten, die mit den primären und sekundären Gruppen von Datenbank-Instances verknüpft sind, und die Parallelität der untergeordneten Automatisierungen verwalten. Nachdem sie die verfügbaren Systems Manager (SSM)-Dokumente zum Patchen überprüft hat, wählt sie das `AWS-RunPatchBaseline`-Document. Mithilfe dieses SSM-Dokuments können ihre Kollegen die zugehörigen Patch-Compliance-Informationen überprüfen, nachdem der Patch-Vorgang abgeschlossen ist.

Um mit der Erstellung ihrer Runbook-Inhalte zu beginnen, überprüft Emily die verfügbaren Automatisierungsaktionen und beginnt mit der Erstellung des Inhalts für das untergeordnete Runbook wie folgt:

1. Zunächst stellt sie Werte für das Schema und die Beschreibung des Runbooks bereit und definiert die Eingabeparameter für das untergeordnete Runbook.

   Durch die Verwendung des `AutomationAssumeRole`-Parameters können Emily und ihre Kollegen eine vorhandene IAM-Rolle verwenden, die der Automatisierung erlaubt, die Aktionen im Runbook für sie auszuführen. Emily verwendet das `InstanceId`-Parameter, um die Instance zu bestimmen, die gepatcht werden soll. Optional können die `Operation`-,`RebootOption`- , und`SnapshotId`-Parameter verwendet werden, um Werte für Dokumentparameter für `AWS-RunPatchBaseline` bereitzustellen. Um zu verhindern, dass für diese Dokumentparameter ungültige Werte bereitgestellt werden, definiert sie die `allowedValues` nach Bedarf.

   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. Wenn die Elemente der obersten Ebene definiert sind, arbeitet Emily mit der Erstellung der Aktionen, welche die `mainSteps` des Runbooks melden. Der erste Schritt gibt den aktuellen Status der Ziel-Instance aus, die im `InstanceId`-Eingabeparameter mit der `aws:executeAwsApi`-Aktion angegeben ist. Die Ausgabe dieser Aktion wird in späteren Aktionen verwendet.

   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. Anstatt den ursprünglichen Zustand jeder Instance, die gepatcht werden muss, manuell zu starten und zu verfolgen, verwendet Emily die Ausgabe der vorherigen Aktion, um die Automatisierung basierend auf dem Status der Ziel-Instance zu verzweigen. Auf diese Weise kann die Automatisierung verschiedene Schritte ausführen, abhängig von den Bedingungen, die in der `aws:branch`-Aktion angegeben sind und verbessert die Gesamteffizienz der Automatisierung ohne manuellen Eingriff.

Wenn der Instance-Status bereits `running` ist, schreitet die Automatisierung mit dem Patchen der Instance mit dem `AWS-RunPatchBaseline`-Dokument unter Verwendung der `aws:runCommand`-Aktion fort.

Wenn der Instancestatus `stopping` ist, fragt die Automatisierung ab, ob die Instance den Status `stopped` mit der Aktion `aws:waitForAwsResourceProperty` erreicht, startet die Instance mit der Aktion `executeAwsApi` und fragt die Instance ab, um den Status `running` zu erreichen, bevor die Instance gepatcht wird.

Wenn der Status der Instance `stopped` ist, startet die Automatisierung die Instance und fragt die Instance ab, einen `running`-Status vor dem Patchen der Instance unter Verwendung der gleichen Aktionen zu erreichen.

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. Nach Abschluss des Patching-Vorgangs möchte Emily, dass die Automatisierung die Ziel-Instance in denselben Zustand versetzt, in dem sie sich vor dem Automatisierungsstart befanden. Sie tut dies, indem sie erneut die Ausgabe der ersten Aktion verwendet. Die Automatisierung verzweigt sich basierend auf dem ursprünglichen Zustand der Ziel-Instance unter Verwendung der `aws:branch`-Aktion. Wenn sich die Instance zuvor in einem anderen Zustand als `running` befand, wird die Instance angehalten. Lautet der Status der Instance `running`, stoppt die Automatisierung.

   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 überprüft den abgeschlossenen untergeordneten Runbook-Inhalt und erstellt das Runbook im selben AWS-Konto und der selben AWS-Region als Ziel-Instances. Jetzt ist sie bereit, mit der Erstellung des übergeordneten Runbooks fortzufahren. Im Folgenden finden Sie den vollständigen untergeordneten Runbook-Inhalt.

   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}}"
            ]
         }
      }
   ]
}
```

Weitere Informationen zu den hier verwendeten Automation-Aktionen finden Sie unter Systems Manager Automation Aktionen-Referenz.

Erstellen des übergeordneten Runbooks

In diesem Beispiel-Runbook wird das Szenario fortgesetzt, das im vorherigen Abschnitt beschrieben wird. Nachdem Emily nun das untergeordnete Runbook erstellt hat, beginnt sie mit der Erstellung des Inhalts für das übergeordnete Runbook wie folgt:

1. Zunächst stellt sie Werte für das Schema und die Beschreibung des Runbooks bereit und definiert die Eingabeparameter für das übergeordnete Runbook.

   Durch die Verwendung des `AutomationAssumeRole`-Parameters können Emily und ihre Kollegen eine vorhandene IAM-Rolle verwenden, die der Automatisierung erlaubt, die Aktionen im Runbook für sie auszuführen. Emily verwendet die `PatchGroupPrimaryKey`- und `PatchGroupPrimaryValue`-Parameter, um das Tag anzugeben, das mit der primären Gruppe von Datenbankinstances verknüpft ist, die gepatcht werden sollen. Sie verwendet den `PatchGroupSecondaryKey`- und `PatchGroupSecondaryValue`-Parameter, um das Tag anzugeben, das mit der sekundären Gruppe von Datenbankinstances verknüpft ist, die gepatcht werden sollen.

   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. Wenn die Elemente der obersten Ebene definiert sind, arbeitet Emily mit der Erstellung der Aktionen, welche die `mainSteps` des Runbooks melden.

   Bei der ersten Aktion wird eine Ratensteuerungsautomatisierung mit dem soeben erstellten untergeordneten Runbook gestartet, das Instances betrifft, die mit dem Tag verknüpft sind, das in den `PatchGroupPrimaryKey`- und `PatchGroupPrimaryValue`-Eingabeparametern angegeben ist. Sie verwendet die Werte, die den Eingabeparametern zur Verfügung gestellt werden, um den Schlüssel und den Wert des Tags anzugeben, welcher der primären Gruppe von Datenbankinstances zugeordnet ist, die sie patchen möchte.

   Nach der Fertigstellung der ersten Automatisierung, startet die zweite Aktion eine andere Ratensteuerungsautomatisierung unter Verwendung des untergeordneten Runbooks, das Instances betrifft, die mit dem Tag verknüpft sind, das in den `PatchGroupSecondaryKey`- und `PatchGroupSecondaryValue`-Eingabeparametern angegeben ist. Sie verwendet die Werte, die den Eingabeparametern zur Verfügung gestellt werden, um den Schlüssel und den Wert des Tags anzugeben, welcher der sekundären Gruppe von Datenbankinstances zugeordnet ist, die sie patchen möchte.

   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 überprüft den abgeschlossenen übergeordneten Runbook-Inhalt und erstellt das Runbook im selben AWS-Konto und der selben AWS-Region als Ziel-Instances. Jetzt ist sie bereit, ihre Runbooks zu testen, um sicherzustellen, dass die Automatisierung wie gewünscht funktioniert, bevor sie in ihre Produktionsumgebung implementiert werden. Im Folgenden finden Sie den vollständigen übergeordneten Runbook-Inhalt.

   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"
         }
      }
   ]
}
```

Weitere Informationen zu den hier verwendeten Automation-Aktionen finden Sie unter Systems Manager Automation Aktionen-Referenz.

Beispiel 2: Skriptbasiertes Runbook

In diesem Beispiel-Runbook wird das folgende Szenario behandelt. Emily ist Systemingenieurin bei AnyCompany Consultants, LLC. Sie hat zuvor zwei Runbooks erstellt, die in einer Untergeordnet-Übergeordnet-Beziehung verwendet werden, um Patch-Gruppen von Amazon Elastic Compute Cloud (Amazon EC2)-Instances zu patchen, die primäre und sekundäre Datenbanken hosten. Anwendungen greifen 24 Stunden am Tag auf diese Datenbanken zu, sodass eine der Datenbankinstances immer verfügbar sein muss.

Basierend auf dieser Anforderung hat sie eine Lösung entwickelt, welche die Instances stufenweise mit dem `AWS-RunPatchBaseline`-Systems Manager (SSM)-Dokument patcht. Mithilfe dieses SSM-Dokuments können ihre Kollegen die zugehörigen Patch-Compliance-Informationen überprüfen, nachdem der Patch-Vorgang abgeschlossen ist.

Die primäre Gruppe von Datenbankinstances wird zuerst gepatcht, gefolgt von der sekundären Gruppe von Datenbankinstances. Um zusätzliche Kosten zu vermeiden, indem Instances ausgeführt werden, die zuvor gestoppt wurden, hat Emily sichergestellt, dass die Automatisierung die gepatchten Instances in ihren ursprünglichen Zustand zurückversetzte, bevor das Patchen stattgefunden hat. Emily verwendete Tags, die den primären und sekundären Gruppen von Datenbankinstances zugeordnet sind, um zu ermitteln, welche Instances in der gewünschten Reihenfolge gepatcht werden sollen.

Ihre bestehende automatisierte Lösung funktioniert, aber sie will ihre Lösung nach Möglichkeit verbessern. Um bei der Wartung des Runbook-Inhalts zu helfen und die Fehlerbehebung zu erleichtern, möchte sie die Automatisierung zu einem einzigen Runbook zusammenfassen und die Anzahl der Eingabeparameter vereinfachen. Außerdem möchte sie vermeiden, dass mehrere untergeordnete Automatisierungen erstellt werden.

Nachdem Emily die verfügbaren Automatisierungsaktionen überprüft hat, stellt sie fest, dass sie ihre Lösung mithilfe der `aws:executeScript`-Aktion verbessern kann, um ihre benutzerdefinierten Python-Skripte auszuführen. Sie beginnt nun mit der Erstellung des Inhalts für das Runbook wie folgt:

1. Zunächst stellt sie Werte für das Schema und die Beschreibung des Runbooks bereit und definiert die Eingabeparameter für das übergeordnete Runbook.

   Durch die Verwendung des `AutomationAssumeRole`-Parameters können Emily und ihre Kollegen eine vorhandene IAM-Rolle verwenden, die der Automatisierung erlaubt, die Aktionen im Runbook für sie auszuführen. Im Gegensatz zum Beispiel 1 ist der `AutomationAssumeRole`-Parameter ist jetzt erforderlich und nicht optional. Da dieses Runbook `aws:executeScript`-Aktionen beinhaltet, ist eine AWS Identity and Access Management-(IAM)-Servicerolle (oder Assume-Rolle) immer erforderlich. Diese Anforderung ist notwendig, da einige der Python-Skripte, die für die Aktionen angegeben sind, AWS-API-Operationen aufrufen.

   Emily verwendet die `PrimaryPatchGroupTag`- und `SecondaryPatchGroupTag`-Parameter, um die Tags anzugeben, die mit der primären und sekundären Gruppe von Datenbankinstances verknüpft sind, die gepatcht werden sollen. Um die erforderlichen Eingabeparameter zu vereinfachen, entscheidet sie sich, `StringMap`-Parameter anstatt mehrerer `String`-Parameter zu verwenden, wie sie im Runbook von Beispiel 1 verwendet wurde. Optional können die `Operation`-,`RebootOption`- , und`SnapshotId`-Parameter verwendet werden, um Werte für Dokumentparameter für `AWS-RunPatchBaseline` bereitzustellen. Um zu verhindern, dass für diese Dokumentparameter ungültige Werte bereitgestellt werden, definiert sie die `allowedValues` nach Bedarf.

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. Wenn die Elemente der obersten Ebene definiert sind, arbeitet Emily mit der Erstellung der Aktionen, welche die `mainSteps` des Runbooks melden. Der erste Schritt sammelt die IDs aller Instances, die dem im `PrimaryPatchGroupTag`-Parameter angegebenen Tag zugeordnet sind, und gibt einen `StringMap`-Parameter aus, der die Instance-ID und den aktuellen Zustand der Instance enthält. Die Ausgabe dieser Aktion wird in späteren Aktionen verwendet.

   Beachten Sie, dass die `script`-Eingabeparameter für JSON-Runbooks nicht unterstützt werden. JSON-Runbooks müssen Skriptinhalt mithilfe des `attachment`-Eingabeparameters bereitstellen.

   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 verwendet die Ausgabe der vorherigen Aktion in einer anderen `aws:executeScript`-Aktion, um zu überprüfen, ob alle Instances, die dem im `PrimaryPatchGroupTag`-Parameter angegebenen Tag zugeordnet sind, in einem `running`-Zustand sind.

   Wenn der Instance-Status bereits `running` oder `shutting-down` ist, durchläuft das Skript weiterhin die verbleibenden Instances.

   Wenn der Status der Instance `stopping` ist, fragt das Skript die Instance ab, den `stopped`-Status zu erreichen und startet die Instance.

   Wenn der Status der Instance `stopped` ist, startet das Skript die Instance.

   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 überprüft, ob alle Instances, die dem im `PrimaryPatchGroupTag`-Parameter angegebenen Tag zugeordnet sind, gestartet wurden oder sich bereits in einem `running`-Zustand befinden. Dann verwendet sie ein anderes Skript, um zu überprüfen, ob alle Instances,

einschließlich derjenigen, die in der vorherigen Aktion gestartet wurden, den `running`-Zustand erreicht haben.

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 verwendet zwei weitere Skripte, um einzelne `String`-Werte des Schlüssels und des Werts des Tags zurückzugeben, das im `PrimaryPatchGroupTag`-Parameter angegeben ist. Die Werte, die von diesen Aktionen zurückgegeben werden, ermöglichen es ihr, Werte direkt für die `Targets`-Parameter für das `AWS-RunPatchBaseline`-Dokument bereitzustellen. Die Automatisierung schreitet dann mit dem Patchen der Instance mit dem `AWS-RunPatchBaseline`-Dokument unter Verwendung der `aws:runCommand`-Aktion fort.

   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. Nach Abschluss des Patching-Voragngs möchte Emily, dass die Automatisierung die Ziel-Instances, die dem im `PrimaryPatchGroupTag`-Parameter angegebenen Tag zugeordnet sind, in den Zustand zurückversetzt, in dem sie sich vor dem Automatisierungsstart befanden. Sie tut dies, indem sie erneut die Ausgabe der ersten Aktion in einem Skript verwendet. Basierend auf dem ursprünglichen Zustand der Ziel-Instance, wenn sich die Instance zuvor in einem anderen Zustand als `running` befand, wird die Instance angehalten. Wenn der Instance-Status bereits `running` ist, durchläuft das Skript weiterhin die verbleibenden Instances.

   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. Der Patchvorgang wird für die Instances abgeschlossen, die mit dem Tag verknüpft sind, das im `PrimaryPatchGroupTag`-Parameter angegeben ist. Jetzt dupliziert Emily alle

vorherigen Aktionen in ihrem Runbook-Inhalt, um auf die Instances abzuzielen, die dem im `SecondaryPatchGroupTag`-Parameter angegebenen Tag zugeordnet sind.

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 überprüft den abgeschlossenen geskripteten Runbook-Inhalt und erstellt das Runbook im selben AWS-Konto und der selben AWS-Region als Ziel-Instances. Jetzt ist sie bereit, ihr Runbook zu testen, um sicherzustellen, dass die Automatisierung wie gewünscht funktioniert, bevor es

in ihre Produktionsumgebung implementiert wird. Im Folgenden finden Sie den vollständigen geskripteten Runbook-Inhalt.

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":"..."
         }
      }
   ]
}
```

Weitere Informationen zu den hier verwendeten Automation-Aktionen finden Sie unter Systems Manager Automation Aktionen-Referenz.

Weitere Runbook-Beispiele

Das folgende Beispiel-Runbook veranschaulicht, wie Sie mithilfe von AWS Systems Manager-Automatisierungsaktionen gängige Bereitstellungs-, Problembehandlungs- und Wartungsaufgaben automatisieren können.

> **Note**
>
> Die Beispiel-Runbooks in diesem Abschnitt werden bereitgestellt, um zu veranschaulichen, wie Sie benutzerdefinierte Runbooks erstellen können, um Ihre spezifischen Betriebsanforderungen zu erfüllen. Diese Runbooks sind nicht für den Einsatz in Produktionsumgebungen vorgesehen. Sie können sie jedoch für Ihren eigenen Gebrauch anpassen.

Beispiele

- Bereitstellung der VPC-Architektur und der Microsoft Active Directory-Domänencontroller
- Wiederherstellen eines Root-Volumes aus dem letzten Snapshot
- Erstellen eines AMI und einer regionenübergreifenden Kopie

Bereitstellung der VPC-Architektur und der Microsoft Active Directory-Domänencontroller

Um die Effizienz zu steigern und allgemeine Aufgaben zu standardisieren, können Sie sich für die Automatisierung von Bereitstellungen entscheiden. Dies ist nützlich, wenn Sie regelmäßig dieselbe Architektur für mehrere Konten und AWS-Regionen bereitstellen. Die Automatisierung von Architekturbereitstellungen kann auch das Potenzial für menschliche Fehler reduzieren, die bei der manuellen Bereitstellung der Architektur auftreten können. AWS Systems Manager Automatisierungskationen können Ihnen dabei helfen. Automation ist eine Funktion von AWS Systems Manager.

Das folgende AWS Systems Manager-Beispiel-Runbook führt diese Aktionen aus:

- Ruft das neueste Windows Server 2016 Amazon Machine Image (AMI) mithilfe von Systems Manager Parameter Store ab, das beim Starten der EC2-Instances verwendet wird, die als Domaincontroller konfiguriert werden. Parameter Store ist eine Fähigkeit von AWS Systems Manager.
- Verwendet die `aws:executeAwsApi` Automatisierungsaktion, um mehrere AWS-API-Aktionen zum Erstellen der VPC-Architektur aufzurufen. Die Domänencontroller-Instances werden in

privaten Subnetzen gestartet und stellen über ein NAT-Gateway eine Verbindung zum Internet her. Dies ermöglicht dem SSM Agent auf den Instances auf die erforderlichen Systems Manager Endpunkte zuzugreifen.

- Verwendet die `aws:waitForAwsResourceProperty` Automation-Aktion zur Bestätigung, dass die durch die vorherige Aktion gestarteten Instances `Online` für AWS Systems Manager sind.
- Verwendet die `aws:runCommand` Automatisierungsaktion zur Konfiguration der als Microsoft Active Directory-Domänencontroller gestarteten Instances.

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: '$.InstanceStatuses[0].InstanceState.Name'
          DesiredValues:
            - running
        nextStep: verifyInstancesOnlineSSM
      - name: verifyInstancesOnlineSSM
        action: aws:waitForAwsResourceProperty
        timeoutSeconds: 600
        inputs:
          Service: ssm
          Api: DescribeInstanceInformation
          InstanceInformationFilterList:
            - key: InstanceIds
              valueSet:
                - '{{ launchDC1.pdcInstanceId }}'
                - '{{ launchDC2.adcInstanceId }}'
          PropertySelector: '$.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 $_
              }
```

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 $_",
                "}"
              ]
            }
          }
        }
      ]
```

```
    }
```

Wiederherstellen eines Root-Volumes aus dem letzten Snapshot

Das Betriebssystem auf einem Root-Volume kann aus verschiedenen Gründen beschädigt werden. Beispielsweise können Instances nach einem Patchvorgang aufgrund eines beschädigten Kernels oder einer beschädigten Registrierung nicht mehr erfolgreich gestartet werden. Die Automatisierung gängiger Fehlerbehebungsaufgaben, wie z. B. die Wiederherstellung eines Root-Volumes aus dem letzten vor dem Patchvorgang erstellten Snapshot, kann die Ausfallzeit reduzieren und die Fehlerbehebung beschleunigen. AWS Systems Manager Automatisierungskationen können Ihnen dabei helfen. Automation ist eine Funktion von AWS Systems Manager.

Das folgende AWS Systems Manager-Beispiel-Runbook führt diese Aktionen aus:

- Verwendet die `aws:executeAwsApi` Automatisierungsaktion zum Abrufen von Details aus dem Root-Volumen der Instance.
- Verwendet die `aws:executeScript` Automatisierungsaktion zum Abrufen des neuesten Snapshots für das Root-Volume.
- Verwendet die `aws:branch` Automatisierungsaktion, um die Automatisierung fortzusetzen, wenn ein Snapshot für das Root-Volume gefunden wird.

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"
               }
           }
       }
   }
```

Erstellen eines AMI und einer regionenübergreifenden Kopie

Das Erstellen eines Amazon Machine Image (AMI) einer Instance ist ein üblicher Prozess, der bei Backup und Wiedherherstellung verwendet wird. Sie können sich auch dafür entscheiden, ein AMI als Teil einer Notfallwiederherstellungsarchitektur in eine andere AWS-Region zu kopieren. Durch die Automatisierung gängiger Wartungsaufgaben kann die Ausfallzeit reduziert werden, wenn ein Problem ein Failover erfordert. AWS Systems Manager Automatisierungskationen können Ihnen dabei helfen. Automation ist eine Funktion von AWS Systems Manager.

Das folgende AWS Systems Manager-Beispiel-Runbook führt diese Aktionen aus:

- Verwendet die `aws:executeAwsApi` Automatisierungsaktion zur Erstellung eines AMI.
- Verwendet die `aws:waitForAwsResourceProperty` Automatisierungsaktion zur Bestätigung der Verfügbarkeit des AMI.
- Verwendet die `aws:executeScript` Automatisierungsaktion zum Kopieren des AMI in die Zielregion.

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"
               }
           }
       }
   }
```

Eingabeparameter erstellen, die Ressourcen auffüllen AWS

Die Automatisierung, eine Funktion von Systems Manager, füllt AWS Ressourcen in die, AWS Management Console die dem Ressourcentyp entsprechen, den Sie für einen Eingabeparameter definieren. Ressourcen in Ihrem AWS-Konto , die mit dem Ressourcentyp übereinstimmen, werden in einer Dropdown-Liste angezeigt, die Sie auswählen können. Sie können Eingabeparametertypen für Amazon Elastic Compute Cloud (Amazon EC2) -Instances, Amazon Simple Storage Service (Amazon S3) -Buckets und AWS Identity and Access Management (IAM) -Rollen definieren. Die unterstützten Typdefinitionen und die regulären Ausdrücke, die zum Suchen übereinstimmender Ressourcen verwendet werden, lauten wie folgt:

- `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/.*$`

Es folgt ein Beispiel für Eingabeparameter-Typen, die im Runbook-Inhalt definiert sind.

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
     }
  ]
}
```

Verwenden von Document Builder zur Erstellung von Runbooks

Wenn die AWS Systems Manager öffentlichen Runbooks nicht alle Aktionen unterstützen, die Sie für Ihre AWS Ressourcen ausführen möchten, können Sie Ihre eigenen Runbooks erstellen. Um ein benutzerdefiniertes Runbook zu erstellen, können Sie manuell eine lokale Datei im YAML- oder JSON-Format mit den entsprechenden Automatisierungsaktionen erstellen. Alternativ können Sie Document Builder in der Systems-Manager-Automation-Konsole verwenden, um ein benutzerdefiniertes Runbook zu erstellen.

Mit Document Builder können Sie Ihrem benutzerdefinierten Runbook Automatisierungsaktionen hinzufügen und die erforderlichen Parameter bereitstellen, ohne die JSON- oder YAML-Syntax verwenden zu müssen. Nachdem Sie Schritte hinzugefügt und das Runbook erstellt haben, konvertiert das System die von Ihnen hinzugefügten Aktionen in das YAML-Format, das von Systems Manager zum Ausführen von Automation verwendet werden kann.

Runbooks unterstützen die Verwendung von Markdown, einer Markup-Sprache, mit der Sie Wiki-Beschreibungen zu Runbooks und einzelnen Schritten innerhalb des Runbooks hinzufügen können. Weitere Informationen zur Verwendung von Markdown finden Sie unter Verwenden von Markdown in AWS.

Erstellen eines Runbooks mithilfe von Document Builder

Bevor Sie beginnen

Wir empfehlen Ihnen, sich über die verschiedenen Aktionen zu informieren, die Sie in einem Runbook verwenden können. Weitere Informationen finden Sie unter Systems Manager Automation Aktionen-Referenz.

So erstellen Sie ein Runbook mit Document Builder

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie Create automation (Automation erstellen).
4. Geben Sie unter Name einen aussagekräftigen Namen für das Runbook ein.
5. Geben Sie für Document description (Dokumentbeschreibung) die Beschreibung des Markdown-Stils für das Runbook an. Sie können Anweisungen für die Verwendung des Runbooks, nummerierte Schritte oder jede andere Art von Informationen zur Beschreibung des Runbooks bereitstellen. Informationen zum Formatieren von Inhalten finden Sie im Standardtext.

> **Tip**
>
> Wechseln Sie zwischen Hide preview (Vorschau ausblenden) und Show preview (Vorschau anzeigen), um zu sehen, wie der Beschreibungsinhalt während der Erstellung aussieht.

6. (Optional) Geben Sie unter Assume role (Rolle übernehmen) den Namen oder den ARN einer Servicerolle ein, um Aktionen in Ihrem Auftrag auszuführen. Wenn Sie keine Rolle angeben, verwendet Automation die Zugriffsberechtigungen des Benutzers, der die Automatisierung durchführt.

   > **Important**
   >
   > Für Runbooks, die sich nicht im Besitz von Amazon befinden und die die `aws:executeScript`-Aktion verwenden, muss eine Rolle angegeben werden. Weitere Informationen finden Sie unter Berechtigungen für die Verwendung von Runbooks.

7. (Optional) Geben Sie unter Outputs (Ausgänge) alle Ausgaben für die Automatisierung dieses Runbooks ein, um sie für andere Prozesse verfügbar zu machen.

   Wenn Ihr Runbook beispielsweise ein neues AMI erstellt, können Sie [“ CreateImage angeben. ImageId„] und verwenden Sie dann diese Ausgabe, um in einer nachfolgenden Automatisierung neue Instances zu erstellen.

8. (Optional) Erweitern Sie den Abschnitt Input parameters (Eingabeparameter) und führen Sie die folgenden Schritte aus.

   1. Geben Sie unter Parameter name (Parametername) einen beschreibenden Namen für den Runbookparameter ein, den Sie erstellen.
   2. Wählen Sie unter Type (Typ) einen Typ für den Parameter, z. B. `String` oder `MapList`.
   3. Führen Sie unter Required (Erforderlich) eine der folgenden Aktionen aus:
      - Wählen Sie Yes (Ja), wenn zur Laufzeit ein Wert für diesen Runbookparameter angegeben werden muss.
      - Wählen Sie No (Nein), wenn der Parameter nicht erforderlich ist, und geben Sie (optional) unter Default value (Standardwert) einen Standardparameterwert ein.
   4. Geben Sie unter Description (Beschreibung) eine Beschreibung für den Runbookparameter ein.

> **Note**
>
> Um weitere Runbookparameter hinzuzufügen, wählen Sie Add a parameter (Parameter hinzufügen). Um einen Runbookparameter zu entfernen, klicken Sie auf die Schaltfläche X (Entfernen).

9. (Optional) Erweitern Sie den Abschnitt Target type (Zieltyp) und wählen Sie einen Zieltyp, um die Arten der Ressourcen zu definieren, auf denen die Automatisierung ausgeführt werden kann. Um beispielsweise ein Runbook auf EC2-Instances zu wählen, wählen Sie `/AWS::EC2::Instance`.

   > **Note**
   >
   > Wenn Sie den Wert '/' angeben, kann das Runbook auf allen Arten von Ressourcen ausgeführt werden. Eine Liste gültiger Ressourcentypen finden Sie unter AWS - Ressourcentypen – Referenz im AWS CloudFormation Benutzerhandbuch.

10. (Optional) Erweitern Sie den Abschnitt Document tags (Dokument-Tags) und geben Sie ein oder mehrere Tag-Schlüssel-Wert-Paare ein, die auf das Runbook angewendet werden sollen. Tags erleichtern die Identifizierung, Organisation und Suche nach Ressourcen. Weitere Informationen finden Sie unter Markierungen von Systems Manager-Dokumenten.
11. Geben Sie im Abschnitt Step 1 (Schritt 1) die folgenden Informationen an.

    - Geben Sie unter Step name (Schrittname) einen beschreibenden Namen für den ersten Schritt der Automatisierung ein.
    - Wählen Sie unter Action type (Aktionstyp) den Aktionstyp aus, der für diesen Schritt verwendet werden soll.

      Eine Liste und Informationen zu den verfügbaren Aktionstypen finden Sie unter Systems Manager Automation Aktionen-Referenz.
    - Geben Sie unter Description (Beschreibung) eine Beschreibung für den Automatisierungsschritt ein. Sie können Markdown verwenden, um Ihren Text zu formatieren.
    - Je nach ausgewähltem Action type (Aktionstyp) geben Sie im Abschnitt Step inputs (Schritteingaben) die erforderlichen Eingaben für den Aktionstyp ein. Wenn Sie beispielsweise die Aktion `aws:approve` ausgewählt haben, müssen Sie einen Wert für die `Approvers`-Eigenschaft angeben.

Informationen zu den Schritteingabefeldern finden Sie im Eintrag Systems Manager Automation Aktionen-Referenz für den ausgewählten Aktionstyp. Zum Beispiel: `aws:executeStateMachine` – Führen Sie eine AWS Step Functions-State Machine aus..

- (Optional) Geben Sie für Additional inputs (Zusätzliche Eingaben) alle zusätzlichen Eingabewerte an, die für das Runbook erforderlich sind. Die verfügbaren Eingabetypen hängen vom Aktionstyp ab, den Sie für den Schritt ausgewählt haben. (Beachten Sie, dass einige Aktionstypen Eingabewerte erfordern.)

  > **Note**
  >
  > Um weitere Eingaben hinzuzufügen, wählen Sie Add optional input (Optionale Eingabe hinzufügen). Um eine Eingabe zu entfernen, wählen Sie die Schaltfläche X (Entfernen).

- (Optional) Geben Sie unter Outputs (Ausgänge) alle Ausgaben für diesen Schritts ein, um sie für andere Prozesse verfügbar zu machen.

  > **Note**
  >
  > Outputs (Ausgaben) sind nicht für alle Aktionstypen verfügbar.

- (Optional) Erweitern Sie den Abschnitt Common properties (Allgemeine Eigenschaften) und geben Sie Eigenschaften für die Aktionen an, die allen Automation-Aktionen gemeinsam sind. Beispielsweise können Sie für Timeout seconds (Timeout in Sekunden) anhand eines Werts in Sekunden angeben, wie lange der Schritt ausgeführt werden kann, bevor er beendet wird.

  Weitere Informationen finden Sie unter Von allen Aktionen gemeinsam genutzte Eigenschaften.

> **Note**
>
> Um weitere Schritte hinzuzufügen, wählen Sie Add step (Schritt hinzufügen) aus und wiederholen Sie das Verfahren zum Erstellen eines Schritts. Um einen Schritt zu entfernen, wählen Sie Remove step (Schritt entfernen).

12. Wählen Sie Create automation (Automation erstellen), um das Runbook zu speichern.

Erstellen eines Runbooks, das Skripte ausführt

Das folgende Verfahren zeigt, wie Sie mit Document Builder in der AWS Systems Manager - Automation-Konsole ein benutzerdefiniertes Runbook erstellen, das ein Skript ausführt.

Im ersten Schritt des von Ihnen erstellten Runbooks wird ein Skript ausgeführt, um eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance zu starten. Im zweiten Schritt wird ein weiteres Skript ausgeführt, um zu überwachen, ob die Instance-Zustandsprüfung auf `ok` geändert werden soll. Anschließend wird für die Automatisierung ein Gesamtzustand von `Success` gemeldet.

Bevor Sie beginnen

Stellen Sie sicher, dass Sie die folgenden Schritte ausgeführt haben:

- Stellen Sie sicher, dass Sie über Administratorrechte verfügen oder dass Ihnen die entsprechenden Berechtigungen für den Zugriff auf Systems Manager in AWS Identity and Access Management (IAM) erteilt wurden.

  Weitere Informationen finden Sie unter Überprüfen des Benutzerzugriffs für Runbooks.
- Stellen Sie sicher, dass Sie in Ihrem AWS-Kontoüber eine IAM-Service-Rolle für Automation (auch als Rolle übernehmen bezeichnet) verfügen. Die Rolle ist erforderlich, da in dieser Anleitung die Aktion `aws:executeScript` verwendet wird.

  Weitere Informationen zum Erstellen dieser Rolle finden Sie unter Konfigurieren eines Service-Rollenzugriffs (Rolle übernehmen) für Automatisierungen.

  Hinweise zur IAM-Service-Rollenanforderung zum Ausführen von `aws:executeScript`, finden Sie unter Berechtigungen für die Verwendung von Runbooks.
- Stellen Sie sicher, dass Sie berechtigt sind, EC2-Instances zu starten.

  Weitere Informationen finden Sie unter IAM und Amazon EC2 im Amazon EC2 EC2-Benutzerhandbuch.

So erstellen Sie ein benutzerdefiniertes Runbook, das Skripts mit Document Builder ausführt

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie Create automation (Automation erstellen).

4. Geben Sie unter Name diesen beschreibenden Namen für das Runbook ein: **`LaunchInstanceAndCheckStatus`**.

5. (Optional) Ersetzen Sie bei Document description (Dokumentbeschreibung) den Standardtext durch eine Beschreibung für dieses Runbooks, indem Sie Markdown verwenden. Im Folgenden wird ein Beispiel gezeigt.

```
##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. Geben Sie unter Assume role (Rolle übernehmen) den ARN der IAM-Service-Rolle für Automation (Rolle übernehmen) für die Automatisierung im Format **`arn:aws:iam::111122223333:role/AutomationServiceRole`** ein. Ersetzen Sie 111122223333 durch Ihre AWS-Konto ID.

   Die von Ihnen angegebene Rolle wird verwendet, um die Berechtigungen bereitzustellen, die zum Starten der Automatisierung erforderlich sind.

   > **Important**
   >
   > Für Runbooks, die sich nicht im Besitz von Amazon befinden und die die `aws:executeScript`-Aktion verwenden, muss eine Rolle angegeben werden. Weitere Informationen finden Sie unter Berechtigungen für die Verwendung von Runbooks.

7. Erweitern Sie Input parameters (Eingabeparameter) und gehen Sie folgendermaßen vor.

   1. Geben Sie unter Parameter name (Parametername) **`imageId`** ein.

2. Wählen Sie für Type (Typ) die Option **`String`** aus.
3. Wählen Sie unter Required (Erforderlich) die Option No.
4. Geben Sie unter Default value (Standardwert) Folgendes ein.

```
{{ ssm:/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2 }}
```

> **Note**
>
> Dieser Wert startet eine Amazon EC2 EC2-Instance mit der neuesten Amazon Linux 1 Amazon Machine Image (AMI) -ID. Wenn Sie ein anderes AMI verwenden möchten, ersetzen Sie den Wert durch Ihre AMI-ID.

5. Geben Sie unter Description (Beschreibung) Folgendes ein.

```
(Optional) The AMI ID to use for launching the instance. The default value uses
 the latest released Amazon Linux AMI ID.
```

8. Wählen Sie Add a parameter (Parameter hinzufügen), um den zweiten Parameter **`tagValue`** zu erstellen, und geben Sie Folgendes ein.

1. Geben Sie unter Parameter name (Parametername) **`tagValue`** ein.
2. Wählen Sie für Type (Typ) die Option **`String`** aus.
3. Wählen Sie unter Required (Erforderlich) die Option No.
4. Für Default value (Standardwert) geben Sie **`LaunchedBySsmAutomation`** ein. Dadurch wird der Instance der Schlüsselpaarwert des Tags `Name:LaunchedBySsmAutomation` hinzugefügt.
5. Geben Sie unter Description (Beschreibung) Folgendes ein.

```
(Optional) The tag value to add to the instance. The default value is
 LaunchedBySsmAutomation.
```

9. Wählen Sie Add a parameter (Parameter hinzufügen), um den dritten Parameter **`instanceType`** zu erstellen, und geben Sie folgende Informationen ein.

1. Geben Sie unter Parameter name (Parametername) **`instanceType`** ein.
2. Wählen Sie für Type (Typ) die Option **`String`** aus.
3. Wählen Sie unter Required (Erforderlich) die Option No.

4. Für Default value (Standardwert) geben Sie **`t2.micro`** ein.
5. Geben Sie unter Parameter description (Parameterbeschreibung) Folgendes ein.

```
(Optional) The instance type to use for the instance. The default value is
 t2.micro.
```

10. Erweitern Sie Target type (Zieltyp) und wählen Sie **`"/"`**.
11. (Optional) Erweitern Sie Document tags (Dokument-Tags), um Ressourcen-Tags auf Ihr Runbook anzuwenden. Geben Sie für Tag key (Tag-Schlüssel **`Purpose`** und für Tag value (Tag-Wert) **`LaunchInstanceAndCheckState`** ein.
12. Führen Sie im Abschnitt Step 1 (Schritt 1) die folgenden Schritte aus.

    1. Geben Sie unter Step name (Schrittname) diesen beschreibenden Schrittnamen für den ersten Schritt der Automatisierung ein: **`LaunchEc2Instance`**.
    2. Wählen Sie unter Action type (Aktionstyp) die Option Run a script (Skript ausführen) (**`aws:executeScript`**).
    3. Geben Sie unter Description (Beschreibung) eine Beschreibung für den Automation-Schritt ein, wie etwa folgende.

```
**About This Step**

   This step first launches an EC2 instance using the ```aws:executeScript```
 action and the provided script.
```

    4. Erweitern Sie Inputs (Eingaben).
    5. Wählen Sie für Runtime (Laufzeit) die Laufzeitsprache aus, die zum Ausführen des bereitgestellten Skripts verwendet werden soll.
    6. Geben Sie unter Handler **`launch_instance`** ein. Dies ist der Funktionsname, der im folgenden Skript deklariert wird.

> **Note**
>
> Dies ist nicht erforderlich für PowerShell.

    7. Ersetzen Sie für Script (Skript) den Standardinhalt durch Folgendes. Stellen Sie sicher, dass das Skript dem entsprechenden Laufzeitwert entspricht.

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. Erweitern Sie Additional inputs (Zusätzliche Eingaben).
9. Wählen Sie für Eingabename die Option InputPayload. Geben Sie unter Input value (Eingabewert) die folgenden YAML-Daten ein.

```
image_id: "{{ imageId }}"
    tag_value: "{{ tagValue }}"
    instance_type: "{{ instanceType }}"
```

13. Erweitern Sie Outputs (Ausgänge) und gehen Sie folgendermaßen vor:
    - Geben Sie unter Name **`payload`** ein.
    - Geben Sie für Selector (Selektor) **`$.Payload`** ein.
    - Wählen Sie für Type (Typ) die Option `StringMap` aus.
14. Klicken Sie auf Schritt hinzufügen, um dem Runbook einen zweiten Schritt hinzuzufügen. Der zweite Schritt fragt den Status der in Schritt 1 gestarteten Instance ab und wartet, bis der zurückgegebene Status `ok` ist.
15. Gehen Sie im Abschnitt Step 2 (Schritt 2) folgendermaßen vor.
    1. Geben Sie unter Step name (Schrittname) diesen beschreibenden Namen für den zweiten Schritt der Automatisierung ein: **`WaitForInstanceStatusOk`**.
    2. Wählen Sie unter Action type (Aktionstyp) die Option Run a script (Skript ausführen) (**`aws:executeScript`**).
    3. Geben Sie unter Description (Beschreibung) eine Beschreibung für den Automation-Schritt ein, wie etwa folgende.

```
**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. Bei Runtime (Laufzeit) wählen Sie die Laufzeitsprache für die Ausführung des bereitgestellten Skripts verwendet werden soll.
5. Geben Sie unter Handler **`poll_instance`** ein. Dies ist der Funktionsname, der im folgenden Skript deklariert wird.

   > **Note**
   >
   > Dies ist nicht erforderlich für PowerShell.

6. Ersetzen Sie für Script (Skript) den Standardinhalt durch Folgendes. Stellen Sie sicher, dass das Skript dem entsprechenden Laufzeitwert entspricht.

   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. Erweitern Sie Additional inputs (Zusätzliche Eingaben).
8. Wählen Sie für Eingabename die Option InputPayload. Geben Sie unter Input value (Eingabewert) Folgendes ein:

```
{{ LaunchEc2Instance.payload }}
```

16. Wählen Sie Create automation (Automation erstellen), um das Runbook zu speichern.

Verwenden von Skripten in Runbooks

Automation-Runbooks unterstützen das Ausführen von Skripts im Rahmen der Automatisierung. Automation ist eine Funktion von AWS Systems Manager. Mithilfe von Runbooks können Sie Skripts direkt in AWS ausführen, ohne eine separate Datenverarbeitungsumgebung zum Ausführen Ihrer Skripts zu erstellen. Da Runbooks Skriptschritte neben anderen Automation-Schritttypen wie Genehmigungen ausführen können, haben Sie in kritischen oder unklaren Situationen die

Möglichkeit, manuell einzugreifen. Sie können die Ausgabe von `aws:executeScript`-Aktionen in Ihren Runbooks zu Amazon CloudWatch Logs verwenden. Weitere Informationen finden Sie unter Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs.

Berechtigungen für die Verwendung von Runbooks

Um ein Runbook zu verwenden, muss Systems Manager die Berechtigungen einer AWS Identity and Access Management(IAM)-Rolle verwenden. Die Methode, die Automation verwendet, um zu bestimmen, von welcher Rolle die Berechtigungen verwendet werden, hängt von einigen Faktoren und davon ab, ob ein Schritt die `aws:executeScript`-Aktion verwendet.

Für Runbooks, die `aws:executeScript` nicht verwenden, verwendet Automation eine von zwei Berechtigungsquellen:

- Die Berechtigungen einer IAM-Service-Rolle oder einer Assume-Rolle, die im Runbook angegeben oder als Parameter übergeben wird.
- Wenn keine IAM-Servicerolle angegeben ist, werden die Berechtigungen des IAM-Benutzers verwendet, der die Automatisierung gestartet hat.

Wenn ein Schritt in einem Runbook die `aws:executeScript`-Aktion enthält, ist jedoch immer eine IAM-Service-Rolle (Rolle übernehmen) erforderlich, wenn das für die Aktion angegebene Python- oder PowerShell-Skript jegliche AWS-API-Operationen aufruft. Automation prüft diese Rolle in der folgenden Reihenfolge:

- Die Berechtigungen einer IAM-Service-Rolle oder einer Assume-Rolle, die im Runbook angegeben oder als Parameter übergeben wird.
- Wenn keine Rolle gefunden wird, versucht Automation, das für `aws:executeScript` angegebene Python- oder PowerShell-Skript ohne Berechtigungen auszuführen. Wenn das Skript einen AWS-API-Vorgang (z. B. die Amazon EC2 `CreateImage`-Operation) aufruft oder versucht, eine Aktion an einer AWS-Ressource (z. B. eine EC2-Instance) durchzuführen, schlägt der Schritt, der das Skript enthält, fehl und Systems Manager gibt eine Fehlermeldung zurück, die den Fehler meldet.

Hinzufügen von Skripts zu Runbooks

Sie können Skripts zu Runbooks hinzufügen, indem Sie das Skript inline als Teil eines Schritts in das Runbook einfügen. Sie können Skripts auch an das Runbook anhängen, indem Sie die Skripts von Ihrem lokalen Computer hochladen oder einen Amazon Simple Storage Service (Amazon S3)-Bucket angeben, in dem sich die Skripts befinden. Nachdem ein Schritt abgeschlossen ist, in dem ein Skript

ausgeführt wird, steht die Ausgabe des Skripts als JSON-Objekt zur Verfügung, das Sie dann als Eingabe für nachfolgende Schritte im Runbook verwenden können.

Skripteinschränkungen für Runbooks

Runbooks erzwingen ein Limit von fünf Dateianhängen. Skripts können entweder in Form eines Python-Skripts (.py), eines PowerShell Core-Skripts (.ps1) oder als Inhalt einer ZIP-Datei angehängt werden.

Verwendung bedingter Anweisungen in Runbooks

Standardmäßig werden die Schritte, die Sie im Abschnitt `mainSteps` eines Runbooks definieren, nacheinander ausgeführt. Wenn eine Aktion abgeschlossen ist, beginnt die nächste im Abschnitt `mainSteps` angegebene Aktion. Wenn eine Aktion nicht erfolgreich ausgeführt wird, schlägt (standardmäßig) die gesamte Automatisierung fehl. Sie können die Automation-Aktion `aws:branch` und die in diesem Abschnitt beschriebenen Optionen für das Runbook zum Erstellen von Automatisierungen verwenden, die bedingte Verzweigungen durchführen. Dies bedeutet, dass Sie Automatisierungen erstellen können, die zu einem anderen Schritt springen, nachdem verschiedene Optionen bewertet wurden oder dynamisch auf Änderungen beim Abschluss eines Schrittes reagieren. Hier finden Sie eine Liste der Optionen, die Sie verwenden können, um dynamische Automatisierungen zu erstellen.

- **`aws:branch`**: Diese Automatisierungsaktion erlaubt das Erstellen einer dynamischen Automatisierung, die mehrere Auswahlmöglichkeiten in einem einzigen Schritt evaluiert und dann auf der Grundlage dieser Evaluierung zu einem anderen Schritt in dem Runbook springt.
- **`nextStep`**: Diese Option gibt an, welcher Schritt in einer Automatisierung nach dem erfolgreichem Abschluss eines Schritts als nächster auszuführen ist.
- **`isEnd`**: Diese Option stoppt eine Automatisierung am Ende eines bestimmten Schrittes. Der Standardwert für diese Option ist "false".
- **`isCritical`**: Diese Option bezeichnet einen Schritt als kritisch für den erfolgreichen Abschluss der Automatisierung. Wenn ein Schritt mit dieser Bezeichnung fehlschlägt, meldet Automation den Endstatus der Automatisierung als `Failed`. Der Standardwert für diese Option ist `true`.
- **`onFailure`**: Diese Option gibt an, ob die Automatisierung bei einem Fehler abgebrochen, fortgesetzt oder bis zu einem bestimmten Schritt übersprungen werden soll. Der Standardwert für diese Option ist "abort".

Der folgende Abschnitt beschreibt die Automation-Aktion `aws:branch`. Weitere Informationen über die Optionen `nextStep`, `isEnd`, `isCritical` und `onFailure` finden Sie unter Beispiel `aws:branch`-Runbooks.

Arbeiten mit der **`aws:branch`**-Aktion

Die Aktion `aws:branch` bietet die dynamischsten Optionen für bedingte Verzweigungen für Automatisierungen. Wie bereits erwähnt, erlaubt diese Aktion, dass Ihre Automatisierung mehrere Bedingungen in einem einzigen Schritt evaluiert und dann auf der Grundlage der Ergebnisse dieser Bewertung zu einem neuen Schritt springt. Die Aktion `aws:branch` funktioniert wie eine `IF-ELIF-ELSE`-Anweisung beim Programmieren.

Hier ist ein YAML-Beispiel für einen `aws:branch`-Schritt:

```
- name: ChooseOSforCommands
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runPowerShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Linux
    Default:
      PostProcessing
```

Wenn Sie die Aktion `aws:branch` für einen Schritt angeben, geben Sie die `Choices` an, die die Automatisierung evaluieren muss. Die Automatisierung kann `Choices` auf der Grundlage des Parameters evaluieren, den Sie im Abschnitt `Parameters` des Runbooks angegeben haben. Die Automatisierung kann `Choices` auch auf der Grundlage der Ausgabe eines vorherigen Schritts evaluieren.

Die Automatisierung evaluiert jede Auswahl mithilfe eines booleschen Ausdrucks. Wenn die Evaluierung zu dem Schluss kommt, dass die erste Auswahl `true` ist, springt die Automatisierung zum nächsten Schritt für diese Auswahl. Wenn die Auswertung zu dem Schluss kommt, dass die erste Auswahl `false` ist, evaluiert die Automatisierung die nächste Auswahl. Wenn Ihr Schritt drei oder mehr `Choices` beinhaltet, evaluiert die Automatisierung die Auswahlen nacheinander, bis eine Auswahl als `true` evaluiert wird. Die Automatisierung springt dann zu dem für die als `true` evaluierte Auswahl angegebenen Schritt.

Wenn keine `Choices` als `true` evaluiert werden, prüft die Automatisierung, ob der Schritt einen `Default`-Wert enthält. Ein `Default`-Wert definiert einen Schritt, zu dem die Automatisierung springen soll, wenn keine der Auswahlmöglichkeiten als `true` evaluiert wird. Wenn kein `Default`-Wert für den Schritt definiert ist, verarbeitet die Automatisierung den nächsten Schritt in dem Runbook.

Hier ist ein `aws:branch` Schritt in YAML namens SfromParameterChooSEO. Der Schritt beinhaltet zwei `Choices`: (`NextStep: runWindowsCommand`) und (`NextStep: runLinuxCommand`). Die Automatisierung evaluiert diese `Choices`, um zu bestimmen, welcher Befehl für das entsprechende Betriebssystem ausgeführt werden soll. Die `Variable` für jede Auswahl verwendet `{{OSName}}`. Dabei handelt es sich um einen Parameter, den der Autor des Runbooks im Abschnitt `Parameters` des Runbooks festgelegt hat.

```
mainSteps:
- name: chooseOSfromParameter
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runWindowsCommand
      Variable: "{{OSName}}"
      StringEquals: Windows
    - NextStep: runLinuxCommand
      Variable: "{{OSName}}"
      StringEquals: Linux
```

Hier ist ein **`aws:branch`** Schritt in YAML namens chooSEO. SfromOutput Der Schritt beinhaltet zwei `Choices`: (`NextStep: runPowerShellCommand`) und (`NextStep: runShellCommand`). Die Automatisierung evaluiert diese `Choices`, um zu bestimmen, welcher Befehl für das entsprechende Betriebssystem ausgeführt werden soll. Die `Variable` für jede Auswahl verwendet `{{GetInstance.platform}}`. Dies ist die Ausgabe aus einem früheren Schritt in dem Runbook. Dieses Beispiel enthält auch eine Option mit dem Namen `Default`. Wenn die Automatisierung beide `Choices` evaluiert und keine davon `true` ist, springt die Automatisierung zu einem Schritt mit dem Namen `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
```

Erstellen eines **aws:branch**-Schritts in einem Runbook

Wenn Sie einen `aws:branch`-Schritt in einem Runbook erstellen, definieren Sie die `Choices`, die die Automatisierung evaluieren soll, um festzustellen, zu welchem Schritt die Automatisierung dann springen soll. Wie bereits erwähnt, werden `Choices` mit einem booleschen Ausdruck evaluiert. Jede Auswahl muss die folgenden Optionen definieren:

- NextStep: Der nächste Schritt im Runbook, der verarbeitet werden muss, wenn die angegebene Option. `true`
- Variable: Geben Sie entweder den Namen eines Parameters an, der im `Parameters` Abschnitt des Runbooks definiert ist, eine im `Variables` Abschnitt definierte Variable, oder geben Sie ein Ausgabeobjekt aus einem vorherigen Schritt an.

  Geben Sie Variablenwerte mithilfe des folgenden Formulars an.

  `Variable: "{{variable name}}"`

  Geben Sie Parameterwerte mithilfe des folgenden Formulars an.

  `Variable: "{{parameter name}}"`

  Geben Sie Ausgabeobjektvariablen in der folgenden Form an.

  `Variable: "{{previousStepName.outputName}}"`

  > **Note**
  >
  > Das Erstellen der Ausgabevariable wird im nächsten Abschnitt ausführlicher beschrieben: Informationen zum Erstellen der Ausgabevariable.

- Operation: Die Kriterien für die Evaluierung der Auswahl, etwa `StringEquals: Linux`. Die Aktion `aws:branch` unterstützt die folgenden Operationen:

Zeichenfolgenoperationen

- StringEquals
- EqualsIgnoreCase
- StartsWith
- EndsWith
- Enthält

Numerische Operationen

- NumericEquals
- NumericGreater
- NumericLesser
- NumericGreaterOrEquals
- NumericLesser
- NumericLesserOrEquals

Boolesche Operation

- BooleanEquals

> **Important**
>
> Wenn Sie ein Runbook erstellen, validiert das System alle Operationen im Runbook. Wenn eine Operation nicht unterstützt wird, gibt das System einen Fehler aus, wenn Sie versuchen, das Runbook zu erstellen.

- Default: Geben Sie einen Rückfallschritt an, zu dem die Automatisierung springen soll, wenn keine der `Choices` `true` ist.

> **Note**
>
> Wenn Sie keinen `Default`-Wert angeben möchten, können Sie die `isEnd`-Option angeben. Wenn keine der `Choices` `true` ist und kein `Default`-Wert angegeben ist, wird die Automatisierung am Ende des Schrittes angehalten.

Verwenden Sie die folgenden Vorlagen für die Konstruktion des Schrittes `aws:branch` in Ihrem Runbook. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

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"
```

```
        }
      }
    ]
}
```

Informationen zum Erstellen der Ausgabevariable

Um eine `aws:branch`-Auswahl zu erstellen, die auf die Ausgabe eines vorherigen Schrittes verweist, müssen Sie den Namen des vorherigen Schrittes und den des Ausgabefeldes angeben. Anschließend kombinieren Sie die Namen des Schrittes und des Feldes im folgenden Format.

`Variable: "{{previousStepName.outputName}}"`

Beispielsweise hat der erste Schritt im folgenden Beispiel den Namen `GetInstance`. Dann gibt es unter `outputs` ein Feld mit dem Namen `platform`. Im zweiten Schritt (`ChooseOSforCommands`) möchte der Autor auf die Ausgabe des Plattform-Feldes als Variable verweisen. Um die Variable zu erstellen, kombinieren Sie einfach den Schrittnamen (GetInstance) und den Namen des Ausgabefeldes (Plattform), um sie zu erstellen`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
```

Hier ist ein Beispiel, das zeigt, wie *„Variable“: „{{ descriptionInstance.Platform }}“* aus dem vorherigen Schritt und der Ausgabe erstellt wird.

```
- 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
```

Beispiel **`aws:branch`**-Runbooks

Hier sind einige Beispiele für Runbooks, die `aws:branch` verwenden.

Beispiel 1: Verwendung von **`aws:branch`** mit einer Ausgabevariablen zur Ausführung von Befehlen auf der Grundlage des Betriebssystemtyps

Im ersten Schritt dieses Beispiels (`GetInstance`) verwendet der Runbook-Autor die `aws:executeAwsApi`-Aktion zum Aufrufen der `ssm` `DescribeInstanceInformation`-API-Operation. Der Autor verwendet diese Aktion, um den Typ des von einer Instance zu verwendenden

Betriebssystems zu bestimmen. Die Aktion `aws:executeAwsApi` gibt die Instance-ID und den Plattformtyp aus.

Im zweiten Schritt (`ChooseOSforCommands`) verwendet der Autor die Aktion `aws:branch` mit zwei `Choices` (`NextStep: runPowerShellCommand`) und (`NextStep: runShellCommand`). Die Automatisierung evaluiert das Betriebssystem der Instance anhand der Ausgabe des vorherigen Schritts (`Variable: "{{GetInstance.platform}}"`). Die Automatisierung springt zu einem Schritt für das angegebene Betriebssystem.

```
---
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
```

Beispiel 2: Verwendung von **`aws:branch`** mit einer Parametervariablen zur Ausführung von Befehlen auf der Grundlage des Betriebssystemtyps

Der Autor des Runbooks definiert verschiedene Parameteroptionen am Anfang des Runbooks im Abschnitt `parameters`. Ein Parameter hat den Namen `OperatingSystemName`. Im ersten Schritt (`ChooseOS`) verwendet der Autor die Aktion `aws:branch` mit zwei `Choices` (`NextStep: runWindowsCommand`) und (`NextStep: runLinuxCommand`). Die Variable für diese `Choices` verweist auf die im Parameter-Abschnitt angegebene Parameteroption (`Variable: "{{OperatingSystemName}}"`). Wenn der Benutzer dieses Runbook ausführt, gibt er zur Laufzeit einen Wert für `OperatingSystemName` an. Die Automatisierung verwendet den Laufzeitparameter während der Evaluierung der `Choices`. Die Automatisierung springt zu einem Schritt für das angegebene Betriebssystem auf der Grundlage des für `OperatingSystemName` angegebenen Laufzeitparameters.

```
---
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
```

Erstellen komplexer verzweigender Automatisierungen mit Operatoren

Sie können Automatisierungen mit komplexen Verzweigungen erstellen, indem Sie die Operatoren `And`, `Or` und `Not` in Ihren `aws:branch`-Schritten verwenden.

Der „Und“-Operator

Verwenden Sie den `And`-Operator, wenn Sie wünschen, dass mehrere Variablen für eine Auswahl `true` sind. Im folgenden Beispiel wird die erste Wahl darauf evaluiert, ob eine Instance `running` ist und das Betriebssystem `Windows` verwendet. Wenn die Evaluierung beider dieser Variablen „true“ ergibt, springt die Automatisierung zum Schritt `runPowerShellCommand`. Wenn eine oder mehrere der Variablen `false` ist, evaluiert die Automatisierung die Variablen für die zweite Auswahl.

```
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
```

Der „Oder“-Operator

Verwenden Sie den `Or`-Operator, wenn Sie wünschen, eine beliebige von mehreren Variablen für eine Auswahl „true“ ist. Im folgenden Beispiel wird die erste Auswahl darauf evaluiert, ob eine

Parameterzeichenfolge `Windows` ist, und ob die Ausgabe eines AWS Lambda -Schrittes „true“ ist. Wenn die Evaluierung feststellt, dass eine dieser Variablen „true“ ist, springt die Automatisierung zum Schritt `RunPowerShellCommand`. Wenn beide Variablen „false“ sind, evaluiert die Automatisierung die Variablen für die zweite Auswahl.

```
- Or:
  - Variable: "{{parameter1}}"
    StringEquals: Windows
  - Variable: "{{BooleanParam1}}"
    BooleanEquals: true
  NextStep: RunPowershellCommand
- Or:
  - Variable: "{{parameter2}}"
    StringEquals: Linux
  - Variable: "{{BooleanParam2}}"
    BooleanEquals: true
  NextStep: RunShellScript
```

Der „Nicht“-Operator

Verwenden Sie den `Not`-Operator, wenn zu einem Schritt gesprungen werden soll, wenn eine Variable nicht „true“ ist. Im folgenden Beispiel wird die erste Auswahl danach evaluiert, ob eine Parameterzeichenfolge `Not Linux` ist. Wenn die Evaluierung feststellt, dass die Variable nicht „Linux“ ist, springt die Automatisierung zum Schritt `sleep2`. Wenn die Evaluierung der ersten Auswahl feststellt, dass sie Linux ist, evaluiert die Automatisierung die nächste Auswahl.

```
mainSteps:
- name: switch
  action: aws:branch
  inputs:
    Choices:
    - NextStep: sleep2
      Not:
        Variable: "{{testParam}}"
        StringEquals: Linux
    - NextStep: sleep1
      Variable: "{{testParam}}"
      StringEquals: Windows
    Default:
      sleep3
```

Beispiele für die Verwendung von bedingten Optionen

Dieser Abschnitt enthält verschiedene Beispiele für die Verwendung dynamischer Optionen in einem Runbook. Jedes Beispiel in diesem Abschnitt erweitert das nachfolgende Runbook. Dieses Runbook verfügt über zwei Aktionen. Die erste Aktion hat den Namen `InstallMsiPackage`. Sie verwendet die Aktion `aws:runCommand` zur Installation einer Anwendung auf einer Windows Server-Instance. Die zweite Aktion hat den Namen `TestInstall`. Sie verwendet die Aktion `aws:invokeLambdaFunction` zum Ausführen eines Tests der installierten Anwendung, sofern die Anwendung erfolgreich installiert wurde. Der erste Schritt gibt `onFailure: Abort` an. Dies bedeutet, dass die Ausführung der Automatisierung vor dem zweiten Schritt gestoppt wird, wenn die Anwendung nicht erfolgreich installiert wird.

Beispiel 1: Runbook mit zwei linearen Aktionen

```
---
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
...
```

Erstellen einer dynamischen Automatisierung, die anhand der Option **`onFailure`** zu verschiedenen Schritten springt

Im folgenden Beispiel werden die Optionen `onFailure: step:`*`step name`*, `nextStep` und `isEnd` zur Erstellung einer dynamischen Automatisierung verwendet. Wenn in diesem Beispiel die `InstallMsiPackage` Aktion fehlschlägt, springt die Automatisierung zu einer Aktion namens PostFailure(`onFailure: step:PostFailure`), um eine AWS Lambda Funktion auszuführen, die eine Aktion ausführt, falls die Installation fehlschlägt. Wenn die Installation erfolgreich ist, springt die Automatisierung zur TestInstall Aktion () über. `nextStep: TestInstall` Die Schritte `TestInstall` und `PostFailure` verwenden die Option `isEnd` (`isEnd: true`), so dass die Automatisierung abschließt, wenn einer dieser Schritte abgeschlossen ist.

> **Note**
>
> Die Verwendung der Option `isEnd` im letzten Schritt des Abschnitts `mainSteps` ist optional. Wenn der letzte Schritt nicht zu anderen Schritten springt, stoppt die Automatisierung nach der Ausführung der Aktion im letzten Schritt.

Beispiel 2: Eine dynamische Automatisierung, die zu verschiedenen Schritten springt

```
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**
>
> Vor der Verarbeitung eines Runbooks überprüft das System, dass das Runbook keine Endlosschleife erstellt. Wenn eine Endlosschleife erkannt wird, gibt Automation einen Fehler und einen Kreis-Trace zurück, aus dem hervorgeht, welche Schritte die Schleife erzeugen.

Erstellen einer dynamischen Automatisierung, die entscheidende Schritte definiert

Sie können angeben, dass ein Schritt für den Erfolg der Automatisierung entscheidend ist. Wenn ein solcher kritischer Schritt fehlschlägt, meldet Automation den Status der Automatisierung als `Failed`. Dies gilt auch dann, wenn ein oder mehrere Schritte erfolgreich ausgeführt wurden. Im folgenden Beispiel identifiziert der Benutzer den Schritt, falls der VerifyDependenciesInstallMsiPackageSchritt fehlschlägt ()on`Failure: step:VerifyDependencies`. Der Benutzer gibt an, dass der Schritt `InstallMsiPackage` nicht kritisch ist (`isCritical: false`). In diesem Beispiel gilt: Wenn die Anwendung nicht installiert werden konnten, verarbeitet Automation den Schritt `VerifyDependencies`, um zu bestimmen, ob eine oder mehrere Abhängigkeiten fehlen, was dazu führte, dass die Anwendung nicht installiert werden konnte.

Beispiel 3: Definieren von kritischen Schritten für die Automatisierung

```
---
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
...
```

Verwenden von Aktionsausgaben als Eingaben

Verschiedene Automatisierungs-Aktionen geben vordefinierte Ausgaben zurück. Sie können diese Ausgaben mithilfe des Formats `{{stepName.outputName}}` als Eingaben an spätere Schritte in Ihrem Runbook übergeben. Sie können benutzerdefinierte Ausgaben für Automatisierungs-Aktionen in Ihren Runbooks definieren. Auf diese Weise können Sie Skripts ausführen oder API-Operationen für andere AWS-Services einmal aufrufen, sodass Sie die Werte als Eingaben in späteren Aktionen wiederverwenden können. Parametertypen in Runbooks sind statisch. Dies bedeutet, dass der Parametertyp nicht geändert werden kann, nachdem er definiert wurde. Um eine Schrittausgabe zu definieren, geben Sie die folgenden Felder an:

- Name: (Erforderlich) Der Ausgabename, der in späteren Schritten verwendet wird, um auf den Ausgabewert zu verweisen.
- Selektor: (Erforderlich) Der JSONPath-Ausdruck, der verwendet wird, um den Ausgabewert zu bestimmen.
- Typ: (Optional) Der Datentyp des Werts, der vom Auswahlfeld zurückgegeben wird. Gültige Typwerte sind `String`, `Integer`, `Boolean`, `StringList`, `StringMap`, `MapList`. Der Standardwert ist `String`.

Wenn der Wert einer Ausgabe nicht dem von Ihnen angegebenen Datentyp entspricht, versucht Automation, den Datentyp zu konvertieren. Wenn der zurückgegebene Wert beispielsweise ein `Integer` ist, der angegebene `Type` jedoch ein `String` ist, ist der endgültige Ausgabewert ein `String`-Wert. Die folgenden Typkonvertierungen werden unterstützt:

- `String`-Werte können in `StringList`, `Integer` und `Boolean` umgewandelt werden.
- `Integer`-Werte können in `String` und `StringList` umgewandelt werden.
- `Boolean`-Werte können in `String` und `StringList` umgewandelt werden.

- `StringList`-, `IntegerList`-, oder `BooleanList`-Werte, die ein Element enthalten, können in `String`, `Integer` oder `Boolean` umgewandelt werden.

Bei der Verwendung von Parametern oder Ausgaben mit Automatisierungs-Aktionen kann der Datentyp nicht dynamisch innerhalb der Eingabe einer Aktion geändert werden.

Hier ist ein Beispiel-Runbook, das veranschaulicht, wie Sie Aktionsausgaben definieren und auf den Wert als Eingabe für eine spätere Aktion verweisen. Die Runbooks tun Folgendes:

- Verwendet die `aws:executeAwsApi` Aktion , um die Amazon EC2 DescribeImages -API-Operation aufzurufen, um den Namen eines bestimmten Windows Server 2016 abzurufenAMI. Es gibt die Image-ID als `ImageId` aus.
- Verwendet die `aws:executeAwsApi` Aktion , um die Amazon EC2 RunInstances -API-Operation aufzurufen und eine Instance zu starten, die die `ImageId` aus dem vorherigen Schritt verwendet. Es gibt die Instance-ID als `InstanceId` aus.
- Verwendet die `aws:waitForAwsResourceProperty` Aktion , um den Amazon EC2 DescribeInstanceStatus API-Vorgang abzufragen und zu warten, bis die Instance den `running` Status erreicht. Die Aktion endet nach 60 Sekunden durch Timeout. Der Schritt endet durch Timeout, wenn die Instance nach 60 Sekunden Abfrage nicht den Status `running` erreicht.
- Verwendet die `aws:assertAwsResourceProperty`-Aktion zum Aufrufen der Amazon EC2 `DescribeInstanceStatus`-API-Operation, um geltend zu machen, dass sich die Instance im `running`-Zustand befindet. Der Schritt schlägt fehl, wenn der Status der Instance nicht `running` ist.

```
---
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"
...
```

Sie können mit jeder der oben beschriebenen Automatisierungsaktionen eine bestimmte API-Operation aufrufen, indem Sie den Service-Namespace, den Namen der API-Operation, die Eingabeparameter und die Ausgabeparameter angeben. Eingaben werden von der ausgewählten API-Operation bestimmt. Sie können die API-Operationen (auch als Methoden bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise werden alle API-Vorgänge (Methoden) für Amazon Relational Database Service (Amazon RDS) auf der folgenden Seite aufgelistet: Amazon RDS-Methoden.

Sie können das Schema für jede Automatisierungsaktion an den folgenden Orten anzeigen:

- `aws:assertAwsResourceProperty` - Geltendmachung eines AWS-Ressourcenstatus oder Ereignisstatus
- `aws:executeAwsApi` – Aufrufen und Ausführen von AWS API-Operationen
- `aws:waitForAwsResourceProperty` - Warten Sie auf eine AWS-Ressourceneigenschaft

Die Schemata umfassen Beschreibungen der erforderlichen Felder für jede Aktion.

Verwenden des/der Selektor/PropertySelector Felder

Jede Automatisierungsaktion erfordert, dass Sie entweder eine Ausgabe `Selector` (für `aws:executeAwsApi`) oder einen `PropertySelector` (für `aws:assertAwsResourceProperty` und `aws:waitForAwsResourceProperty`) enthalten. Diese Felder werden verwendet, um die JSON-Antwort einer AWS API-Operation zu verarbeiten. Diese Felder verwenden Sie die JSONPath-Syntax.

Hier finden Sie ein Beispiel, das dieses Konzept für die Aktion `aws:executeAwsAPi` erläutert.

```
---
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"
...
```

Im `aws:executeAwsApi`-Schritt`getImageId` ruft die Automatisierung die `DescribeImages`-API-Operation auf und empfängt eine Antwort von `ec2`. Die Automatisierung wendet dann `Selector - "$.Images[0].ImageId"` auf die API-Antwort an und weist der `ImageId`-Ausgabevariablen den ausgewählten Wert zu. Weitere Schritte in dieser Automatisierung können den Wert von `ImageId` verwenden, indem `"{{ getImageId.ImageId }}"` angegeben wird.

Hier finden Sie ein Beispiel, das dieses Konzept für die Aktion `aws:waitForAwsResourceProperty` erläutert.

```
---
- 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
...
```

Im `aws:waitForAwsResourceProperty`-Schritt`waitUntilInstanceStateRunning` ruft die Automatisierung die `DescribeInstanceStatus`-API-Operation auf und empfängt eine Antwort von `ec2`. Die Automatisierung wendet dann `PropertySelector - "$.InstanceStatuses[0].InstanceState.Name"` auf die Antwort an und prüft, ob der angegebene zurückgegebene Wert einem Wert in der Liste `DesiredValues` entspricht (in diesem

Fall `running`). Der Schritt wiederholt den Prozess, bis die Antwort den Instance-Status `running` zurückgibt.

Verwenden von JSONPath in Runbooks

Ein JSONPath-Ausdruck ist eine Zeichenfolge, die mit „$“ beginnt. die zur Auswahl einer oder mehrerer Komponenten in einem JSON-Element verwendet wird. Die folgende Liste enthält Informationen zu JSONPath-Operatoren, die von Systems Manager Automation unterstützt werden:

- Dot-notated child (.): Verwendung mit einem JSON-Objekt. Dieser Operator wählt den Wert eines bestimmten Schlüssels aus.
- Deep-scan (..): Verwendung mit einem JSON-Element. Dieser Operator untersucht das JSON-Element Ebene für Ebene und wählt eine Liste von Werten mit dem spezifischen Schlüssel aus. Der Rückgabetyp dieses Operators ist immer ein JSON-Array. Im Kontext eines Ausgabetyps für Automatisierungsaktionen kann der Operator entweder StringList oder sein MapList.
- Array-Index ([ ]): Verwendung mit einem JSON-Array. Dieser Operator ruft den Wert eines bestimmten Index ab.
- Filter ([?(***expression***)]): Wird mit einem JSON-Array verwendet. Dieser Operator filtert JSON-Array-Werte, die den im Filterausdruck definierten Kriterien entsprechen. Filterausdrücke können nur die folgenden Operatoren verwenden: ==,! =, >, <, >= oder <=. Die Kombination mehrerer Filterausdrücke mit AND (&&) oder OR (||) wird nicht unterstützt. Der Rückgabetyp dieses Operators ist immer ein JSON-Array.

Sehen Sie sich zum besseren Verständnis der JSONPath-Operatoren die folgende JSON-Antwort von der ec2-API-Operation `DescribeInstances` an. Unter dieser Antwort sehen Sie verschiedene Beispiele mit unterschiedlichen Ergebnissen durch die Verwendung verschiedener JSONPath-Ausdrücke für die Antwort von der API-Operation `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": []
        }
    ]
}
```

JSONPath-Beispiel 1: Abrufen einer bestimmten Zeichenfolge aus einer JSON-Antwort

```
JSONPath:
$.Reservations[0].Instances[0].ImageId

Returns:
"ami-12345678"

Type: String
```

JSONPath-Beispiel 2: Abrufen eines bestimmten booleschen Werts aus einer JSON-Antwort

```
JSONPath:
$.Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.DeleteOnTermination

Returns:
true

Type: Boolean
```

JSONPath-Beispiel 3: Abrufen einer bestimmten Ganzzahl aus einer JSON-Antwort

```
JSONPath:
$.Reservations[0].Instances[0].State.Code

Returns:
16

Type: Integer
```

JSONPath Beispiel 4: Eine JSON-Antwort tief scannen und dann alle Werte für VolumeId als abrufen StringList

```
JSONPath:
$.Reservations..BlockDeviceMappings..VolumeId

Returns:
[
   "vol-000000000000",
   "vol-111111111111"
]
```

```
Type: StringList
```

JSONPath-Beispiel 5: Abrufen eines bestimmten BlockDeviceMappings Objekts als StringMap

```
JSONPath:
$.Reservations[0].Instances[0].BlockDeviceMappings[0]

Returns:
{
   "Ebs" : {
      "DeleteOnTermination" : true,
      "Status" : "attached",
      "VolumeId" : "vol-000000000000"
   },
   "DeviceName" : "/dev/xvda"
}

Type: StringMap
```

JSONPath Beispiel 6: Eine JSON-Antwort gründlich scannen und dann alle State-Objekte als abrufen MapList

```
JSONPath:
$.Reservations..Instances..State

Returns:
[
   {
      "Code" : 16,
      "Name" : "running"
   },
   {
      "Code" : 80,
      "Name" : "stopped"
   }
]

Type: MapList
```

JSONPath Beispiel 7: Nach Instances im Status **`running`** filtern

```
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
```

JSONPath Beispiel 8: Gibt die **`ImageId`** der Instances zurück, die sich nicht im **`running`**-Status befinden

```
JSONPath:
$.Reservations..Instances[?(@.State.Name != 'running')].ImageId

Returns:
[
  "ami-12345678"
]

Type: StringList | String
```

Erstellen von Webhook-Integrationen für Automation

Um während einer Automatisierung Nachrichten über Webhooks zu senden, erstellen Sie eine Integration. Integrationen können während einer Automatisierung mithilfe der neuen

Aktion `aws:invokeWebhook` in Ihrem Runbook aufgerufen werden. Wenn Sie noch keinen Webhook erstellt haben, finden Sie weitere Informationen unter Erstellen von Webhooks für Integrationen. Weitere Informationen über die Aktion `aws:invokeWebhook` finden Sie unter `aws:invokeWebhook` – Automation-Webhook-Integration aufrufen.

Wie in den folgenden Verfahren gezeigt, können Sie Integrationen über die Automation-Konsole von Systems Manager oder mit Ihrem bevorzugten Befehlszeilen-Tool erstellen.

Erstellen von Integrationen (Konsole)

So erstellen Sie eine Integration für Automation (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Registerkarte Integrations (Integrationen) aus.
4. Wählen Sie Add integration (Integration hinzufügen) und dann Webhook aus.
5. Geben Sie die erforderlichen Werte und optionale Werte ein, die Sie für die Integration einbeziehen möchten.
6. Wählen Sie Add (Hinzufügen) aus, um die Integration zu erstellen.

Erstellen von Integrationen (Befehlszeile)

Um eine Integration mit Befehlszeilen-Tools zu erstellen, muss der erforderliche `SecureString`-Parameter für eine Integration erstellt werden. Automation nutzt einen reservierten Namespace im Parameter Store, einer Funktion von Systems Manager, um Informationen über Ihre Integration zu speichern. Wenn Sie eine Integration über die AWS Management Console erstellen, übernimmt Automation diesen Vorgang für Sie. Nach dem Namespace geben Sie den Typ der zu erstellenden Integration und dann deren Namen an. Derzeit unterstützt Automation Integrationen vom Typ `webhook`.

Folgende Felder werden für Integrationen vom Typ `webhook` unterstützt:

- Beschreibung
- Header
- Nutzlast
- URL

Bevor Sie beginnen

Falls noch nicht erfolgt, installieren und konfigurieren Sie die AWS Command Line Interface (AWS CLI) oder AWS Tools for PowerShell. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

So erstellen Sie eine Integration für Automation (Befehlszeile)

- Führen Sie die folgenden Befehle aus, um den erforderlichen `SecureString`-Parameter für eine Integration zu erstellen. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen. Der Namespace `/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/webhook/` ist im Parameter Store für Integrationen reserviert. Im Namen des Parameters muss dieser Namespace verwendet werden, gefolgt vom Namen der Integration. Zum Beispiel `/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"}'
```

Erstellen von Webhooks für Integrationen

Beachten Sie beim Erstellen von Webhooks bei Ihrem Anbieter Folgendes:

- Das Protokoll muss HTTPS lauten.
- Benutzerdefinierte Anforderungs-Header werden unterstützt.
- Ein Standardanforderungstext kann angegeben werden.
- Der Standardanforderungstext kann überschrieben werden, wenn eine Integration mit der Aktion `aws:invokeWebhook` aufgerufen wird.

Behandeln von Timeouts in Runbooks

Die Eigenschaft `timeoutSeconds` wird von allen Automatisierungsaktionen gemeinsam genutzt. Sie können diese Eigenschaft verwenden, um den Ausführungstimeout-Wert für eine Aktion anzugeben. Außerdem können Sie die Auswirkung des Timeouts einer Aktion auf die Automatisierung und den gesamten Ausführungsstatus ändern. Zu diesem Zweck definieren Sie auch die gemeinsam genutzten Eigenschaften `onFailure` und `isCritical` für eine Aktion.

Je nach Anwendungsfall möchten Sie vielleicht, dass Ihre Automatisierung mit einer anderen Aktion fortgesetzt wird und der Gesamtstatus der Automation nicht betroffen ist, wenn es zum Timeout einer Aktion kommt. In diesem Beispiel geben Sie mit der Eigenschaft `timeoutSeconds` an, wie lange gewartet werden soll, bevor es zum Timeout der Aktion kommt. Anschließend geben Sie die Aktion oder den Schritt an, zu dem die Automatisierung bei einem Timeout übergehen soll. Geben Sie einen Wert im Format `step:`*`step name`* für die Eigenschaft `onFailure` anstelle des Standardwerts `Abort` an. Beim Timeout einer Aktion wird der Automatisierungs-Ausführungsstatus standardmäßig `Timed Out` lauten. Um zu verhindern, dass sich ein Timeout auf den Automatisierungs-Ausführungsstatus auswirkt, geben Sie `false` für die Eigenschaft `isCritical` an.

Das folgende Beispiel zeigt, wie die gemeinsam genutzten Eigenschaften für eine in diesem Szenario beschriebene Aktion definiert werden.

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"
}
```

Weitere Informationen zu Eigenschaften, die von allen Automatisierungsaktionen gemeinsam genutzt werden, finden Sie unter Von allen Aktionen gemeinsam genutzte Eigenschaften.

Referenz zu Systems Manager Automation

Um Sie bei Ihren ersten Schritten zu unterstützen, bietet AWS Systems Manager vordefinierte Runbooks. Diese Runbooks werden von Amazon Web Services AWS Support und AWS Config verwaltet. Die Runbook-Referenz beschreibt jede der vordefinierten Runbooks, die von Systems Manager, AWS Support und AWS Config bereitgestellt werden. Weitere Informationen finden Sie unter Referenz zu Systems Manager Automation.

Tutorials

Die folgenden Tutorials unterstützen Sie bei der Verwendung von AWS Systems Manager Automation zur Bewältigung gängiger Anwendungsfälle. Diese Tutorials veranschaulichen, wie Sie Ihre eigenen Runbooks, von Automation bereitgestellte vordefinierte Runbooks und andere Systems-Manager-Funktionen mit anderen AWS-Services verwenden.

Inhalt

- Aktualisieren von AMIs
  - Aktualisieren eines Linux AMI
  - Aktualisieren eines Linux AMI (AWS CLI)
  - Aktualisieren eines Windows Server-AMI
  - Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store
    - Aufgabe 1: Erstellen eines Parameters im Systems Manager-Parameter Store
    - Aufgabe 2: Erstellen einer IAM-Rolle für AWS Lambda
    - Aufgabe 3: Erstellen einer AWS Lambda -Funktion
    - Aufgabe 4: Erstellen eines Runbooks und Patchen des AMI
  - Aktualisierung AMIs mithilfe von Automation und Jenkins
  - Aktualisieren von AMIs für Auto-Scaling-Gruppen
    - Erstellen Sie das PatchAMI ASG-Runbook AndUpdate
- Verwendung von AWS Support-Self-Service-Runbooks
  - Ausführen des EC2Rescue-Tools auf nicht erreichbaren Instances
    - Funktionsweise
    - Bevor Sie beginnen
      - Gewähren von AWSSupport-EC2Rescue-Berechtigungen zum Durchführen von Aktionen auf Ihren Instances

        - Erteilen von Berechtigungen mithilfe von IAM-Richtlinien
        - Erteilen von Berechtigungen mithilfe einer Vorlage AWS CloudFormation
    - Ausführen der Automation
- Zurücksetzen von Passwörtern und SSH-Schlüsseln auf EC2-Instances
    - Funktionsweise
    - Bevor Sie beginnen
        - Erteilen von AWSSupport -EC2Rescue-Berechtigungen zur Durchführung von Aktionen auf Ihren Instances
            - Erteilen von Berechtigungen mithilfe von IAM-Richtlinien
            - Erteilen von Berechtigungen mithilfe einer Vorlage AWS CloudFormation
    - Ausführen der Automation
- Übergabe von Daten an Automation mithilfe von Eingangstransformatoren

Aktualisieren von AMIs

Die folgenden Tutorials erläutern, wie Sie Amazon Machine Image (AMIs) aktualisieren, um die neuesten Patches einzubeziehen.

Themen

- Aktualisieren eines Linux AMI
- Aktualisieren eines Linux AMI (AWS CLI)
- Aktualisieren eines Windows Server-AMI
- Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store
- Aktualisierung AMIs mithilfe von Automation und Jenkins
- Aktualisieren von AMIs für Auto-Scaling-Gruppen

Aktualisieren eines Linux AMI

Diese Vorgehensweise für Systems Manager Automation zeigt Ihnen, wie Sie die Konsole oder das AWS CLI - und das `AWS-UpdateLinuxAmi`-Runbook verwenden, um ein Linux-AMI mit den neuesten Patches der von Ihnen angegebenen Pakete zu aktualisieren. Automation ist eine Funktion von AWS Systems Manager. Das `AWS-UpdateLinuxAmi`-Runbook automatisiert auch die Installation zusätzlicher websitespezifischer Pakete und Konfigurationen. Mit dieser

exemplarischen Vorgehensweise können Sie eine Vielzahl von Linux-Distributionen aktualisieren, darunter CentOSUbuntu Server, RHEL, SLES oder Amazon Linux. AMIs Eine vollständige Liste der unterstützten Linux-Versionen finden Sie unter Patch Manager-Voraussetzungen.

Mit dem `AWS-UpdateLinuxAmi`-Runbook können Sie Aufgaben zur Imagewartung automatisieren, ohne das Runbook in JSON oder YAML erstellen zu müssen. Sie können das Runbook `AWS-UpdateLinuxAmi` verwenden, um die folgenden Arten von Aufgaben auszuführen.

- Aktualisieren Sie alle Verteilungspakete und jegliche Amazon-Software auf Amazon Linux, Red Hat Enterprise Linux, Ubuntu Server, SUSE Linux Enterprise Server, oder CentOS Amazon Machine Image (AMI). Dies ist das Runbook-Standardverhalten.
- Installieren Sie AWS Systems Manager SSM Agent auf einem vorhandenen Image, um Systems Manager Manager-Funktionen zu aktivieren, z. B. das Ausführen von Remotebefehlen mithilfe von AWS Systems Manager Run Command oder die Erfassung von Softwareinventar mithilfe von Inventar.
- Installieren Sie zusätzliche Softwarepakete.

Bevor Sie beginnen

Bevor Sie mit der Arbeit mit Runbooks beginnen, konfigurieren Sie Rollen und optional die Funktionen EventBridge für die Automatisierung. Weitere Informationen finden Sie unter Einrichten der Automatisierung. Für diese exemplarische Vorgehensweise müssen Sie außerdem den Namen eines AWS Identity and Access Management (IAM-) Instanzprofils angeben. Weitere Informationen zum Erstellen eines IAM-Instanzprofils finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

Das Runbook `AWS-UpdateLinuxAmi` akzeptiert die folgenden Eingabeparameter.

Parameter	Typ	Beschreibung
SourceAmiID	String	(Erforderlich) Die Quell-AMI-ID.
IamInstanceProfileName	String	(Erforderlich) Der Name der IAM-Instanzprofilrolle, die Sie unter Für Systems Manager erforderliche Instanzberechtigungen konfigurieren

Parameter	Typ	Beschreibung
		erstellt haben. Die Instance-Profilrolle erteilt der Automation die Berechtigung, auf Ihren Instances Aktionen durchzuführen, wie etwa das Ausführen von Befehlen oder das Starten und Beenden von Services. Das Runbook verwendet nur den Namen der Instance-Profilrolle. Wenn Sie den Amazon-Ressourcennamen (ARN) angeben, schlägt die Automatisierung fehl.
AutomationAssumeRolle	String	(Erforderlich) Der Name der IAM-Servicerolle, die Sie in Einrichten der Automatisierung erstellt haben. Mit der Servicerolle (auch als assume-Rolle bezeichnet) gestatten Sie der Automatisierung, Ihre IAM-Rolle zu übernehmen und in Ihrem Auftrag Aktionen auszuführen. Mit der Servicerolle gestatten Sie der Automation beispielsweise beim Ausführen der Aktion `aws:createImage` in einem Runbook, ein neues AMI zu erstellen. Für diesen Parameter muss der vollständige ARN angegeben werden.

Parameter	Typ	Beschreibung
TargetAmiName	String	(Optional) Der Name des neuen AMI nach seiner Erstellung. Der Standardn ame ist eine systemgenerierte Zeichenfolge, die die Quell-AMI-ID sowie Uhrzeit und Datum der Erstellung enthält.
InstanceType	String	(Optional) Der Typ der zu startenden Instance als Arbeitsbereich hosten. Die Instance-Typen sind je nach Region unterschiedlich. Der Standardtyp ist t2.micro.
PreUpdateDrehbuch	String	(Optional) Die URL eines Skripts, das ausgeführt werden muss, bevor Updates übernommen werden. Standard („none“) ist die Ausführung keines Skripts.
PostUpdateDrehbuch	String	(Optional) Die URL eines Skripts, das ausgeführt werden muss, nachdem Paketupdates angewendet werden. Standard („none“) ist die Ausführung keines Skripts.
IncludePackages	String	(Optional) Aktualisieren Sie nur diese benannten Pakete. Standardmäßig werden alle (\"all\") verfügbaren Updates übernommen.

Parameter	Typ	Beschreibung
ExcludePackages	String	(Optional) Namen der Pakete, die bei Updates unter allen Umständen zurückgehalten werden müssen. Standardm äßig wird kein (\"none\") Paket ausgeschlossen.

Automation-Schritte

Das `AWS-UpdateLinuxAmi`-Runbook umfasst standardmäßig die folgenden Automatisierungsaktionen.

Schritt 1: launchInstance (**`aws:runInstances`**-Aktion)

Dieser Schritt startet eine Instance mit Amazon Elastic Compute Cloud (Amazon EC2)-Benutzerdaten und eine IAM-Instance-Profilrolle. Userdata installiert je nach Betriebssystem den entsprechenden SSM Agent. Durch Installieren von SSM Agent können Sie Systems Manager-Funktionen verwenden, wie etwa Run Command, State Manager und Inventory.

Schritt 2: updateOSSoftware (**`aws:runCommand`**-Aktion)

Dieser Schritt führt die folgenden Befehle auf der gestarteten Instance aus:

- Lädt ein Update-Skript aus Amazon S3 herunter.
- Führt ein optionales Pre-Update-Skript aus.
- Aktualisiert Verteilungspakete und Amazon-Software.
- Führt ein optionales Post-Update-Skript aus.

Das Ausführungsprotokoll wird im Ordner /tmp gespeichert, damit es der Benutzer zu einem späteren Zeitpunkt ansehen kann.

Falls Sie eine bestimmte Reihe von Paketen aktualisieren möchten, können Sie die Liste mithilfe des `IncludePackages`-Parameters bereitstellen. Bei der Bereitstellung versucht das System nur diese Pakete und deren abhängige Objekte zu aktualisieren. Es werden keine weiteren Updates vorgenommen. Wenn standardmäßig keine include-Pakete festgelegt sind, aktualisiert das Programm alle verfügbaren Pakete.

Falls Sie eine bestimmte Reihe von Paketen von der Aktualisierung ausschließen möchten, können Sie die Liste mithilfe des `ExcludePackages`-Parameters bereitstellen. Wenn diese Pakete bereitgestellt werden, bleiben sie in ihrer aktuellen Version, unabhängig von anderen festgelegten Optionen. Wenn keine exclude-Pakete festgelegt sind, werden standardmäßig keine Pakete ausgeschlossen.

Schritt 3: StopInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt stoppt die aktualisierte Instance.

Schritt 4: CreateImage (**`aws:createImage`**-Aktion)

Dieser Schritt erstellt ein neues AMI mit einem aussagekräftigen Namen, der es mit der Quell-ID und dem Zeitpunkt der Erstellung verknüpft. Zum Beispiel: „AMIGeneriert von EC2 Automation am {{global:Date_Time}} von {{Id}}“, wobei DATE_TIME und SourceAmi SourceID Automatisierungsvariablen darstellen.

Schritt 5: TerminateInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt bereinigt die Automatisierung durch Beenden der ausgeführten Instance.

Output

Die Automatisierung gibt die neue AMI-ID als Ausgabe zurück.

> **Note**
>
> Standardmäßig erstellt das System eine temporäre Instance in der Standard-VPC (172.30.0.0/16), wenn Automation das `AWS-UpdateLinuxAmi`-Runbook ausführt. Wenn Sie die Standard-VPC gelöscht haben, erhalten Sie den folgenden Fehler:
> `VPC not defined 400`
> Zur Behebung dieses Problems erstellen Sie eine Kopie des `AWS-UpdateLinuxAmi`-Runbooks und geben eine Subnetz-ID an. Weitere Informationen finden Sie unter VPC nicht definiert 400.

So erstellen Sie ein gepatchtes AMImit der Automation (AWS Systems Manager)

1. AWS Systems Manager Öffnen Sie die Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.

3. Wählen Sie Execute automation (Automatisierung ausführen).
4. Wählen Sie in der Liste Automation-Dokument **`AWS-UpdateLinuxAmi`**.
5. Überprüfen Sie im Abschnitt Document details (Dokumentdetails), ob Document version (Dokumentversion) auf Default version at runtime (Standardversion bei Laufzeit) gesetzt ist.
6. Wählen Sie Weiter aus.
7. Wählen Sie im Abschnitt Execution mode (Ausführungsmodus) die Option Simple Execution (Einfache Ausführung) aus.
8. Geben Sie im Abschnitt Input parameters (Eingabeparameter) die Informationen ein, die Sie im Abschnitt Before You Begin (Bevor Sie beginnen) erfasst haben.
9. Wählen Sie Execute (Ausführen). Die Konsole zeigt den Status der Automation-Ausführung an.

Nach dem Abschluss der Automatisierung starten Sie eine Test-Instance über das aktualisierte AMI, um die Änderungen zu überprüfen.

> **Note**
>
> Falls ein Schritt in der Automatisierung fehlschlägt, werden die Informationen zu dem Fehler auf der Seite Automation Executions (Automation-Ausführungen) aufgelistet. Die Automatisierung ist so konzipiert, dass sie die temporäre Instance nach erfolgreichem Abschluss aller Aufgaben beendet. Wenn ein Schritt fehlschlägt, beendet das System die Instance möglicherweise nicht. Wenn also ein Schritt fehlschlägt, beenden Sie die temporäre Instance manuell.

Aktualisieren eines Linux AMI (AWS CLI)

Diese exemplarische Vorgehensweise zur AWS Systems Manager Automatisierung zeigt Ihnen, wie Sie das Runbook AWS Command Line Interface (AWS CLI) und das Systems Manager `AWS-UpdateLinuxAmi` Manager-Runbook verwenden, um ein Linux Amazon Machine Image (AMI) automatisch mit den neuesten Versionen der von Ihnen angegebenen Pakete zu patchen. Automatisierung ist eine Fähigkeit von. AWS Systems Manager Das `AWS-UpdateLinuxAmi`-Runbook automatisiert auch die Installation zusätzlicher websitespezifischer Pakete und Konfigurationen. Mit dieser exemplarischen Vorgehensweise können Sie eine Vielzahl von Linux-Distributionen aktualisieren, darunter CentOSUbuntu Server, RHEL, SLES oder Amazon Linux. AMIs Eine vollständige Liste der unterstützten Linux-Versionen finden Sie unter Patch Manager-Voraussetzungen.

Das `AWS-UpdateLinuxAmi`-Runbook ermöglicht Ihnen die Automatisierung von Image-Verwaltungsaufgaben ohne Erstellen des Runbooks in JSON oder YAML. Sie können das Runbook `AWS-UpdateLinuxAmi` verwenden, um die folgenden Arten von Aufgaben auszuführen.

- Aktualisieren Sie alle Distributionspakete und Amazon-Software auf einem Amazon Linux-Red Hat Enterprise Linux,Ubuntu Server, SLES- oder Cent-Betriebssystem Amazon Machine Image (AMI). Dies ist das Runbook-Standardverhalten.
- Installieren Sie AWS Systems Manager SSM Agent auf einem vorhandenen Image, um Systems Manager Manager-Funktionen zu aktivieren, z. B. das Ausführen von Remotebefehlen mithilfe von AWS Systems Manager Run Command oder die Erfassung von Softwareinventar mithilfe von Inventar.
- Installieren Sie zusätzliche Softwarepakete.

Bevor Sie beginnen

Bevor Sie mit der Arbeit mit Runbooks beginnen, konfigurieren Sie Rollen und optional die Funktionen EventBridge für die Automatisierung. Weitere Informationen finden Sie unter Einrichten der Automatisierung. Für diese exemplarische Vorgehensweise müssen Sie außerdem den Namen eines AWS Identity and Access Management (IAM-) Instanzprofils angeben. Weitere Informationen zum Erstellen eines IAM-Instanzprofils finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

Das Runbook `AWS-UpdateLinuxAmi` akzeptiert die folgenden Eingabeparameter.

Parameter	Typ	Beschreibung
SourceAmiID	String	(Erforderlich) Die Quell-AMI-ID. Mithilfe eines AWS Systems Manager Parameter Store öffentlichen Parameter s können Sie automatisch auf die neueste ID eines Amazon EC2 AMI für Linux verweisen . Weitere Informationen finden Sie unter Abfragen der neuesten Amazon AMI Linux-

Parameter	Typ	Beschreibung
		IDs mithilfe von AWS Systems ManagerParameter Store.
IamInstanceProfileName	String	(Erforderlich) Der Name der IAM-Instanzprofilrolle, die Sie unter Für Systems Manager erforderliche Instanzbe rechtigungen konfigurieren erstellt haben. Die Instance-Profilrolle erteilt der Automatio n die Berechtigung, auf Ihren Instances Aktionen durchzufü hren, wie etwa das Ausführen von Befehlen oder das Starten und Beenden von Services. Das Runbook verwendet nur den Namen der Instance-Profilrolle.

Parameter	Typ	Beschreibung
AutomationAssumeRolle	String	(Erforderlich) Der Name der IAM-Servicerolle, die Sie in Einrichten der Automatisierung erstellt haben. Mit der Servicerolle (auch als assume-Rolle bezeichnet) gestatten Sie der Automatisierung, Ihre IAM-Rolle zu übernehmen und in Ihrem Auftrag Aktionen auszuführen. Mit der Servicerolle gestatten Sie der Automation beispielsweise beim Ausführen der Aktion `aws:createImage` in einem Runbook, ein neues AMI zu erstellen. Für diesen Parameter muss der vollständige ARN angegeben werden.
TargetAmiName	String	(Optional) Der Name des neuen AMI nach seiner Erstellung. Der Standardname ist eine systemgenerierte Zeichenfolge, die die Quell-AMI-ID sowie Uhrzeit und Datum der Erstellung enthält.
InstanceType	String	(Optional) Der Typ der zu startenden Instance als Arbeitsbereich hosten. Die Instance-Typen sind je nach Region unterschiedlich. Der Standardtyp ist t2.micro.

Parameter	Typ	Beschreibung
PreUpdateDrehbuch	String	(Optional) Die URL eines Skripts, das ausgeführt werden muss, bevor Updates übernommen werden. Standard („none“) ist die Ausführung keines Skripts.
PostUpdateDrehbuch	String	(Optional) Die URL eines Skripts, das ausgeführt werden muss, nachdem Paketupdates angewendet werden. Standard („none“) ist die Ausführung keines Skripts.
IncludePackages	String	(Optional) Aktualisieren Sie nur diese benannten Pakete. Standardmäßig werden alle (\"all\") verfügbaren Updates übernommen.
ExcludePackages	String	(Optional) Namen der Pakete, die bei Updates unter allen Umständen zurückgehalten werden müssen. Standardm äßig wird kein (\"none\") Paket ausgeschlossen.

Automation-Schritte

Das `AWS-UpdateLinuxAmi`-Runbook enthält standardmäßig die folgenden Schritte.

Schritt 1: launchInstance (**`aws:runInstances`**-Aktion)

In diesem Schritt wird eine Instance gestartet, die Amazon Elastic Compute Cloud (Amazon EC2)-Benutzerdaten und eine IAM-Instance-Profilrolle verwendet. Userdata installiert je nach Betriebssystem den entsprechenden SSM-Agent. Durch Installieren von SSM Agent können

Sie Systems Manager-Funktionen verwenden, wie etwa Run Command, State Manager und Inventory.

Schritt 2: updateOSSoftware (**`aws:runCommand`**-Aktion)

Dieser Schritt führt die folgenden Befehle auf der gestarteten Instance aus:

- Lädt ein Update-Skript von Amazon Simple Storage Service (Amazon S3) herunter.
- Führt ein optionales Pre-Update-Skript aus.
- Aktualisiert Verteilungspakete und Amazon-Software.
- Führt ein optionales Post-Update-Skript aus.

Das Ausführungsprotokoll wird im Ordner /tmp gespeichert, damit es der Benutzer zu einem späteren Zeitpunkt ansehen kann.

Falls Sie eine bestimmte Reihe von Paketen aktualisieren möchten, können Sie die Liste mithilfe des `IncludePackages`-Parameters bereitstellen. Bei der Bereitstellung versucht das System nur diese Pakete und deren abhängige Objekte zu aktualisieren. Es werden keine weiteren Updates vorgenommen. Wenn standardmäßig keine include-Pakete festgelegt sind, aktualisiert das Programm alle verfügbaren Pakete.

Falls Sie eine bestimmte Reihe von Paketen von der Aktualisierung ausschließen möchten, können Sie die Liste mithilfe des `ExcludePackages`-Parameters bereitstellen. Wenn diese Pakete bereitgestellt werden, bleiben sie in ihrer aktuellen Version, unabhängig von anderen festgelegten Optionen. Wenn keine exclude-Pakete festgelegt sind, werden standardmäßig keine Pakete ausgeschlossen.

Schritt 3: StopInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt stoppt die aktualisierte Instance.

Schritt 4: CreateImage (**`aws:createImage`**-Aktion)

Dieser Schritt erstellt ein neues AMI mit einem aussagekräftigen Namen, der es mit der Quell-ID und dem Zeitpunkt der Erstellung verknüpft. Beispiel: „Von EC2 Automation am {{global:Date_Time}} von {{Id}} generiertes AMI“, wobei DATE_TIME und SourceAmi SourceID Automatisierungsvariablen darstellen.

Schritt 5: TerminateInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt bereinigt die Automatisierung durch Beenden der ausgeführten Instance.

Output

Die Automatisierung gibt die neue AMI-ID als Ausgabe zurück.

> **Note**
>
> Standardmäßig erstellt das System eine temporäre Instance in der Standard-VPC (172.30.0.0/16), wenn Automation das `AWS-UpdateLinuxAmi`-Runbook ausführt. Wenn Sie die Standard-VPC gelöscht haben, erhalten Sie den folgenden Fehler:
> `VPC not defined 400`
> Zur Behebung dieses Problems erstellen Sie eine Kopie des `AWS-UpdateLinuxAmi`-Runbooks und geben eine Subnetz-ID an. Weitere Informationen finden Sie unter VPC nicht definiert 400.

So erstellen Sie ein gepatchtes AMI mithilfe von Automation

1. Installieren und konfigurieren Sie AWS Command Line Interface ()AWS CLI, falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um das `AWS-UpdateLinuxAmi`-Runbook zu starten. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

   ```
   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'
   ```

   Der Befehl gibt eine Ausführungs-ID zurück. Kopieren Sie diese ID in die Zwischenablage. Sie werden diese ID zum Anzeigen des Status der Automatisierung verwenden.

   ```
   {
       "AutomationExecutionId": "automation execution ID"
   ```

```
}
```

3. Führen Sie den folgenden Befehl aus AWS CLI, um die Automatisierung mit dem anzuzeigen:

```
aws ssm describe-automation-executions
```

4. Führen Sie den folgenden Befehl aus, um Details über den Automatisierungsprozess anzuzeigen. Ersetzen Sie *automation execution ID* (Automatisierungs-Ausführungs-ID) mit Ihren eigenen Informationen.

```
aws ssm get-automation-execution --automation-execution-id automation execution ID
```

   Die Aktualisierung kann 30 Minuten oder länger in Anspruch nehmen.

   > **Note**
   >
   > Sie können auch den Status der Automatisierung in der Konsole überwachen. Wählen Sie in der Liste die Automatisierung, die Sie gerade ausgeführt haben, und wählen Sie dann die Registerkarte Steps (Schritte). Diese Registerkarte zeigt Ihnen den Status der Automatisierungsaktionen.

Nach dem Abschluss der Automatisierung starten Sie eine Test-Instance über das aktualisierte AMI, um die Änderungen zu überprüfen.

> **Note**
>
> Falls ein Schritt in der Automatisierung fehlschlägt, werden die Informationen zu dem Fehler auf der Seite Automation Executions (Automation-Ausführungen) aufgelistet. Die Automatisierung ist so konzipiert, dass sie die temporäre Instance nach erfolgreichem Abschluss aller Aufgaben beendet. Wenn ein Schritt fehlschlägt, beendet das System die Instance möglicherweise nicht. Wenn also ein Schritt fehlschlägt, beenden Sie die temporäre Instance manuell.

Aktualisieren eines Windows Server-AMI

Das Runbook `AWS-UpdateWindowsAmi` ermöglicht die Automatisierung von Image-Verwaltungsaufgaben auf Ihrem Amazon Windows Amazon Machine Image (AMI), ohne dass Sie

das Runbook in JSON oder YAML erstellen müssen. Dieses Runbook wird unterstützt für Windows Server 2008 R2 oder höher. Sie können das Runbook `AWS-UpdateWindowsAmi` verwenden, um die folgenden Arten von Aufgaben auszuführen.

- Installieren Sie alle Windows-Updates und aktualisieren Sie die Amazon-Software (Standardverhalten).
- Installieren Sie spezifische Windows-Updates und aktualisieren Sie die Amazon-Software.
- Passen Sie ein AMI mithilfe Ihrer Skripts an.

Bevor Sie beginnen

Bevor Sie mit Runbooks arbeiten, konfigurieren Sie Rollen für Automation, um eine `iam:PassRole`-Richtlinie hinzuzufügen, die auf den ARN des Instance-Profils verweist, dem Sie den Zugriff gewähren möchten. Konfigurieren Sie optional Amazon EventBridge for Automation, eine Funktion von AWS Systems Manager. Weitere Informationen finden Sie unter Einrichten der Automatisierung. Für diese exemplarische Vorgehensweise müssen Sie außerdem den Namen eines AWS Identity and Access Management (IAM-) Instance-Profils angeben. Weitere Informationen zum Erstellen eines IAM-Instanzprofils finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

> **Note**
>
> Updates für AWS Systems Manager SSM Agent werden in der Regel für verschiedene Regionen zu verschiedenen Zeiten angeboten. Wenn Sie ein AMI anpassen oder aktualisieren, verwenden Sie nur die für die Region, in der Sie arbeiten, veröffentlichten Quell-AMIs. Auf diese Weise stellen Sie sicher, dass Sie mit dem neuesten SSM Agent für diese Region arbeiten und vermeiden Kompatibilitätsprobleme.

Das Runbook `AWS-UpdateWindowsAmi` akzeptiert die folgenden Eingabeparameter.

Parameter	Typ	Beschreibung
SourceAmiId	String	(Erforderlich) Die Quell-AMI-ID. Sie können automatisch auf die neueste Windows Server-AMI mithilfe eines

Parameter	Typ	Beschreibung
		Systems ManagerParameter Store öffentlich-Parameter verweisen. Weitere Informationen finden Sie unter Query for the latest Windows AMI IDs using AWS Systems ManagerParameter Store.
SubnetId	String	(Optional) Das Subnetz, in dem Sie die temporäre Instance starten möchten. Sie müssen einen Wert für diesen Parameter angeben, wenn Sie Ihre Standard-VPC gelöscht haben.
IamInstanceProfileName	String	(Erforderlich) Der Name der IAM-Instanzprofilrolle, die Sie unter Für Systems Manager erforderliche Instanzberechtigungen konfigurieren erstellt haben. Die Instance-Profilrolle erteilt der Automation die Berechtigung, auf Ihren Instances Aktionen durchzuführen, wie etwa das Ausführen von Befehlen oder das Starten und Beenden von Services. Das Runbook verwendet nur den Namen der Instance-Profilrolle.

Parameter	Typ	Beschreibung
AutomationAssumeRolle	String	(Erforderlich) Der Name der IAM-Servicerolle, die Sie in Einrichten der Automatisierung erstellt haben. Mit der Servicerolle (auch als assume-Rolle bezeichnet) gestatten Sie der Automatisierung, Ihre IAM-Rolle zu übernehmen und in Ihrem Auftrag Aktionen auszuführen. Mit der Servicerolle gestatten Sie der Automation beispielsweise beim Ausführen der Aktion `aws:createImage` in einem Runbook, ein neues AMI zu erstellen. Für diesen Parameter muss der vollständige ARN angegeben werden.
TargetAmiName	String	(Optional) Der Name des neuen AMI nach seiner Erstellung. Der Standardname ist eine systemgenerierte Zeichenfolge, die die Quell-AMI-ID sowie Uhrzeit und Datum der Erstellung enthält.
InstanceType	String	(Optional) Der Typ der zu startenden Instance als Arbeitsbereich hosten. Die Instance-Typen sind je nach Region unterschiedlich. Der Standardtyp ist t2.medium.

Parameter	Typ	Beschreibung
PreUpdateDrehbuch	String	(Optional) Ein Skript, das ausgeführt werden muss, bevor das AMI aktualisiert wird. Geben Sie ein Skript im Runbook oder zur Laufzeit als Parameter an.
PostUpdateDrehbuch	String	(Optional) Ein Skript, das ausgeführt werden muss, nachdem das AMI aktualisiert wird. Geben Sie ein Skript im Runbook oder zur Laufzeit als Parameter an.
IncludeKbs	String	(Optional) Geben Sie mindestens eine Microsoft Knowledge Base (KB)-Artikel-ID an, die einbezogen werden soll. Sie können mehrere IDs anhand kommaseparierter Werte installieren. Gültige Formate: KB9876543 oder 9876543.
ExcludeKbs	String	(Optional) Geben Sie mindestens eine Microsoft Knowledge Base (KB)-Artikel-ID an, die ausgeschlossen werden soll. Sie können mehrere IDs anhand kommaseparierter Werte ausschließen. Gültige Formate: KB9876543 oder 9876543.

Parameter	Typ	Beschreibung
Kategorien	String	(Optional) Geben Sie mindestens eine Updatekategorie an. Sie können Kategorien anhand kommaseparierter Werte filtern. Optionen: Wichtiges Update, Sicherheitsupdate, Definitionsupdate, Update-Rollup, Service Pack, Tool, Update oder Treiber. Zu den gültigen Formaten gehört ein einzelner Eintrag. Beispiel: Wichtiges Update. Sie können auch eine kommaseparierte Liste angeben: Wichtiges Update,Sicherheitsupdate,Definitionsupdate.
SeverityLevels	String	(Optional) Geben Sie mindestens eine MSRC-Ebene an, die einem Update zugeordnet ist. Sie können Dringlichkeitsstufen anhand kommaseparierter Werte filtern. Optionen: Kritisch, Wichtig, Niedrige, Mittel oder Nicht angegeben. Zu den gültigen Formaten gehört ein einzelner Eintrag. Beispiel: Wichtig. Sie können auch eine kommaseparierte Liste angeben: Kritisch,Wichtig,Niedrig.

Automation-Schritte

Das `AWS-UpdateWindowsAmi`-Runbook enthält standardmäßig die folgenden Schritte.

Schritt 1: launchInstance (**`aws:runInstances`**-Aktion)

Dieser Schritt startet eine Instance mit einer IAM-Instance-Profilrolle über das angegebene `SourceAmiID`.

Schritt 2: runPreUpdate Skript (**`aws:runCommand`**Aktion)

Mit diesem Schritt können Sie ein Skript als Zeichenfolge angeben, das ausgeführt wird, bevor Updates installiert werden.

Schritt 3: updateEC2Config (**`aws:runCommand`**-Aktion)

In diesem Schritt wird das `AWS-InstallPowerShellModule` Runbook verwendet, um ein AWS öffentliches PowerShell Modul herunterzuladen. Systems Manager überprüft die Integrität des Modul mithilfe eines SHA-256-Hash. Systems Manager überprüft dann das Betriebssystem, um zu bestimmen, ob EC2Config oder EC2Launch aktualisiert werden müssen. EC2Config wird unter Windows Server 2008 R2 bis Windows Server 2012 R2 ausgeführt. EC2Launch wird unter Windows Server 2016 ausgeführt.

Schritt 4: updateSSMAgent (**`aws:runCommand`**-Aktion)

Dieser Schritt aktualisiert SSM Agent mithilfe des `AWS-UpdateSSMAgent`-Runbooks.

Schritt 5: Update AWSPVDriver (**`aws:runCommand`**Aktion)

In diesem Schritt werden die AWS PV-Treiber mithilfe des `AWS-ConfigureAWSPackage` Runbooks aktualisiert.

Schritt 6: updateAwsEna NetworkDriver (**`aws:runCommand`**Aktion)

In diesem Schritt werden die AWS ENA-Netzwerktreiber mithilfe des `AWS-ConfigureAWSPackage` Runbooks aktualisiert.

Schritt 7: installWindowsUpdates (**`aws:runCommand`**Aktion)

Dieser Schritt installiert Windows-Updates mithilfe des `AWS-InstallWindowsUpdates`-Runbooks. Standardmäßig sucht und installiert Systems Manager alle fehlenden Updates. Sie können das Standardverhalten ändern, indem Sie einen der folgenden Parameter festlegen: `IncludeKbs`, `ExcludeKbs`, `Categories` oder `SeverityLevels`.

Schritt 8: runPostUpdate Script (**`aws:runCommand`**Aktion)

Mit diesem Schritt können Sie ein Skript als Zeichenfolge angeben, das ausgeführt wird, nachdem Updates installiert wurden.

Schritt 9: runSysprepGeneralize (**`aws:runCommand`**Aktion)

In diesem Schritt wird das `AWS-InstallPowerShellModule` Runbook verwendet, um ein AWS öffentliches PowerShell Modul herunterzuladen. Systems Manager überprüft die Integrität des Modul mithilfe eines SHA-256-Hash. Systems Manager führt dann Sysprep mit AWS unterstützten Methoden für EC2Launch (Windows Server 2016) oder EC2Config (Windows Server 2008 R2 bis 2012 R2) aus.

Schritt 10: stopInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt stoppt die aktualisierte Instance.

Schritt 11: createImage (**`aws:createImage`**-Aktion)

Dieser Schritt erstellt ein neues AMI mit einem aussagekräftigen Namen, der es mit der Quell-ID und dem Zeitpunkt der Erstellung verknüpft. Beispiel: „Von EC2 Automation am {{global:Date_Time}} von {{Id}} generiertes AMI“, wobei DATE_TIME und SourceAmi SourceID Automatisierungsvariablen darstellen.

TerminateInstance **`aws:changeInstanceState`**Schritt 12: (Aktion)

Dieser Schritt bereinigt die Automatisierung durch Beenden der ausgeführten Instance.

Output

In diesem Abschnitt können Sie die Ausgabe verschiedener Schritte oder Werte eines beliebigen Parameters als die Automation-Ausgabe bestimmen. Standardmäßig ist die Ausgabe die ID des aktualisierten Windows-AMI, das von der Automatisierung erstellt wurde.

> **Note**
>
> Standardmäßig verwendet das System die Standard-VPC (172.30.0.0/16), wenn Automation das `AWS-UpdateWindowsAmi`-Runbbok ausführt und eine temporäre Instance erstellt. Wenn Sie die Standard-VPC gelöscht haben, erhalten Sie den folgenden Fehler:
> VPC nicht definiert 400

Zur Behebung dieses Problems erstellen Sie eine Kopie des `AWS-UpdateWindowsAmi`-Runbooks und geben eine Subnetz-ID an. Weitere Informationen finden Sie unter VPC nicht definiert 400.

So erstellen Sie ein gepatchtes Windows-AMI mit der Automation

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um das `AWS-UpdateWindowsAmi`-Runbook zu starten. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen. Der Beispielbefehl unten verwendet ein aktuelles Amazon EC2 AMI zum Minimieren der Anzahl von Patches, die angewendet werden müssen. Wenn Sie diesen Befehl mehrmals ausführen, müssen Sie einen eindeutigen Wert für `targetAMIname` angeben. AMI-Namen müssen einzigartig sein.

   ```
   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'
   ```

   Der Befehl gibt eine Ausführungs-ID zurück. Kopieren Sie diese ID in die Zwischenablage. Sie werden diese ID zum Anzeigen des Status der Automatisierung verwenden.

   ```
   {
       "AutomationExecutionId": "automation execution ID"
   }
   ```

3. Führen Sie den folgenden Befehl aus AWS CLI, um die Automatisierung mit dem anzuzeigen:

   ```
   aws ssm describe-automation-executions
   ```

4. Führen Sie den folgenden Befehl aus, um Details über den Automatisierungsprozess anzuzeigen.

```
aws ssm get-automation-execution
    --automation-execution-id automation execution ID
```

> **Note**
>
> Abhängig von der Anzahl der angewendeten Patches kann der Windows-Patch-Vorgang in dieser Beispielautomatisierung 30 Minuten oder länger in Anspruch nehmen.

Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store

Im folgenden Beispiel wird das Modell verwendet, bei dem eine Organisation ihre eigenen, proprietären AMIs verwaltet und regelmäßig patcht, anstatt aus Amazon Elastic Compute Cloud (Amazon EC2)-AMIs aufzubauen.

Das folgende Verfahren zeigt, wie Betriebssystem-Patches (OS) automatisch auf ein Betriebssystem angewendet werdenAMI, das bereits als die aktuellsten up-to-date oder aktuellsten giltAMI. In diesem Beispiel `SourceAmiId` wird der Standardwert des Parameters durch einen AWS Systems Manager Parameter Store Parameter definiert, der aufgerufen wird`latestAmi`. Der Wert von `latestAmi` wird durch eine AWS Lambda Funktion aktualisiert, die am Ende der Automatisierung aufgerufen wird. Durch diesen Automatisierungsprozess werden der Zeit- und Arbeitsaufwand für das Patchen minimiert, da AMIs das Patchen immer auf die meisten angewendet wird. up-to-date AMI Parameter Storeund Automatisierung sind Fähigkeiten von. AWS Systems Manager

Bevor Sie beginnen

Konfigurieren Sie Automatisierungsrollen und optional Amazon EventBridge for Automation. Weitere Informationen finden Sie unter Einrichten der Automatisierung.

Inhalt

- Aufgabe 1: Erstellen eines Parameters im Systems Manager-Parameter Store
- Aufgabe 2: Erstellen einer IAM-Rolle für AWS Lambda
- Aufgabe 3: Erstellen einer AWS Lambda -Funktion
- Aufgabe 4: Erstellen eines Runbooks und Patchen des AMI

Aufgabe 1: Erstellen eines Parameters im Systems Manager-Parameter Store

Erstellen Sie einen Zeichenfolgen-Parameter in Parameter Store, der die folgenden Informationen verwendet:

- Name: `latestAmi`.
- Value (Wert): Eine AMI-ID. Zum Beispiel: `ami-188d6e0e`.

Informationen zur Erstellung eines Parameter Store-Zeichenfolgenparameters finden Sie unter Erstellen von Systems Manager-Parametern.

Aufgabe 2: Erstellen einer IAM-Rolle für AWS Lambda

Gehen Sie wie folgt vor, um eine IAM-Servicerolle für AWS Lambda zu erstellen. Diese Richtlinien erteilen Lambda die Berechtigung zum Aktualisieren des Werts des `latestAmi`-Parameters mithilfe einer Lambda-Funktion und von Systems Manager.

So erstellen Sie eine IAM-Service-Rolle für Lambda

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create policy (Richtlinie erstellen).
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie den Standardinhalt durch die folgende Richtlinie. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:region:123456789012:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
```

```
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:region:123456789012:log-group:/aws/lambda/function
 name:*"
            ]
        }
    ]
}
```

5. Wählen Sie Weiter: Markierungen.
6. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern.
7. Wählen Sie Weiter: Prüfen aus.
8. Geben Sie auf der Seite Review Policy (Richtlinie prüfen) im Feld Name (Name) einen Namen für die Inline-Richtlinie ein, z. B. **`amiLambda`**.
9. Wählen Sie Richtlinie erstellen aus.
10. Wiederholen Sie die Schritte 2 und 3.
11. Fügen Sie die folgende Richtlinie ein. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:PutParameter",
            "Resource": "arn:aws:ssm:region:123456789012:parameter/latestAmi"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:DescribeParameters",
            "Resource": "*"
        }
    ]
}
```

12. Wählen Sie Weiter: Markierungen.
13. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern.

14. Wählen Sie Weiter: Prüfen aus.
15. Geben Sie auf der Seite Review Policy (Richtlinie prüfen) im Feld Name (Name) einen Namen für die Inline-Richtlinie ein, z. B. **`amiParameter`**.
16. Wählen Sie Richtlinie erstellen aus.
17. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
18. Wählen Sie direkt unter Anwendungsfall die Option Lambda und dann Weiter aus.
19. Suchen Sie auf der Seite Berechtigungsrichtlinien anfügen im Feld Suche die beiden Richtlinien, die Sie zuvor erstellt haben.
20. Aktivieren Sie das Kontrollkästchen neben den Richtlinien und wählen Sie anschließend Weiter aus.
21. Geben Sie unter Role name (Rollenname) einen Namen für Ihre neue Rolle, wie z. B. **`lambda-ssm-role`**, oder einen anderen von Ihnen bevorzugten Namen ein.

    > **Note**
    >
    > Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung nicht geändert werden.

22. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, nachzuverfolgen oder zu steuern, und wählen Sie dann Rolle erstellen aus.

Aufgabe 3: Erstellen einer AWS Lambda -Funktion

Führen Sie die folgenden Schritte zum Erstellen einer Lambda-Funktion aus, die den Wert des `latestAmi`-Parameters automatisch aktualisiert.

Eine Lambda-Funktion erstellen

1. Melden Sie sich bei https://console.aws.amazon.com/lambda/ an AWS Management Console und öffnen Sie die AWS Lambda Konsole.
2. Wählen Sie Funktion erstellen.
3. Wählen Sie auf der Seite Create function die Option Author from scratch.
4. Geben Sie für Function name (Funktionsname) **`Automation-UpdateSsmParam`** ein.
5. Wählen Sie für Runtime (Laufzeit) die Option Python 3.8 aus.

6. Wählen Sie unter Architektur den Computerprozessortyp aus, den Lambda zum Ausführen der Funktion verwenden soll, x86_64 oder arm64,
7. Erweitern Sie im Abschnitt Berechtigungen die Option Standardausführungsrolle ändern.
8. Wählen Sie Use an existing role (Vorhandene Rolle verwenden) aus und wählen Sie dann die Servicerolle für Lambda aus, die Sie in Aufgabe 2 erstellt haben.
9. Wählen Sie Funktion erstellen.
10. Löschen Sie im Bereich Code-Quelle in der Registerkarte lambda_function den vorab ausgefüllten Code im Feld und fügen Sie das folgende Codebeispiel ein.

```
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. Klicken Sie auf Datei, Speichern.
12. Um die Lambda-Funktion zu testen, wählen Sie im Menü Test die Option Testereignis konfigurieren aus.
13. Geben Sie für Event name (Ereignisname) einen Namen für das Testereignis ein, z. B. **`MyTestEvent`**.
14. Ersetzen Sie den vorhandenen Text durch folgendes JSON-Objekt. Ersetzen Sie *`AMI ID`* mit Ihren eigenen Informationen, um Ihren `latestAmi`-Parameterwert festzulegen.

```
{
   "parameterName":"latestAmi",
   "parameterValue":"AMI ID"
}
```

15. Wählen Sie Speichern.
16. Wählen Sie Test aus, um die Funktion zu testen. Auf der Registerkarte Ausführungsergebnis sollte der Status als Erfolgreich gemeldet werden, zusammen mit anderen Details zur Aktualisierung.

Aufgabe 4: Erstellen eines Runbooks und Patchen des AMI

Verwenden Sie die folgende Vorgehensweise zum Erstellen und Ausführen eines Runbooks, das das von Ihnen angegebene AMI für den latestAmi-Parameter patcht. Nach dem Abschluss der Automatisierung wird der Wert latestAmi mit der ID des neu gepatchten AMI aktualisiert. Bei nachfolgenden Automatisierungen verwenden Sie das AMI, das in der vorherigen Ausführung erstellt wurde.

Erstellen und Ausführen des Runbooks

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie für Dokument erstellen die Option Automatisierung aus.
4. Geben Sie unter Name **`UpdateMyLatestWindowsAmi`** ein.
5. Wählen Sie die Registerkarte Editor und wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie bei Aufforderung OK aus.
7. Ersetzen Sie im Feld Dokument-Editor den Standardinhalt durch den folgenden Inhalt des YAML-Beispiel-Runbooks.

```
---
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. Wählen Sie Create automation (Automation erstellen).
9. Wählen Sie im Navigationsbereich Automation (Automatisierung) und Execute automation (Automatisierung ausführen) aus.
10. Wählen Sie auf der Seite Choose document (Dokument wählen), die Registerkarte Owned by me (In meinem Besitz).
11. Suchen Sie nach dem UpdateMyLatestWindowsAmiRunbook und wählen Sie die Schaltfläche auf der UpdateMyLatestWindowsAmiKarte aus.
12. Wählen Sie Weiter aus.
13. Wählen Sie Simple execution (Einfache Ausführung) aus.
14. Geben Sie Werte für die Eingabeparameter an.
15. Wählen Sie Execute (Ausführen).
16. Wählen Sie nach dem Abschluss der Automatisierung Parameter Store im Navigationsbereich und bestätigen Sie, dass der neue Wert für `latestAmi`-Treffer, die von der Automatisierung zurückgegeben werden. Sie können auch überprüfen, ob die neue AMI-ID der Automation-Ausgabe im Abschnitt AMIs der Amazon EC2-Konsole entspricht.

Aktualisierung AMIs mithilfe von Automation und Jenkins

Wenn Ihr Unternehmen Jenkins Software in einer CI/CD-Pipeline verwendet, können Sie Automatisierung als Post-Build-Schritt hinzufügen, um Anwendungsversionen in () vorzuinstallieren. Amazon Machine Images AMIs Automatisierung ist eine Fähigkeit von. AWS Systems Manager Sie können die Jenkins Planungsfunktion auch verwenden, um Automation aufzurufen und Ihre eigene Patch-Frequenz für Ihr Betriebssystem (OS) zu erstellen.

Das folgende Beispiel zeigt, wie Automation von einem Jenkins Server aus aufgerufen wird, der entweder lokal oder in Amazon Elastic Compute Cloud (Amazon EC2) läuft. Für die Authentifizierung verwendet der Jenkins Server AWS Anmeldeinformationen, die auf einer IAM-Richtlinie basieren, die Sie im Beispiel erstellen und an Ihr Instance-Profil anhängen.

> **Note**
>
> Beachten Sie bei der Konfiguration Ihrer Instance unbedingt die bewährten Jenkins Sicherheitsmethoden.

Bevor Sie beginnen

Führen Sie die folgenden Aufgaben aus, bevor Sie Automation mit konfigurierenJenkins:

- Schließen Sie das Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store-Beispiel ab. Im folgenden Beispiel wird das in diesem Beispiel erstellte UpdateMyLatestWindowsAmiRunbook verwendet.
- Konfigurieren Sie IAM-Rollen für Automation. Systems Manager benötigt eine Instance-Profilrolle und einen Servicerollen-ARN zur Verarbeitung von Automatisierungen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.

Um eine IAM-Richtlinie für den Server zu erstellen Jenkins

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create policy (Richtlinie erstellen).
3. Wählen Sie den Tab JSON.

4. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": "ssm:StartAutomationExecution",
                "Resource": [
                    "arn:aws:ssm:region:account ID:document/
UpdateMyLatestWindowsAmi",
                    "arn:aws:ssm:region:account ID:automation-definition/
UpdateMyLatestWindowsAmi:$DEFAULT"
                ]
            }
        ]
    }
```

5. Wählen Sie Richtlinie prüfen.
6. Geben Sie auf der Seite Review Policy (Richtlinie prüfen) im Feld Name (Name) einen Namen für die Inline-Richtlinie ein, z. B. **`JenkinsPolicy`**.
7. Wählen Sie Richtlinie erstellen aus.
8. Wählen Sie im Navigationsbereich Rollen aus.
9. Wählen Sie das Instanzprofil aus, das an Ihren Jenkins Server angehängt ist.
10. Wählen Sie auf der Registerkarte Berechtigungen die Option Berechtigungen hinzufügen, Richtlinien anfügen.
11. Geben Sie im Abschnitt Andere Berechtigungsrichtlinien den Namen der Richtlinie ein, die Sie in den vorherigen Schritten erstellt haben. Zum Beispiel JenkinsPolicy.
12. Aktivieren Sie das Kontrollkästchen neben Ihrer Richtlinie, und wählen Sie Richtlinien anfügen aus.

Verwenden Sie das folgende Verfahren, um das AWS CLI auf Ihrem Jenkins Server zu konfigurieren.

Um den Jenkins Server für die Automatisierung zu konfigurieren

1. Stellen Sie mit Ihrem bevorzugten Browser eine Connect zu Ihrem Jenkins Server über Port 8080 her, um auf die Verwaltungsschnittstelle zuzugreifen.

2. Geben Sie das Passwort ein, welches Sie unter `/var/lib/jenkins/secrets/initialAdminPassword` finden. Um Ihr Kennwort anzuzeigen, führen Sie den folgenden Befehl aus.

   ```
   sudo cat /var/lib/jenkins/secrets/initialAdminPassword
   ```

3. Das Jenkins Installationsskript leitet Sie zur Jenkins Seite Anpassen weiter. Wählen Sie Installieren von empfohlenen Plugins.
4. Sobald die Installation abgeschlossen ist, wählen Sie Administratoranmeldedaten, dann Anmeldeinformationen speichern und anschließend Verwendung starten ausJenkins.
5. Wählen Sie im linken Navigationsbereich Verwalten Jenkins und dann Plugins verwalten aus.
6. Wählen Sie die Registerkarte Available (Verfügbar) und geben Sie dann **`Amazon EC2 plugin`** ein.
7. Aktivieren Sie das Kontrollkästchen für **`Amazon EC2 plugin`** und klicken Sie dann auf Installation ohne Neustart.
8. Nach abgeschlossener Installation wählen Sie Zurück zur oberen Seite.
9. Wählen Sie Verwalten Jenkins und anschließend Knoten und Clouds verwalten aus.
10. Wählen Sie im Abschnitt Clouds konfigurieren die Option Neue Cloud hinzufügen und dann Amazon EC2 aus.
11. Geben Sie Ihre Daten in die verbleibenden Felder ein. Stellen Sie sicher, dass Sie die Option EC2-Instance-Profil zum Abrufen von Anmeldeinformationen verwenden, ausgewählt haben.

Gehen Sie wie folgt vor, um Ihr Jenkins Projekt so zu konfigurieren, dass Automation aufgerufen wird.

So konfigurieren Sie Ihren Jenkins Server für den Aufruf von Automation

1. Öffnen Sie die Jenkins Konsole in einem Webbrowser.
2. Wählen Sie das Projekt, das Sie mit Automation konfigurieren möchten, und wählen Sie dann Configure (Konfigurieren).
3. Wählen Sie auf der Registerkarte Build Add Build Step (Build-Schritt hinzufügen) aus.
4. Wählen Sie je nach Betriebssystem Execute shell (Shell ausführen) oder Execute Windows batch command (Windows-Batchbefehl ausführen).
5. Führen Sie im Feld Befehl einen AWS CLI Befehl wie den folgenden aus. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

```
aws ssm start-automation-execution \
        --document-name runbook name \
        --region AWS-Region of your source AMI \
        --parameters runbook parameters
```

Der folgende Beispielbefehl verwendet das UpdateMyLatestWindowsAmiRunbook und den Systems Manager Manager-Parameter, die in `latestAmi` Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store erstellt wurden.

```
aws ssm start-automation-execution \
        --document-name UpdateMyLatestWindowsAmi \
        --parameters \
            "sourceAMIid='{{ssm:latestAmi}}'"
        --region region
```

JenkinsIn sieht der Befehl wie das Beispiel im folgenden Screenshot aus.

6. Wählen Sie im Jenkins Projekt Build Now aus. Jenkinsgibt eine Ausgabe zurück, die dem folgenden Beispiel ähnelt.

Aktualisieren von AMIs für Auto-Scaling-Gruppen

Im folgenden Beispiel wird eine Auto-Scaling-Gruppe mit einem neu gepatchten AMI aktualisiert. Dieser Ansatz gewährleistet, dass neue Images automatisch den verschiedenen Computing-Umgebungen zur Verfügung gestellt werden, die Auto–Scaling-Gruppen verwenden.

Der letzte Schritt der Automatisierung in diesem Beispiel verwendet eine Python-Funktion, um eine neue Startvorlage zu erstellen, die das neu gepatchte AMI verwendet. Anschließend wird die Auto-Scaling-Gruppe aktualisiert, um die neue Startvorlage zu verwenden. In diesem Auto–Scaling-Szenariotyp können Benutzer vorhandene Instances in der Auto–Scaling-Gruppe beenden, um den Start einer neuen Instance zu erzwingen, die das neue Image verwendet. Andernfalls konnten Benutzer warten und das Skalieren der Ereignisse nach oben oder unten zulassen, um auf natürliche Weise neuere Instances zu starten.

Bevor Sie beginnen

Bevor Sie mit diesem Beispiel beginnen, führen Sie die folgenden Aufgaben aus.

- Konfigurieren Sie IAM-Rollen für Automation, eine Funktion von AWS Systems Manager. Systems Manager benötigt eine Instance-Profilrolle und einen Servicerollen-ARN zur Verarbeitung von Automatisierungen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.

Erstellen Sie das PatchAMI ASG-Runbook AndUpdate

Gehen Sie wie folgt vor, um das PatchAMI AndUpdate ASG-Runbook zu erstellen, das die von Ihnen für den SourceAMI-Parameter angegebenen Patches durchführt. AMI Das Runbook aktualisiert auch eine Auto-Scaling-Gruppe, um das neueste, gepatchte AMI zu verwenden.

Erstellen und Ausführen des Runbooks

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/systems-manager/. AWS Systems Manager
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie Automation im Dropdown-Menü Erstellen eines Dokuments.
4. Geben Sie im Feld Name **`PatchAMIAndUpdateASG`** ein.
5. Wählen Sie die Registerkarte Editor und wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie OK aus, wenn Sie dazu aufgefordert werden, und löschen Sie den Inhalt im Feld Document editor (Dokumenteditor).

7. Fügen Sie im Feld Document editor (Dokumenteditor) den folgenden Inhalt des YAML-Beispiel-Runbooks ein.

```
---
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. Wählen Sie Create automation (Automation erstellen).
9. Wählen Sie im Navigationsbereich Automation (Automatisierung) und Execute automation (Automatisierung ausführen) aus.
10. Wählen Sie auf der Seite Choose document (Dokument wählen), die Registerkarte Owned by me (In meinem Besitz).
11. Suchen Sie nach dem PatchAMI AndUpdate ASG-Runbook und wählen Sie die Schaltfläche auf der PatchAmi ASG-Karte aus. AndUpdate
12. Wählen Sie Weiter aus.
13. Wählen Sie Simple execution (Einfache Ausführung) aus.
14. Geben Sie Werte für die Eingabeparameter an. Stellen Sie sicher, dass die von Ihnen angegebenen `SubnetId` und `SecurityGroupIds` den Zugriff auf die öffentlichen Systems-Manager-Endpunkte oder Ihre Schnittstellenendpunkte für Systems Manager zulassen.
15. Wählen Sie Execute (Ausführen).
16. Wählen Sie nach Abschluss der Automatisierung in der Amazon-EC2-Konsole Auto Scaling und dann Launch Templates (Startvorlagen) aus. Stellen Sie sicher, dass die neue Startvorlage angezeigt wird und dass sie das neue AMI verwendet.
17. Klicken Sie auf Auto Scaling und wählen Sie dann Auto-Scaling-Gruppen. Stellen Sie sicher, dass die Auto-Scaling-Gruppe die neue Startkonfiguration verwendet.
18. Beenden Sie mindestens eine Instance in Ihrer Auto–Scaling-Gruppe. Ersatz-Instances werden unter Verwendung der neuen AMI gestartet.

Verwendung von AWS Support-Self-Service-Runbooks

In diesem Abschnitt wird beschrieben, wie Sie einige der Self-Service-Automatisierungen verwenden, die vom AWS Support-Team erstellt wurden. Diese Automatisierungen unterstützen Sie bei der Verwaltung Ihrer AWS-Ressourcen.

Support Automation Workflows

Support Automation Workflows (SAW) sind Automatisierungs-Runbooks, die vomAWS Support-Team erstellt wurden und verwaltet werden. Diese Runbooks helfen Ihnen bei der Behebung häufiger

Probleme mit Ihren AWS-Ressourcen, überwachen proaktiv und identifizieren Netzwerkprobleme, erfassen und analysieren Protokolle und vieles mehr.

SAW-Runbooks verwenden das **`AWSSupport`**-Präfix. Zum Beispiel `AWSSupport-ActivateWindowsWithAmazonLicense`.

Außerdem haben AWS-Enterprise- und Business Support-Kunden auch Zugriff auf Runbooks, welche das **`AWSPremiumSupport`**-Präfix verwenden. Zum Beispiel `AWSPremiumSupport-TroubleshootEC2DiskUsage`.

Weitere Informationen zu AWS Support finden Sie unter Erste Schritte mit AWS Support.

Themen

- Ausführen des EC2Rescue-Tools auf nicht erreichbaren Instances
- Zurücksetzen von Passwörtern und SSH-Schlüsseln auf EC2-Instances

Ausführen des EC2Rescue-Tools auf nicht erreichbaren Instances

EC2Rescue kann Ihnen bei der Diagnose und Behebung von Problemen auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances für Linux und Windows Server helfen. Sie können das Tool manuell ausführen, wie unter Verwenden von EC2Rescue für Linux Server und Verwenden von EC2Rescue für Windows Server beschrieben. Sie können das Tool auch automatisch mit der Systems Manager Automation und dem **`AWSSupport-ExecuteEC2Rescue`**-Runbook ausführen. Automatisierung ist eine Fähigkeit von AWS Systems Manager. Das **`AWSSupport-ExecuteEC2Rescue`**-Runbook ist darauf ausgelegt, eine Kombination aus Systems Manager-Aktionen, AWS CloudFormation -Aktionen und Lambda-Funktionen auszuführen, die die normalerweise erforderlichen Schritte zur Verwendung von EC2Rescue automatisieren.

Sie können das **`AWSSupport-ExecuteEC2Rescue`**-Runbook verwenden, um verschiedene Arten von Problemen bei Betriebssystemen (OS) zu behandeln und möglicherweise zu lösen. Instances mit verschlüsselten Root-Volumes werden nicht unterstützt. Eine vollständige Liste finden Sie in den folgenden Themen:

Windows: Weitere Informationen finden Sie unter Rescue-Aktion in Verwenden von EC2Rescue für Windows Server mit der Befehlszeile.

Linux und macOS: Einige EC2Rescue für Linux-Module erkennen Probleme und versuchen, diese zu lösen. Weitere Informationen finden Sie in der `aws-ec2rescue-linux`Dokumentation zu den einzelnen Modulen unterGitHub.

Funktionsweise

Die Fehlerbehebung bei einer Instance mit Automation und dem **`AWSSupport-ExecuteEC2Rescue`**-Runbook funktioniert folgendermaßen:

- Sie geben die ID der nicht erreichbaren Instance an und starten das Runbook.
- Das System erstellt eine temporäre VPC und führt dann eine Reihe von Lambda-Funktionen aus, um die VPC zu konfigurieren.
- Das System identifiziert ein Subnetz für Ihre temporäre VPC in derselben Availability Zone wie die ursprüngliche Instance.
- Das System startet eine temporäre, SSM-fähige Helferobjekt-Instance.
- Das System stoppt Ihre ursprüngliche Instance und erstellt einen Backup. Anschließend fügt es das ursprüngliche Stamm-Volume an die Helferobjekt-Instance an.
- Das System verwendet Run Command zur Ausführung von EC2Rescue auf der Helferobjekt-Instance. EC2Rescue erkennt Probleme auf dem angehängten, ursprünglichen Stamm-Volume und versucht, sie zu beheben. Nach der Fertigstellung hängt EC2Rescue das Stamm-Volume wieder an die ursprüngliche Instance an.
- Das System startet die ursprüngliche Instance neu und beendet die temporäre Instance. Das System beendet ebenso die temporäre VPC und die Lambda-Funktionen, die zu Beginn der Automatisierung erstellt wurden.

Bevor Sie beginnen

Bevor Sie die folgende Automation ausführen, führen Sie die folgenden Schritte aus:

- Kopieren Sie die Instance-ID der nicht erreichbaren Instance. Sie legen diese ID im Verfahren fest.
- Erfassen Sie optional die ID eines Subnetzes in derselben Availability Zone wie Ihre unerreichbare Instance. Die EC2Rescue Instance wird in diesem Subnetz erstellt. Wenn Sie kein Subnetz angeben, erstellt Automation eine neue temporäre VPC in Ihrem. AWS-Konto Stellen Sie sicher, AWS-Konto dass mindestens eine VPC verfügbar ist. Sie können standardmäßig fünf VPCs in einer Region erstellen. Wenn Sie bereits fünf VPCs in der Region erstellt haben, schlägt die Automatisierung fehl, ohne dass Änderungen an Ihrer Instance vorgenommen werden. Weitere Informationen zu Amazon VPC-Kontingenten finden Sie unter VPC und Subnetze im Amazon VPC-Benutzerhandbuch.

- Optional können Sie eine AWS Identity and Access Management (IAM-) Rolle für die Automatisierung erstellen und angeben. Falls Sie diese Rolle nicht festlegen, wird die Automatisierung im Kontext des Benutzers ausgeführt, der die Automatisierung ausgeführt hat.

Gewähren von **`AWSSupport-EC2Rescue`**-Berechtigungen zum Durchführen von Aktionen auf Ihren Instances

EC2Rescue benötigt die Berechtigung, um während der Automatisierung eine Reihe von Aktionen auf Ihren Instances durchzuführen. Diese Aktionen rufen die AWS Lambda Dienste IAM und Amazon EC2 auf, um sicher und geschützt zu versuchen, Probleme mit Ihren Instances zu beheben. Wenn Sie in Ihrer AWS-Konto und/oder Ihrer VPC über Administratorberechtigungen verfügen, können Sie die Automatisierung möglicherweise ausführen, ohne Berechtigungen zu konfigurieren, wie in diesem Abschnitt beschrieben. Falls Sie keine Administratorberechtigungen besitzen, müssen Sie oder ein Administrator Berechtigungen anhand einer der folgenden Optionen konfigurieren.

- Erteilen von Berechtigungen mithilfe von IAM-Richtlinien
- Erteilen von Berechtigungen mithilfe einer Vorlage AWS CloudFormation

Erteilen von Berechtigungen mithilfe von IAM-Richtlinien

Sie können entweder die folgende IAM-Richtlinie als eingebundene Richtlinie an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen. Sie können aber auch eine neue verwaltete IAM-Richtlinie erstellen und diese an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen. Weitere Informationen zum Hinzufügen einer eingebundenen Richtlinie zu Ihrem Benutzerkonto, Ihrer Gruppe oder Ihrer Rolle finden Sie unter Verwenden von eingebundenen Richtlinien. Weitere Informationen zum Erstellen einer neuen verwalteten Richtlinien finden Sie unter Verwenden von eingebundenen Richtlinien.

> **Note**
>
> Wenn Sie eine neue IAM-verwaltete Richtlinie erstellen, müssen Sie ihr auch die AutomationRole verwaltete AmazonSSM-Richtlinie hinzufügen, damit Ihre Instances mit der Systems Manager Manager-API kommunizieren können.

IAM-Richtlinie für -EC2Rescue AWSSupport

Ersetzen Sie *`account ID`* (Konto-ID) mit Ihren eigenen Informationen.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Action": [
            "lambda:InvokeFunction",
            "lambda:DeleteFunction",
            "lambda:GetFunction"
         ],
         "Resource": "arn:aws:lambda:*:account ID: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::account ID:role/AWSSupport-EC2Rescue-*",
            "arn:aws:iam::account ID: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"
        }
    ]
}
```

Erteilen von Berechtigungen mithilfe einer Vorlage AWS CloudFormation

AWS CloudFormation automatisiert den Prozess der Erstellung von IAM-Rollen und -Richtlinien mithilfe einer vorkonfigurierten Vorlage. Erstellen Sie mit den folgenden Schritten die erforderlichen IAM-Rollen und IAM-Richtlinien für die EC2Rescue-Automatisierung mithilfe von AWS CloudFormation.

So erstellen Sie die erforderlichen IAM-Rollen und IAM-Richtlinien für EC2Rescue

1. Laden Sie AWSSupport-EC2RescueRole.zip herunter und extrahieren Sie die `AWSSupport-EC2RescueRole.json`-Datei in ein Verzeichnis auf Ihrem lokalen Computer.

2. Wenn Sie AWS-Konto sich in einer speziellen Partition befinden, bearbeiten Sie die Vorlage, um die ARN-Werte in die für Ihre Partition zu ändern.

   Ändern Sie beispielsweise für `arn:aws` alle Fälle von in `arn:aws-cn`.
3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
4. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
5. Wählen Sie auf der Seite Create stack (Stack erstellen) unter Prerequisite - Prepare template (Voraussetzung – Vorlage vorbereiten) die Option Template is ready (Vorlage ist bereit) aus.
6. Wählen Sie unter Vorlage angeben die Option Vorlagendatei hochladen aus.
7. Wählen Sie Choose file (Datei auswählen) aus, navigieren Sie dann zu der `AWSSupport-EC2RescueRole.json`-Datei aus dem Verzeichnis, in dem Sie sie extrahiert haben, und wählen Sie sie aus.
8. Wählen Sie Weiter aus.
9. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) für das Feld Stack name (Stack-Name) einen Namen ein, um diesen Stack zu identifizieren. Wählen Sie dann Next (Weiter) aus.
10. (Optional) Wenden Sie im Bereich Tags ein oder mehrere Tag-Schlüsselname/-wertpaare auf den Stack an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Beispielsweise können Sie einen Stack kennzeichnen, um den Typ der ausgeführten Aufgaben, die Typen von Zielen oder anderen Ressourcen und die Umgebung zu identifizieren, in der er ausgeführt wird.
11. Wählen Sie Next (Weiter)
12. Überprüfen Sie auf der Seite „Überprüfen“ die Stack-Details, scrollen Sie dann nach unten und wählen Sie die Option Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen erstellt werden.
13. Wählen Sie Stack erstellen aus.

    AWS CloudFormation zeigt für einige Minuten den Status CREATE_IN_PROGRESS an. Nach dem Erstellen des Stacks ändert sich der Status in CREATE_COMPLETE. Sie können auch auf das Aktualisierungssymbol klicken, um den Status des Erstellungsprozesses zu überprüfen.

14. Wählen Sie in der Stacks-Liste die Option neben den Stack, den Sie gerade erstellt haben, und wählen Sie dann die Registerkarte Outputs (Ausgaben).
15. Notieren Sie sich den Wert. Das ist der ARN von AssumeRole. Sie geben diesen ARN an, wenn Sie die Automatisierung in der nächsten Prozedur ausführen, Ausführen der Automation.

Ausführen der Automation

> **Important**
>
> Der folgende Automatisierung hält die nicht erreichbare Instance an. Das Anhalten der Instance kann zu Datenverlusten auf den angehängten Instance-Speicher-Volumes (sofern vorhanden) führen. Das Anhalten der Instance kann auch dazu führen, dass die öffentliche IP-Adresse geändert wird, wenn keine elastische IP-Adresse zugeordnet ist.

Führen Sie die **`AWSSupport-ExecuteEC2Rescue`**-Automation aus.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie Execute automation (Automatisierung ausführen).
4. Wählen Sie im Abschnitt Automation document (Automatisierungsdokument) die Option Owned by Amazon (Im Besitz von Amazon) aus der Liste aus.
5. Wählen Sie in der Runbooks-Liste die Schaltfläche auf der Karte für **`AWSSupport-ExecuteEC2Rescue`** und wählen Sie danach Weiter.
6. Klicken Sie auf der Seite Execute automation document (Automation-Dokument ausführen) auf Simple execution (Einfache Ausführung).
7. Überprüfen Sie im Abschnitt Document details (Dokumentdetails), ob Document version (Dokumentversion) auf die höchste Standardversion gesetzt ist. Beispiel: $DEFAULT oder 3 (default) (3 (Standard)).
8. Geben Sie im Abschnitt Input Parameters die folgenden Parameter an.
    a. Geben Sie für UnreachableInstanceIddie ID der nicht erreichbaren Instanz an.
    b. (Optional) Geben Sie für EC2 RescueInstanceType einen Instance-Typ für die EC2Rescue-Instance an. Der Standard-Instance-Typ lautet `t2.medium`.

c. Denn AutomationAssumeRolewenn Sie Rollen für diese Automatisierung mithilfe des weiter oben in diesem Thema beschriebenen AWS CloudFormation Verfahrens erstellt haben, wählen Sie den ARN aus AssumeRole , den Sie in der AWS CloudFormation Konsole erstellt haben.

d. (Optional) Geben Sie für einen S3-Bucket an LogDestination, wenn Sie bei der Fehlerbehebung für Ihre Instance Protokolle auf Betriebssystemebene sammeln möchten. Protokolle werden automatisch in den angegebenen Bucket hochgeladen.

e. Geben Sie für SubnetIdein Subnetz in einer vorhandenen VPC in derselben Availability Zone wie die nicht erreichbare Instance an. Standardmäßig erstellt Systems Manager eine neue VPC, aber Sie können ein Subnetz in einer vorhandenen VPC angeben, wenn Sie möchten.

> **Note**
>
> Wenn Sie die Option zum Erstellen eines Buckets oder einer Subnetz-ID nicht sehen, überprüfen Sie, ob Sie die neueste Default-Version des Runbooks verwenden.

9. (Optional) Wenden Sie im Bereich Tags mindestens ein Tag-Schlüsselname-/-Wert-Paar an, um die Automatisierung zu identifizieren, z. B. `Key=Purpose,Value=EC2Rescue`.
10. Wählen Sie Execute (Ausführen).

Das Runbook erstellt ein Backup AMI als Teil der Automatisierung. Alle anderen von der Automatisierung erstellten Ressourcen werden automatisch gelöscht, aber dieses AMI verbleibt in Ihrem Konto. Der AMI-Name wird unter Verwendung der folgenden Konvention generiert:

Backup-AMI: AWSSupport -EC2Rescue: *`UnreachableInstanceId`*

Sie finden dieses AMI in der Amazon EC2-Konsole, indem Sie nach der Automation-Ausführungs-ID suchen.

Zurücksetzen von Passwörtern und SSH-Schlüsseln auf EC2-Instances

Sie können das `AWSSupport-ResetAccess`-Runbook verwenden, um die Generierung des lokalen Administratorkennworts auf Amazon Elastic Compute Cloud Amazon-EC2-Instances für Windows Server automatisch wieder zu aktivieren und einen neuen SSH-Schlüssel auf EC2-Instances für Linux zu generieren. Das `AWSSupport-ResetAccess` Runbook ist so konzipiert, dass es eine Kombination von AWS Systems Manager Aktionen, AWS CloudFormation Aktionen und AWS

Lambda Funktionen ausführt, die die Schritte automatisieren, die normalerweise zum Zurücksetzen des lokalen Administratorkennworts erforderlich sind.

Mithilfe von Automation, einer Funktion von, können Sie das `AWSSupport-ResetAccess` Runbook verwenden AWS Systems Manager, um die folgenden Probleme zu lösen:

Windows

Sie haben das EC2-Schlüsselpaar verloren: Um dieses Problem zu lösen, können Sie das AWSSupportResetAccess-Runbook verwenden, um eine kennwortfähige Instance AMI aus Ihrer aktuellen Instance zu erstellen, eine neue Instance über das AMI zu starten und ein key pair auszuwählen, das Ihnen gehört.

Sie haben das lokale Administratorpasswort verloren: Um dieses Problem zu beheben, können Sie das `AWSSupport-ResetAccess`-Runbook verwenden, um ein neues Passwort zu generieren, das Sie mit dem aktuellen EC2-Schlüsselpaar entschlüsseln können.

Linux

Sie haben Ihr EC2-Schlüsselpaar verloren oder Sie haben den SSH-Zugriff auf die Instance mit einem Schlüssel konfiguriert, den Sie verloren haben: Um dieses Problem zu beheben, können Sie das `AWSSupport-ResetAccess`-Runbook verwenden, um einen neuen SSH-Schlüssel für Ihre aktuelle Instance zu erstellen, mit dem Sie erneut eine Verbindung mit der Instance herstellen können.

> **Note**
>
> Wenn Ihre EC2-Instance für Windows Server für Systems Manager konfiguriert ist, können Sie auch Ihr lokales Administratorpasswort mit EC2Rescue und AWS Systems Manager Run Command zurücksetzen. Weitere Informationen finden Sie unter Verwenden von EC2Rescue für Windows Server mit Systems Manager Run Command im Amazon EC2 EC2-Benutzerhandbuch.

Ähnliche Informationen

Stellen Sie mithilfe von PuTTY im Amazon EC2 EC2-Benutzerhandbuch von Windows aus eine Connect zu Ihrer Linux-Instance her

Funktionsweise

Die Fehlerbehebung bei einer Instance mit Automation und dem `AWSSupport-ResetAccess`-Runbook funktioniert folgendermaßen:

- Sie geben die ID der Instance an und führen das Runbook aus.
- Das System erstellt eine temporäre VPC und führt dann eine Reihe von Lambda-Funktionen aus, um die VPC zu konfigurieren.
- Das System identifiziert ein Subnetz für Ihre temporäre VPC in derselben Availability Zone wie die ursprüngliche Instance.
- Das System startet eine temporäre, SSM-fähige Helferobjekt-Instance.
- Das System stoppt Ihre ursprüngliche Instance und erstellt einen Backup. Anschließend fügt es das ursprüngliche Stamm-Volume an die Helferobjekt-Instance an.
- Das System verwendet Run Command zur Ausführung von EC2Rescue auf der Helferobjekt-Instance. Unter Windows ermöglicht EC2Rescue die Passwortgenerierung für den lokalen Administrator unter Verwendung von EC2Config oder EC2Launch auf dem zugeordneten, ursprünglichen Stamm-Volume. Unter Linux generiert und fügt EC2Rescue einen neuen SSH-Schlüssel ein und speichert den privaten Schlüssel verschlüsselt in Parameter Store. Nach der Fertigstellung hängt EC2Rescue das Stamm-Volume wieder an die ursprüngliche Instance an.
- Das System erstellt eine neue Amazon Machine Image (AMI) Ihrer Instance, nachdem die Kennwortgenerierung aktiviert wurde. Mit diesem AMI können Sie gegebenenfalls eine neue EC2-Instance erstellen und ein neues Schlüsselpaar zuordnen.
- Das System startet die ursprüngliche Instance neu und beendet die temporäre Instance. Das System beendet ebenso die temporäre VPC und die Lambda-Funktionen, die zu Beginn der Automatisierung erstellt wurden.
- Windows: Ihre Instance generiert ein neues Kennwort, das Sie unter Verwendung des aktuellen Schlüsselpaars, das der Instance zugewiesen ist, über die Amazon EC2-Konsole decodieren können.

  Linux: Sie können sich per SSH mit der Instance unter Verwendung des SSH-Schlüssels verbinden, der im Systems-Manager-Parameter-Store als /ec2rl/openssh/***instance ID***/key gespeichert ist.

Bevor Sie beginnen

Bevor Sie die folgende Automation ausführen, führen Sie die folgenden Schritte aus:

- Kopieren Sie die Instance-ID der Instance, auf der Sie das Administratorpasswort zurücksetzen möchten. Sie legen diese ID im Verfahren fest.
- Erfassen Sie optional die ID eines Subnetzes in derselben Availability Zone wie Ihre unerreichbare Instance. Die EC2Rescue Instance wird in diesem Subnetz erstellt. Wenn Sie kein Subnetz angeben, erstellt Automation eine neue temporäre VPC in Ihrem. AWS-Konto Stellen Sie sicher, AWS-Konto dass mindestens eine VPC verfügbar ist. Sie können standardmäßig fünf VPCs in einer Region erstellen. Wenn Sie bereits fünf VPCs in der Region erstellt haben, schlägt die Automatisierung fehl, ohne dass Änderungen an Ihrer Instance vorgenommen werden. Weitere Informationen zu Amazon VPC-Kontingenten finden Sie unter VPC und Subnetze im Amazon VPC-Benutzerhandbuch.
- Optional können Sie eine AWS Identity and Access Management (IAM-) Rolle für die Automatisierung erstellen und angeben. Falls Sie diese Rolle nicht festlegen, wird die Automatisierung im Kontext des Benutzers ausgeführt, der die Automatisierung ausgeführt hat.

Erteilen von AWSSupport -EC2Rescue-Berechtigungen zur Durchführung von Aktionen auf Ihren Instances

EC2Rescue benötigt die Berechtigung, um während der Automatisierung eine Reihe von Aktionen auf Ihren Instances durchzuführen. Diese Aktionen rufen die AWS Lambda Dienste IAM und Amazon EC2 auf, um sicher und geschützt zu versuchen, Probleme mit Ihren Instances zu beheben. Wenn Sie in Ihrer AWS-Konto und/oder Ihrer VPC über Administratorberechtigungen verfügen, können Sie die Automatisierung möglicherweise ausführen, ohne Berechtigungen zu konfigurieren, wie in diesem Abschnitt beschrieben. Falls Sie keine Administratorberechtigungen besitzen, müssen Sie oder ein Administrator Berechtigungen anhand einer der folgenden Optionen konfigurieren.

- Erteilen von Berechtigungen mithilfe von IAM-Richtlinien
- Erteilen von Berechtigungen mithilfe einer Vorlage AWS CloudFormation

Erteilen von Berechtigungen mithilfe von IAM-Richtlinien

Sie können entweder die folgende IAM-Richtlinie als eingebundene Richtlinie an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen. Sie können aber auch eine neue verwaltete IAM-Richtlinie erstellen und diese an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen. Weitere Informationen zum Hinzufügen einer eingebundenen Richtlinie zu Ihrem Benutzerkonto, Ihrer Gruppe oder Ihrer Rolle finden Sie unter Verwenden von eingebundenen Richtlinien. Weitere Informationen zum Erstellen einer neuen verwalteten Richtlinien finden Sie unter Verwenden von eingebundenen Richtlinien.

> **Note**
>
> Wenn Sie eine neue IAM-verwaltete Richtlinie erstellen, müssen Sie ihr auch die AutomationRole verwaltete AmazonSSM-Richtlinie hinzufügen, damit Ihre Instances mit der Systems Manager Manager-API kommunizieren können.

IAM-Richtlinie für **`AWSSupport-ResetAccess`**

Ersetzen Sie *`account ID`* (Konto-ID) mit Ihren eigenen Informationen.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Action": [
            "lambda:InvokeFunction",
            "lambda:DeleteFunction",
            "lambda:GetFunction"
         ],
         "Resource": "arn:aws:lambda:*:account ID: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::account ID:role/AWSSupport-EC2Rescue-*",
            "arn:aws:iam::account ID: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"
      }
   ]
}
```

Erteilen von Berechtigungen mithilfe einer Vorlage AWS CloudFormation

AWS CloudFormation automatisiert den Prozess der Erstellung von IAM-Rollen und -Richtlinien mithilfe einer vorkonfigurierten Vorlage. Erstellen Sie mit den folgenden Schritten die erforderlichen IAM-Rollen und IAM-Richtlinien für die EC2Rescue-Automatisierung mithilfe von AWS CloudFormation.

So erstellen Sie die erforderlichen IAM-Rollen und IAM-Richtlinien für EC2Rescue

1. Laden Sie `AWSSupport-EC2RescueRole.zip` herunter und extrahieren Sie die `AWSSupport-EC2RescueRole.json`-Datei in ein Verzeichnis auf Ihrem lokalen Computer.
2. Wenn Sie AWS-Konto sich in einer speziellen Partition befinden, bearbeiten Sie die Vorlage, um die ARN-Werte in die für Ihre Partition zu ändern.

   Ändern Sie beispielsweise für `arn:aws` alle Fälle von in `arn:aws-cn`.
3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
4. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
5. Wählen Sie auf der Seite Create stack (Stack erstellen) unter Prerequisite - Prepare template (Voraussetzung – Vorlage vorbereiten) die Option Template is ready (Vorlage ist bereit) aus.
6. Wählen Sie unter Vorlage angeben die Option Vorlagendatei hochladen aus.
7. Wählen Sie Choose file (Datei auswählen) aus, navigieren Sie dann zu der `AWSSupport-EC2RescueRole.json`-Datei aus dem Verzeichnis, in dem Sie sie extrahiert haben, und wählen Sie sie aus.
8. Wählen Sie Weiter aus.
9. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) für das Feld Stack name (Stack-Name) einen Namen ein, um diesen Stack zu identifizieren. Wählen Sie dann Next (Weiter) aus.
10. (Optional) Wenden Sie im Bereich Tags ein oder mehrere Tag-Schlüsselname/-wertpaare auf den Stack an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Beispielsweise können Sie einen Stack kennzeichnen, um den Typ der ausgeführten

Aufgaben, die Typen von Zielen oder anderen Ressourcen und die Umgebung zu identifizieren, in der er ausgeführt wird.

11. Wählen Sie Next (Weiter)
12. Überprüfen Sie auf der Seite Überprüfen die Stack-Details, scrollen Sie dann nach unten und wählen Sie die Option Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen erstellt werden.
13. AWS CloudFormation zeigt für einige Minuten den Status CREATE_IN_PROGRESS an. Nach dem Erstellen des Stacks ändert sich der Status in CREATE_COMPLETE. Sie können auch auf das Aktualisierungssymbol klicken, um den Status des Erstellungsprozesses zu überprüfen.
14. Wählen Sie in der Stackliste die Option neben dem Stack, den Sie gerade erstellt haben, und wählen Sie dann die Registerkarte Outputs (Ausgaben) aus.
15. Kopieren Sie den Value (Wert). Das ist der ARN von AssumeRole. Sie geben diesen ARN bei der Ausführung der Automation an.

Ausführen der Automation

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der AWS Systems Manager -Konsole das `AWSSupport-ResetAccess`-Runbook ausführen.

> **Important**
>
> Die folgende Automatisierung hält die Instance an. Das Anhalten der Instance kann zu Datenverlusten auf den angehängten Instance-Speicher-Volumes (sofern vorhanden) führen. Das Anhalten der Instance kann auch dazu führen, dass die öffentliche IP-Adresse geändert wird, wenn keine elastische IP-Adresse zugeordnet ist. Um diese Konfigurationsänderungen zu vermeiden, verwenden Sie Run Command, um den Zugriff zurückzusetzen. Weitere Informationen finden Sie unter Verwenden von EC2Rescue für Windows Server mit Systems Manager Run Command im Amazon EC2 EC2-Benutzerhandbuch.

Um die - Automatisierung auszuführen AWSSupport ResetAccess

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie Execute automation (Automatisierung ausführen).

4. Wählen Sie im Abschnitt Automation document (Automatisierungsdokument) die Option Owned by Amazon (Im Besitz von Amazon) aus der Liste aus.
5. Wählen Sie in der Runbooks-Liste die Schaltfläche auf der Karte für AWSSupport- ResetAccess und klicken Sie dann auf Weiter.
6. Klicken Sie auf der Seite Execute automation document (Automation-Dokument ausführen) auf Simple execution (Einfache Ausführung).
7. Überprüfen Sie im Abschnitt Document details (Dokumentdetails), ob Document version (Dokumentversion) auf die höchste Standardversion gesetzt ist. Beispiel: $DEFAULT oder 3 (default) (3 (Standard)).
8. Geben Sie im Abschnitt Input Parameters die folgenden Parameter an.

   a. Geben Sie für InstanceID die ID der nicht erreichbaren Instance an.

   b. Geben Sie für SubnetIdein Subnetz in einer vorhandenen VPC in derselben Availability Zone wie die angegebene Instance an. Standardmäßig erstellt Systems Manager eine neue VPC, aber Sie können ein Subnetz in einer vorhandenen VPC angeben, wenn Sie möchten.

   > **Note**
   >
   > Wenn Sie die Option zur Angabe einer Subnetz-ID nicht sehen, überprüfen Sie, ob Sie die neueste Default-Version des Runbooks verwenden.

   c. Geben Sie für EC2 RescueInstance Type einen Instance-Typ für die EC2Rescue-Instance an. Der Standard-Instance-Typ lautet `t2.medium`.

   d. Denn AssumeRolewenn Sie Rollen für diese Automatisierung mithilfe des weiter oben in diesem Thema beschriebenen AWS CloudFormation Verfahrens erstellt haben, geben Sie den AssumeRole ARN an, den Sie in der AWS CloudFormation Konsole notiert haben.
9. (Optional) Wenden Sie im Bereich Tags mindestens ein Tag-Schlüsselname-/-Wert-Paar an, um die Automatisierung zu identifizieren, z. B. `Key=Purpose,Value=ResetAccess`.
10. Wählen Sie Execute (Ausführen).
11. Zur Überwachung des Fortschritts der Automatisierung wählen Sie die laufende Automatisierung und dann die Registerkarte Steps (Schritte). Wenn die Automatisierung abgeschlossen ist, wählen Sie die Registerkarte Descriptions (Beschreibungen) und dann View output (Ausgabe anzeigen), um die Ergebnisse anzuzeigen. Zum Anzeigen der Ausgabe der einzelnen Schritte wählen Sie die Registerkarte Steps (Schritte) und dann neben einem Schritt View Outputs (Ausgabe anzeigen) aus.

Das Runbook erstellt ein Backup AMI und ein kennwortaktiviertes AMI als Teil der Automatisierung. Alle anderen von der Automatisierung erstellten Ressourcen werden automatisch gelöscht, aber diese AMIs verbleiben in Ihrem Konto. Die AMIs-Namen werden unter Verwendung der folgenden Konvention generiert:

- Backup AMI: `AWSSupport-EC2Rescue:`*`InstanceID`*
- *`Kennwortfähiges AMI: AWSSupport -EC2Rescue: Kennwortaktiviertes AMI aus Instanz-ID`*

Sie finden diese AMIs, indem Sie nach der Automation-Ausführungs-ID suchen.

Für Linux wird der neue private SSH-Schlüssel für Ihre Instance verschlüsselt in Parameter Store gespeichert. Der Parametername lautet /ec2rl/openssh/***`instance ID`***/key.

Übergabe von Daten an Automation mithilfe von Eingangstransformatoren

Dieses AWS Systems Manager-Automation-Tutorial zeigt, wie Sie die Funktion des Eingabetransformators von Amazon EventBridge verwenden, um die `instance-id` einer Amazon Elastic Compute Cloud (Amazon EC2)-Instance aus einem Instance-Statusänderungsereignis zu extrahieren. Automation ist eine Funktion von AWS Systems Manager. Wir verwenden den Eingangstransformator, um diese Daten als `InstanceId`-Eingabeparameter an das `AWS-CreateImage`-Runbook zu übergeben. Die Regel wird ausgelöst, wenn eine beliebige Instance in den Status „`stopped`“ übergeht.

Weitere Informationen zum Arbeiten mit Eingabetransformatoren finden Sie unter Anleitung: Verwenden des Eingangstransformators zur Anpassung der an das Ereignisziel übergebenen Daten im Amazon EventBridge Benutzerhandbuch.

Bevor Sie beginnen

Stellen Sie sicher, dass Sie der Systems Manager-Automatisierungs-Servicerolle die erforderlichen Berechtigungen und Vertrauensrichtlinien für EventBridge hinzugefügt haben. Weitere Informationen finden Sie unter Übersicht über die Verwaltung von Zugriffsberechtigungen für Ihre EventBridge-Ressourcen im Amazon EventBridge-Benutzerhandbuch.

So verwenden Sie Eingangstransformatoren mit Automatisierung

1. Öffnen Sie die Amazon EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.

3. Wählen Sie Create rule (Regel erstellen).
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
5. Wählen Sie als Event bus (Event Bus) den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf übereinstimmende Ereignisse reagiert, die von Ihrem eigenen AWS-Konto stammen, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es stets an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus.
7. Wählen Sie Next (Weiter).
8. Wählen Sie für Event source (Ereignisquelle) AWS events or EventBridge partner events (-Ereignisse oder EventBridge-Partnerereignisse).
9. Wählen Sie im Abschnitt Ereignismuster die Option Ereignismusterformular aus.
10. Als Event source (Ereignisquelle) wählen Sie AWS-Services aus.
11. Wählen Sie für AWS-Service EC2 aus.
12. Wählen Sie in Event Type (Ereignistyp) EC2 Instance State-change Notification (Benachrichtigung über die Statusänderung der EC2-Instance) aus
13. Für Specific state(s) (Spezifische(r) Zustand(e)), wählen Sie stopped (gestoppt).
14. Wählen Sie Next (Weiter).
15. Bei Target types (Zieltypen) wählen Sie AWS-Service aus.
16. Für Select target (Ziel auswählen), wählen Sie Systems Manager Automation.
17. Wählen Sie unter Document (Dokument) die Option AWS-CreateImage aus.
18. Wählen Sie im Abschnitt Configure automation parameter(s) (Automatisierungsparameter konfigurieren) Input Transformer (Eingangstransformator) aus.
19. Geben Sie für Input path (Eingabepfad) den Wert **`{"instance":"$.detail.instance-id"}`** ein.
20. Geben Sie für Template (Vorlage) den Wert **`{"InstanceId":[<instance>]}`** ein.
21. Wählen Sie für Execution role (Ausführungsrolle) die Option Use existing role (Vorhandene Rolle) verwenden und wählen Sie Ihre Automation-Servicerolle.
22. Wählen Sie Next (Weiter).

23. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Tagging Your Amazon EventBridge Resources (Taggen Ihrer Amazon EventBridge Resources) im Amazon EventBridge-Benutzerhandbuch.
24. Wählen Sie Next (Weiter).
25. Überprüfen Sie die Details der Regel und wählen Sie dann Create rule (Regel erstellen) aus.

Grundlegendes zu Automatisierungsstatus

AWS Systems Manager Die Automatisierung meldet detaillierte Statusinformationen über die verschiedenen Status, die eine Automatisierungsaktion oder ein Automatisierungsschritt beim Ausführen einer Automatisierung durchläuft, und für die gesamte Automatisierung. Automatisierung ist eine Fähigkeit von. AWS Systems Manager Sie können Automatisierungsstatus mithilfe der folgenden Methoden überwachen:

- Überwachen Sie den Ausführungsstatus in der Systems Manager Automation-Konsole.
- Verwenden Sie Ihre bevorzugten Befehlszeilen-Tools. Für AWS Command Line Interface (AWS CLI) können Sie describe-automation-step-executions oder get-automation-execution verwenden. Für die können Sie Get-SSM Execution oder Get-SSM verwenden. AWS Tools for Windows PowerShell AutomationStep AutomationExecution
- Konfigurieren Sie Amazon so EventBridge , dass es auf Änderungen des Aktions- oder Automatisierungsstatus reagiert.

Weitere Informationen zum Umgang mit Timeouts in einer Automatisierung finden Sie unterBehandeln von Timeouts in Runbooks.

Informationen zu Automatisierungsstatus

Automation meldet Statusdetails für einzelne Automatisierungsaktionen zusätzlich zur Gesamtautomatisierung.

Der Gesamtautomatisierungsstatus kann von dem Status, der von einer einzelnen Aktion oder einem Schritt gemeldet wird, wie in den folgenden Tabellen angegeben, abweichen.

Detaillierter Status für Aktionen

Status	Details
Ausstehend	Der Schritt wurde noch nicht ausgeführt. Wenn Ihre Automatisierung bedingte Aktionen verwendet, bleiben die Schritte in diesem Zustand, nachdem eine Automatisierung abgeschlossen wurde, wenn die Bedingung für die Ausführung des Schritts nicht erfüllt wurde. Schritte bleiben auch in diesem Zustand, wenn die Automatisierung abgebrochen wird, bevor der Schritt ausgeführt wird.
InProgress	Der Schritt läuft.
Ein Moment	Der Schritt wartet auf Eingabe.
Herzlichen Glückwunsch	Der Schritt wurde erfolgreich ausgeführt. Diese ist ein Terminalstatus.
TimedOut	Ein Schritt oder eine Genehmigung wurde nicht vor dem angegebenen Zeitüberschreitung szeitraum abgeschlossen. Diese ist ein Terminalstatus.
Abbrechen	Der Schritt wird gerade angehalten, nachdem er von einem Anforderer abgebrochen wurde.
Abgebrochen	Der Schritt wurde von einem Anforderer angehalten, bevor er abgeschlossen wurde. Diese ist ein Terminalstatus.
Fehlgeschlagen	Der Schritt wurde nicht erfolgreich abgeschlo ssen. Diese ist ein Terminalstatus.
Exited	Nur durch die `aws:loop`-Aktion zurückgek ehrt. Die Schleife wurde nicht vollständig abgeschlossen. Ein Schritt innerhalb der Schleife wurde mithilfe der Eigenschaften

Status	Details
	`nextStep`, `onCancel` oder `onFailure` zu einem Schritt außerhalb der Schleife verschoben.

Detaillierter Status für eine Automatisierung

Status	Details
Ausstehend	Die Automatisierung hat noch nicht begonnen.
InProgress	Die Automatisierung läuft.
Ein Moment	Die Automatisierung wartet auf Eingabe.
Herzlichen Glückwunsch	Die Automatisierung wurde erfolgreich abgeschlossen. Diese ist ein Terminalstatus.
TimedOut	Ein Schritt oder eine Genehmigung wurde nicht vor dem angegebenen Zeitüberschreitung szeitraum abgeschlossen. Diese ist ein Terminalstatus.
Abbrechen	Die Automatisierung wird gerade angehalten, nachdem sie von einem Anforderer abgebrochen wurde.
Abgebrochen	Die Automatisierung wurde von einem Anforderer angehalten, bevor diese abgeschlossen wurde. Diese ist ein Terminalstatus.
Fehlgeschlagen	Die Automatisierung wurde nicht erfolgreich abgeschlossen. Diese ist ein Terminalstatus.

Fehlerbehebung für Systems Manager Automation.

Verwenden Sie die folgenden Informationen, um Probleme mit AWS Systems Manager Automation zu beheben, einer Funktion von AWS Systems Manager. Dieses Thema enthält spezifische Aufgaben zum Beheben von Problemen basierend auf Automation-Fehlermeldungen.

Themen

- Häufige Automation-Fehler
- Fehler beim Start der Automation-Ausführung
- Ausführung gestartet, aber Status ist fehlgeschlagen
- Ausführung gestartet, aber mit Zeitüberschreitung

Häufige Automation-Fehler

Dieser Abschnitt enthält Informationen zu gängigen Automation-Fehlern.

VPC nicht definiert 400

Standardmäßig erstellt das System eine temporäre Instance in der Standard-VPC (172.30.0.0/16), wenn Automation das Runbook `AWS-UpdateLinuxAmi` oder das Runbook `AWS-UpdateWindowsAmi` ausführt. Wenn Sie die Standard-VPC gelöscht haben, erhalten Sie den folgenden Fehler:

```
VPC not defined 400
```

Um dieses Problem zu lösen, müssen Sie einen Wert für den `SubnetId`-Eingabeparameter angeben.

Fehler beim Start der Automation-Ausführung

Eine Automatisierung kann mit einem Fehler „Zugriff verweigert“ oder einem ungültigen Fehler „Rolle übernehmen“ fehlschlagen, wenn Sie Rollen und Richtlinien für die Automatisierung nicht ordnungsgemäß konfiguriert haben AWS Identity and Access Management (IAM).

Zugriff verweigert

Die folgenden Beispiele beschreiben Situationen, in denen eine Automatisierung nicht gestartet werden konnte, weil der Zugriff verweigert wurde.

Zugriff auf die Systems Manager API verweigert

Fehlermeldung: `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)`

- Mögliche Ursache 1: Der Benutzer, der versucht, die Automatisierung zu starten, verfügt nicht über die Berechtigung zum Aufrufen der `StartAutomationExecution`-API. Um dieses Problem zu beheben, fügen Sie die erforderliche IAM-Richtlinie dem Benutzer an, der zum Starten der Automatisierung verwendet wurde.
- Mögliche Ursache 2: Der Benutzer, der versucht, die Automatisierung zu starten, verfügt über die Berechtigung zum Aufrufen der `StartAutomationExecution`-API, jedoch nicht über die Berechtigung zum Aufrufen der API mithilfe des spezifischen Runbooks. Um dieses Problem zu beheben, fügen Sie die erforderliche IAM-Richtlinie dem Benutzer an, der zum Starten der Automatisierung verwendet wurde.

Der Zugriff wurde aufgrund fehlender PassRole Berechtigungen verweigert

Fehlermeldung: `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)`

Der Benutzer, der versucht, die Automatisierung zu starten, hat keine PassRole Berechtigung für die Übernahme der Rolle. Um dieses Problem zu beheben, fügen Sie die iam: PassRole -Richtlinie der Rolle des Benutzers hinzu, der versucht, die Automatisierung zu starten. Weitere Informationen finden Sie unter Aufgabe 2: Hängen Sie die iam: PassRole -Richtlinie an Ihre Automation-Rolle an.

Ungültige Übernahmerolle

Beim Ausführen einer Automatisierung wird eine Übernahmerolle entweder im Runbook bereitgestellt oder als Parameterwert für das Runbook weitergeleitet. Unterschiedliche Arten von Fehlern können auftreten, wenn die Übernahmerolle nicht angegeben oder nicht ordnungsgemäß konfiguriert ist.

Falsch formatierte Übernahmerolle

Fehlermeldung: `The format of the supplied assume role ARN isn't valid.`
Die Übernahmerolle ist falsch formatiert. Um dieses Problem zu lösen, stellen Sie sicher, dass beim Starten der Automatisierung eine gültige Übernahmerolle in Ihrem Runbook oder als Laufzeitparameter angegeben ist.

Rolle annehmen kann nicht übernommen werden

Fehlermeldung: `The defined assume role is unable to be assumed. (Service: AWSSimpleSystemsManagement; Status Code: 400; Error Code: InvalidAutomationExecutionParametersException; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)`

- Mögliche Ursache 1: Die Übernahmerolle ist nicht vorhanden. Sie lösen dieses Problem, indem Sie die Rolle erstellen. Weitere Informationen finden Sie unter the section called “Einrichten der Automatisierung”. Spezifische Details zum Erstellen dieser Rolle sind im folgenden Thema beschrieben: Aufgabe 1: Erstellen einer Servicerolle für Automation.
- Mögliche Ursache 2: Die Rollenübernahme hat keine Vertrauensbeziehung zum Systems Manager-Service. Um dieses Problem zu lösen, erstellen Sie die Vertrauensbeziehung. Weitere Informationen finden Sie unter Ich kann keine Rolle übernehmen im IAM-Benutzerhandbuch.

Ausführung gestartet, aber Status ist fehlgeschlagen

Aktionsspezifische Fehler

Runbooks enthalten Schritte und die Schritte werden der Reihe nach ausgeführt. Jeder Schritt ruft mindestens eine AWS-Service -API auf. Die APIs bestimmen die Eingaben, das Verhalten und die Ausgaben des Schritts. Es gibt mehrere Stellen, an denen ein Fehler kann dazu führen, dass ein Schritt fehlschlägt. Fehlermeldungen geben an, wann und wo ein Fehler aufgetreten ist.

Wenn Sie eine Fehlermeldung in der Amazon Elastic Compute Cloud (Amazon EC2)-Konsole sehen, wählen Sie den Link View Outputs (Ausgaben anzeigen) für den fehlgeschlagenen Link. Um eine Fehlermeldung von zu sehen AWS CLI, rufen Sie an `get-automation-execution` und suchen Sie nach dem `FailureMessage` Attribut in einem Fehler`StepExecution`.

In den folgenden Beispielen ist ein Schritt im Zusammenhang mit der `aws:runInstance`-Aktion fehlgeschlagen. Jedes Beispiel untersucht einen anderen Fehlertyp.

Fehlendes Image

Fehlermeldung: `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.
```

Die `aws:runInstances`-Aktion erhält eine Eingabe für eine nicht vorhandene `ImageId`. Um dieses Problem zu beheben, aktualisieren Sie das Runbook oder Parameterwerte mit der richtigen AMI-ID.

Gehen Sie davon aus, dass der Rollenrichtlinie die erforderlichen Berechtigungen fehlen

Fehlermeldung: `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.`

Die Übernahmerolle verfügt nicht über ausreichende Berechtigungen zum Aufrufen der `RunInstances`-API auf EC2-Instances. Um dieses Problem zu beheben, fügen Sie der Übernahmerolle eine IAM-Richtlinie hinzu, die über die Berechtigung zum Aufrufen der `RunInstances`-API verfügt. Weitere Informationen hierzu finden Sie unter Methode 2: Konfigurieren von Automation-Rollen mit IAM.

Unerwarteter Status

Fehlermeldung: `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.`

- Mögliche Ursache 1: Es liegt ein Problem mit der Instance oder dem Amazon EC2-Service vor. Um dieses Problem zu beheben, melden Sie sich bei der Instance an oder überprüfen Sie das Instance-Systemprotokoll, um zu verstehen, warum die Instance mit dem Herunterfahren begonnen hat.
- Mögliche Ursache 2: Das angegebene Benutzerdatenskript für die `aws:runInstances`-Aktion weist ein Problem oder eine falsche Syntax auf. Überprüfen Sie die Syntax des Benutzerdatenskripts. Stellen Sie außerdem sicher, dass die Benutzerdatenskripts die Instance nicht herunterfahren oder andere Skripts aufrufen, die die Instance herunterfahren.

Aktionsspezifische Fehlerverweise

Sollte ein Schritt fehlschlagen, gibt die Fehlermeldung an, welcher Service aufgerufen wurde, als der Fehler aufgetreten ist. In der folgenden Tabelle sind die von der jeweiligen Aktion aufgerufenen Services aufgelistet. Die Tabelle enthält außerdem Links zu Informationen über jeden Service.

Aktion	AWS-Services durch diese Aktion aufgerufen	Weitere Informationen zu diesem Service	Inhalt der Fehlerbehebung
`aws:runInstances`	Amazon EC2	Amazon EC2 EC2-Benutzerhandbuch	Fehlerbehebung bei EC2-Instances
`aws:changeInstanceState`	Amazon EC2	Amazon EC2 EC2-Benutzerhandbuch	Fehlerbehebung bei EC2-Instances
`aws:runCommand`	Systems Manager	AWS Systems Manager Run Command	Fehlerbehebung von Systems Manager Run Command
`aws:createImage`	Amazon EC2	Amazon Machine Images	
`aws:createStack`	AWS CloudFormation	AWS CloudFormation Benutzerhandbuch	Fehlersuche AWS CloudFormation
`aws:deleteStack`	AWS CloudFormation	AWS CloudFormation Benutzerhandbuch	Fehlersuche AWS CloudFormation
`aws:deleteImage`	Amazon EC2	Amazon Machine Images	
`aws:copyImage`	Amazon EC2	Amazon Machine Images	
`aws:createTag`	Amazon EC2, Systems Manager	EC2-Ressourcen und -Tags	
`aws:invokeLambdaFunction`	AWS Lambda	AWS Lambda Entwicklerhandbuch	Problembehebung bei Lambda

Interner Fehler bei Automation-Service

Fehlermeldung: `Internal Server Error. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

Ein Problem mit dem Automation-Service verhindert, dass das angegebene Runbook korrekt ausgeführt wird. Um dieses Problem zu lösen, wenden Sie sich an AWS Support. Geben Sie die Ausführungs-ID und Kunden-ID an, wenn verfügbar.

Ausführung gestartet, aber mit Zeitüberschreitung

Fehlermeldung: `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.`

Für einen Schritt in der `aws:runInstances`-Aktion ist eine Zeitüberschreitung aufgetreten. Dies kann der Fall sein, wenn die Schrittaktion länger dauert als der angegebene Wert für `timeoutSeconds` im Schritt. Um dieses Problem zu lösen, geben Sie einen längeren Wert für den `timeoutSeconds`-Parameter in der `aws:runInstances`-Aktion an. Wenn das Problem dadurch nicht behoben werden kann, untersuchen Sie, warum der Schritt länger dauert als erwartet

AWS Systems Manager Change Calendar

Change Calendar, eine Funktion vonAWS Systems Manager, erlaubt Ihnen Datums- und Uhrzeitbereiche einzurichten, in denen von Ihnen angegebene Aktionen (z B. in Systems Manager Automation-Runbooks) in Ihrem AWS-Konto ausgeführt oder nicht ausgeführt werden können. In Change Calendar werden diese Bereiche Ereignisse genannt. Wenn Sie einen Change Calendar-Eintrag erstellen, erstellen Sie ein Systems Manager-Dokument vom Typ `ChangeCalendar`. In Change Calendar wird das Dokument als iCalendar 2.0-Daten im Klartextformat gespeichert. Ereignisse, die Sie dem Change Calendar-Eintrag hinzufügen, werden Teil des Dokuments. Um mit Change Calendar zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Change Calendar aus.

Sie können einen Kalender und seine Ereignisse in der Systems Manager Konsole erstellen. Sie können auch eine iCalendar (`.ics`)-Datei importieren, die Sie von einem unterstützten Drittanbieter-Kalenderanbieter exportiert haben, um dessen Ereignisse Ihrem Kalender hinzuzufügen. Zu den unterstützten Anbietern zählen Google Kalender, Microsoft Outlook und iCloud-Kalender.

Ein Change Calendar-Eintrag kann zwei Typen haben:

`DEFAULT_OPEN` oder "Standardmäßig geöffnet"

Alle Aktionen können standardmäßig ausgeführt werden, außer während Kalenderereignissen. Während der Ereignisse lautet der Status eines `DEFAULT_OPEN`-Kalenders `CLOSED`. Die Ausführung von Ereignissen wird dann blockiert.

`DEFAULT_CLOSED` oder "Standardmäßig geschlossen"

Alle Aktionen werden standardmäßig blockiert, außer während Kalenderereignissen. Während der Ereignisse lautet der Status eines `DEFAULT_CLOSED`-Kalenders `OPEN`. Aktionen dürfen ausgeführt werden.

Sie können festlegen, dass alle geplanten Automatisierungsworkflows, Wartungsfenster und State Manager-Zuordnungen automatisch zu einem Kalender hinzugefügt werden. Sie können außerdem jeden dieser einzelnen Typen aus der Kalenderanzeige entfernen.

An wen richtet sich Change Calendar?

- AWS-Kunden, die die folgenden Aktionstypen ausführen:
  - Automatisierungs-Runbooks erstellen oder ausführen.
  - Änderungsanforderung in Change Manager erstellen.
  - Wartungsfenster ausführen.
  - Assoziationen in State Manager erstellen.

  Automatisierung, Change Manager, Maintenance Windows und State Manager sind alle Fähigkeiten von AWS Systems Manager. Durch die Integration dieser Funktionen in Change Calendar, können Sie diese Aktionstypen je nach dem aktuellen Status des Änderungskalenders, den Sie mit jedem einzelnen verknüpfen, zulassen oder blockieren.
- Administratoren, die dafür verantwortlich sind, die Konfigurationen von verwalteten Systems-Manager-Knoten konsistent, stabil und funktionsfähig zu halten.

Vorteile von Change Calendar

Im Folgenden sind einige Vorteile von Change Calendar aufgeführt.

- Änderungen überprüfen, bevor sie angewendet werden

Ein Change Calendar-Eintrag kann dazu beitragen, dass potenziell destruktive Änderungen in Ihrer Umgebung überprüft werden, bevor sie angewendet werden.

- Änderungen nur zu angemessenen Zeiten anwenden

  Change Calendar-Einträge helfen, Ihre Umgebung während der Ereigniszeiten stabil zu halten. Beispielsweise können Sie einen Change Calendar-Eintrag erstellen, um Änderungen zu blockieren, wenn Sie eine hohe Nachfrage nach Ihren Ressourcen erwarten wird (z. B. während einer Konferenz oder einer öffentlichen Marketingaktion). Ein Kalendereintrag kann auch Änderungen blockieren, wenn Sie eine eingeschränkte Administratorunterstützung erwarten, z. B. während eines Urlaubs oder einer Urlaubszeit. Sie können einen Kalendereintrag verwenden, um Änderungen außer zu bestimmten Tages- oder Wochenzeiten zuzulassen, zu denen nur begrenzter Administratorsupport zur Fehlerbehebung bei fehlgeschlagenen Aktionen oder Bereitstellungen zur Verfügung steht.
- Aktuellen oder bevorstehenden Status des Kalenders abrufen

  Sie können die Systems Manager `GetCalendarState`-API-Operation ausführen, um Ihnen den aktuellen Status des Kalenders, den Status zu einer bestimmten Zeit oder den nächsten geplanten Wechsel des Kalenderstatus anzuzeigen.
- EventBridge-Unterstützung

  Diese Systems Manager Funktion wird als Event (Ereignis)-Typ in Amazon EventBridge Regeln unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager.

Themen

- Einrichten von Change Calendar
- Arbeiten mit Change Calendar
- Hinzufügen von Change Calendar-Abhängigkeiten zu Automation-Runbooks
- Fehlerbehebung für Change Calendar

Einrichten von Change Calendar

Führen Sie Folgendes ausChange Calendar, bevor Sie eine Funktion von verwenden AWS Systems Manager.

Installieren der neuesten Befehlszeilen-Tools

Installieren Sie die neuesten Befehlszeilen-Tools, um Statusinformationen über Kalender zu erhalten.

Anforderung	Beschreibung
AWS CLI	(Optional) Um mit AWS Command Line Interface (AWS CLI) Statusinformationen zu Kalendern abzurufen, installieren Sie die neueste Version von AWS CLI auf Ihrem lokalen Computer. Weitere Informationen zum Installieren oder Upgraden der CLI finden Sie unter Installieren, Aktualisieren und Deinstallieren der AWS CLI im AWS Command Line Interface -Benutzer handbuch.
AWS Tools for PowerShell	(Optional) Um die Tools für zum Abrufen von Statusinformationen PowerShell zu Kalendern zu verwenden, installieren Sie die neueste Version von Tools für PowerShell auf Ihrem lokalen Computer. Weitere Informationen zur Installation oder Aktualisierung der Tools für PowerShell finden Sie unter Installieren von AWS Tools for PowerShell im AWS Tools for PowerShell Benutzerhandbuch.

Berechtigungen einrichten

Wenn Ihrem Benutzer, Ihrer Gruppe oder Rolle Administratorrechte zugewiesen sind, haben Sie vollen Zugriff auf Change Calendar. Wenn Sie nicht über Administratorrechte verfügen, muss Ihnen ein Administrator die Berechtigung erteilen, indem er entweder die von `AmazonSSMFullAccess` verwaltete Richtlinie zuweist oder eine Richtlinie, die Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle die erforderlichen Berechtigungen erteilt, zuweist.

Die folgenden Berechtigungen sind erforderlich, um mit Change Calendar zu arbeiten.

Change Calendar-Einträge

Um einen Change Calendar-Eintrag zu erstellen, zu aktualisieren oder zu löschen, einschließlich des Hinzufügens und Entfernens von Ereignissen aus dem Eintrag, muss eine Richtlinie, die Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zugeordnet ist, die folgenden Aktionen erlauben:

- `ssm:CreateDocument`
- `ssm:DeleteDocument`
- `ssm:DescribeDocument`
- `ssm:DescribeDocumentPermission`
- `ssm:GetCalendar`
- `ssm:ListDocuments`
- `ssm:ModifyDocumentPermission`
- `ssm:PutCalendar`
- `ssm:UpdateDocument`
- `ssm:UpdateDocumentDefaultVersion`

Kalenderstatus

Um Informationen über den aktuellen oder bevorstehenden Status des Kalenders zu erhalten, muss eine Richtlinie, die Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zugeordnet ist, die folgende Aktion erlauben:

- `ssm:GetCalendarState`

Betriebliche Ereignisse

Um betriebliche Ereignisse wie Wartungsfenster, Zuordnungen und geplante Automatisierungen anzuzeigen, muss die Richtlinie, die Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zugeordnet ist, die folgenden Aktionen zulassen:

- `ssm:DescribeMaintenanceWindows`
- `ssm:DescribeMaintenanceWindowExecution`
- `ssm:DescribeAutomationExecutions`
- `ssm:ListAssociations`

> **Note**
>
> Change Calendar-Einträge, die anderen Konten als Ihrem gehören (d. h. von anderen erstellt wurden), sind schreibgeschützt, auch wenn sie mit Ihrem Konto geteilt werden. Wartungsfenster, State Manager Verknüpfungen und Automatisierungen werden nicht gemeinsam genutzt.

Arbeiten mit Change Calendar

Mit der AWS Systems Manager-Konsole können Sie Einträge in Change Calendar, eine Funktion von AWS Systems Manager, hinzufügen, verwalten oder löschen. Sie können Ereignisse auch von unterstützten Drittanbieter-Kalenderanbietern importieren, indem Sie eine iCalendar (`.ics`) -Datei importieren, die Sie aus dem Quellkalender exportiert haben. Und Sie können die `GetCalendarState`-API-Operation oder den `get-calendar-state` AWS Command Line Interface (AWS CLI)-Befehl verwenden, um Informationen über den Status von Change Calendar zu einem bestimmten Zeitpunkt zu erhalten.

Themen

- Erstellen eines Änderungskalenders
- Erstellen und Verwalten von Ereignissen in Change Calendar
- Importieren und Verwalten von Ereignissen aus Drittanbieter-Kalendern
- Aktualisieren eines Änderungskalenders
- Freigeben eines Änderungskalenders
- Einen Änderungskalender löschen
- Abrufen des Status eines Änderungskalenders

Erstellen eines Änderungskalenders

Wenn Sie einen Eintrag in Change Calendar, eine Funktion von AWS Systems Manager, erstellen, erstellen Sie ein Systems Manager-Dokument (SSM-Dokument), das das `text`-Format verwendet.

So erstellen Sie einen Änderungskalender

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie Create calendar (Kalender erstellen).

   –oder–

   Wenn die Homepage von Change Calendar zuerst geöffnet wird, wählen Sie Create change calendar (Änderungskalender erstellen) aus.
4. Geben Sie auf der Seite Create calendar (Kalender erstellen) unter Calendar details (Kalenderdetails) einen Namen für Ihren Kalendereintrag ein. Namen von Kalendereinträgen können Buchstaben, Zahlen, Punkte, Striche und Unterstriche enthalten. Der Name sollte spezifisch genug sein, um den Zweck des Kalendereintrags auf einen Blick zu erkennen. Ein Beispiel ist **`support-off-hours`**. Sie können diesen Namen nicht mehr aktualisieren, nachdem Sie den Kalendereintrag erstellt haben.
5. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung für Ihren Kalendereintrag ein.
6. (Optional) Klicken Sie im Bereich Importieren eines Kalanders auf Datei auswählen, um eine iCalendar (`.ics`)-Datei auszuwählen, die Sie von einem Drittanbieter-Kalenderanbieter exportiert haben. Beim Importieren der Datei werden die Ereignisse zu Ihrem Kalender hinzugefügt.

   Zu den unterstützten Anbietern zählen Google Kalender, Microsoft Outlook und iCloud-Kalender.

   Weitere Informationen finden Sie unter Importieren von Ereignissen von Drittanbietern-Kalenderanbietern.
7. Wählen Sie in Calendar type (Kalendertyp) eine der folgenden Optionen.

   - Open by default (Standardmäßig geöffnet) - der Kalender ist geöffnet (Automatisierungsaktionen können bis zum Start eines Ereignisses ausgeführt werden) und dann für die Dauer eines zugehörigen Ereignisses geschlossen.
   - Closed by default (Standardmäßig geschlossen) - der Kalender ist geschlossen (Automatisierungsaktionen können erst ab dem Beginn eines Ereignisses ausgeführt werden), aber für die Dauer eines zugehörigen Ereignisses geöffnet.
8. (Optional) Wählen Sie unter Änderungsmanagementereignisse die Option Änderungsmanagementereignisse zum Kalender hinzufügen aus. Mit dieser Auswahl werden alle geplanten Wartungsfenster, State Manager-Verknüpfungen, Automatisierungsworkflows und Change Manager-Änderungsanforderungen in Ihrer monatlichen Kalenderanzeige angezeigt.

> **Tip**
>
> Wenn Sie diese Ereignistypen später dauerhaft aus der Kalenderanzeige entfernen möchten, bearbeiten Sie den Kalender, deaktivieren Sie dieses Kontrollkästchen und wählen Sie dann Speichern.

9. Wählen Sie Create calendar (Kalender erstellen).

   Nachdem der Kalendereintrag erstellt wurde, zeigt Systems Manager Ihren Kalendereintrag in der Liste Change Calendar an. Die Spalten zeigen die Kalenderversion und die AWS-Konto-Kontonummer des Kalenderbesitzers an. Ihr Kalendereintrag kann keine Aktionen verhindern oder zulassen, bis Sie mindestens ein Ereignis erstellt oder importiert haben. Informationen zum Erstellen eines Ereignisses finden Sie unter Erstellen eines Change Calendar-Ereignisses. Weitere Informationen zum Importieren von Ereignissen finden Sie unter Importieren von Ereignissen von Drittanbietern-Kalenderanbietern.

Erstellen und Verwalten von Ereignissen in Change Calendar

Nachdem Sie einen Kalender in AWS Systems Manager Change Calendar erstellt haben, können Sie Ereignisse erstellen, aktualisieren und löschen, die in Ihrem geöffneten oder geschlossenen Kalender enthalten sind. Change Calendar ist eine Funktion von AWS Systems Manager.

> **Tip**
>
> Alternativ zum Erstellen von Ereignissen direkt in der Systems Manager-Konsole können Sie eine iCalendar (`.ics`)-Datei aus einer unterstützten Kalenderanwendung eines Drittanbieters importieren. Weitere Informationen finden Sie unter Importieren und Verwalten von Ereignissen aus Drittanbieter-Kalendern.

Themen

- Erstellen eines Change Calendar-Ereignisses
- Aktualisieren eines Change Calendar-Ereignisses
- Löschen eines Change Calendar-Ereignisses

Erstellen eines Change Calendar-Ereignisses

Wenn Sie ein Ereignis zu einem Eintrag in Change Calendar hinzufügen, eine Funktion von AWS Systems Manager, geben Sie einen Zeitraum an, in dem die Standardaktion des Kalendereintrags ausgesetzt wird. Wenn der Kalendereintragstyp beispielsweise standardmäßig geschlossen ist, ist der Kalender für Änderungen während der Ereignisse geöffnet. (Alternativ können Sie ein empfohlenes Ereignis erstellen, das im Kalender nur der Information dient.)

Derzeit können Sie ein Change Calendar-Ereignis nur über die Konsole erstellen. Ereignisse werden zum Change Calendar-Dokument hinzugefügt, das Sie beim Erstellen eines Change Calendar-Eintrags erstellen.

Erstellen eines Change Calendar-Ereignisses

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalendereintrags aus, dem Sie ein Ereignis hinzufügen möchten.
4. Wählen Sie auf der Detailseite des Kalendereintrags Create event (Ereignis erstellen).
5. Geben Sie auf der Seite Create scheduled event (Geplantes Ereignis erstellen) unter Event details (Ereignisdetails) einen Anzeigenamen für Ihr Ereignis ein. Ereignisnamen können Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten. Der Name sollte spezifisch genug sein, um den Zweck des Ereignisses zu identifizieren. Ein Beispiel ist **`nighttime-hours`**.
6. Geben Sie unter Description (Beschreibung) eine Beschreibung für Ihr Ereignis ein. Zum Beispiel **`The support team isn't available during these hours`**.
7. (Optional) Wenn dieses Ereignis nur als visuelle Benachrichtigung oder Erinnerung dienen soll, aktivieren Sie das Kontrollkästchen Advisory (Empfehlung). Empfohlene Ereignisse haben im Kalender keine konkrete Funktion. Sie dienen nur zu Information für diejenigen, die den Kalender anzeigen.
8. Geben Sie unter Event start date (Startdatum des Ereignisses) einen Tag im Format `MM/DD/YYYY` ein oder wählen Sie einen Tag aus, an dem das Ereignis gestartet werden soll. Geben Sie außerdem eine Uhrzeit an dem angegebenen Tag im Format `hh:mm:ss` (Stunden, Minuten und Sekunden) ein, zu der das Ereignis starten soll.

9. Geben Sie unter Event end date (Enddatum des Ereignisses) einen Tag im Format `MM/DD/YYYY` ein oder wählen Sie einen Tag aus, an dem das Ereignis enden soll. Geben Sie außerdem eine Uhrzeit an dem angegebenen Tag im Format `hh:mm:ss` (Stunden, Minuten und Sekunden) ein, zu der das Ereignis enden soll.
10. Wählen Sie unter Schedule time zone (Zeitzone des Zeitplans) eine Zeitzone, die für die Start- und Endzeit des Ereignisses gilt. Sie können einen Teil des Stadtnamens oder den Zeitzonenunterschied zu Greenwich Mean Time (GMT) eingeben, um eine Zeitzone schneller zu finden. Die Standardeinstellung ist Coordinated Universal Time (UTC).
11. (Optional) Um ein täglich, wöchentlich oder monatlich wiederkehrendes Ereignis zu erstellen, aktivieren Sie Recurrence (Wiederholung). Geben Sie dann die Häufigkeit und das optionale Enddatum der Wiederholung an.
12. Wählen Sie Create scheduled event (Zeitgesteuertes Ereignis erstellen). Das neue Ereignis wird zu Ihrem Kalendereintrag hinzugefügt und auf der Registerkarte Events (Ereignisse) der Detailseite des Kalendereintrags angezeigt.

Aktualisieren eines Change Calendar-Ereignisses

Gehen Sie wie folgt vor, um ein Change Calendar-Ereignis in der AWS Systems Manager-Konsole zu aktualisieren. Change Calendarist eine Funktion von AWS Systems Manager.

Aktualisieren eines Change Calendar-Ereignisses

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalendereintrags, für den Sie ein Ereignis bearbeiten möchten.
4. Wählen Sie auf der Detailseite des Kalendereintrags Events (Ereignisse).
5. Wählen Sie auf der Kalenderseite das Ereignis, das Sie bearbeiten möchten.

> **Tip**
>
> Verwenden Sie die Schaltflächen oben links, um ein Jahr oder einen Monat zurück oder vorwärts zu gehen. Ändern Sie bei Bedarf die Zeitzone, indem Sie die richtige Zeitzone in der Liste oben rechts auswählen.

6. Wählen Sie unter Event details (Ereignisdetails) die Option Edit (Bearbeiten) aus.

   Um den Namen und die Beschreibung des Ereignisses zu ändern, ergänzen oder ersetzen Sie den aktuellen Text.
7. Um den Wert Event start date (Startdatum des Ereignisses) zu ändern, wählen Sie das aktuelle Startdatum und dann ein neues Datum im Kalender aus. Um die Startzeit zu ändern, wählen Sie die aktuelle Startzeit und dann eine neue Uhrzeit in der Liste aus.
8. Um den Wert Event end date (Enddatum des Ereignisses) zu ändern, wählen Sie das aktuelle Enddatum und dann ein neues Datum im Kalender aus. Um die Endzeit zu ändern, wählen Sie die aktuelle Endzeit und dann eine neue Uhrzeit in der Liste aus.
9. Um den Wert Schedule time zone (Zeitzone des Zeitplans) zu ändern, wählen Sie eine Zeitzone aus, die für die Start- und Endzeit des Ereignisses gelten soll. Sie können einen Teil des Stadtnamens oder den Zeitzonenunterschied zu Greenwich Mean Time (GMT) eingeben, um eine Zeitzone schneller zu finden. Die Standardeinstellung ist Coordinated Universal Time (UTC).
10. (Optional) Wenn dieses Ereignis nur als visuelle Benachrichtigung oder Erinnerung dienen soll, aktivieren Sie das Kontrollkästchen Advisory (Empfehlung). Empfohlene Ereignisse haben im Kalender keine konkrete Funktion. Sie dienen nur zu Information für diejenigen, die den Kalender anzeigen.
11. Wählen Sie Save (Speichern). Ihre Änderungen werden auf der Registerkarte Events (Ereignisse) der Detailseite des Kalendereintrags angezeigt. Wählen Sie das Ereignis, das Sie aktualisiert haben, um Ihre Änderungen anzuzeigen.

Löschen eines Change Calendar-Ereignisses

Sie können jeweils ein Ereignis mithilfe der AWS Management Console in Change Calendar löschen, eine Funktion von AWS Systems Manager.

> **Tip**
>
> Wenn Sie bei der Erstellung des Kalenders die Option Änderungsmanagementereignisse zum Kalender hinzufügen gewählt haben, können Sie Folgendes tun:
>
> - Um einen Typ eines Änderungsmanagementereignisses vorübergehend aus der Kalenderanzeige auszublenden, wählen Sie für den Typ das X oben in der Monatsvorschau.
> - Um diese Typen dauerhaft aus der Kalenderanzeige zu entfernen, bearbeiten Sie den Kalender, deaktivieren Sie das Kontrollkästchen Änderungsmanagementereignisse

zum Kalender hinzufügen und wählen Sie dann Speichern. Wenn Sie Typen aus der Kalenderanzeige entfernen, werden sie nicht aus Ihrem Konto gelöscht.

Löschen eines Change Calendar-Ereignisses

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalendereintrags aus, aus dem Sie ein Ereignis löschen möchten.
4. Wählen Sie auf der Detailseite des Kalendereintrags Events (Ereignisse).
5. Wählen Sie auf der Kalenderseite das Ereignis, das Sie löschen möchten.

   Tip

   Verwenden Sie die Schaltflächen oben links, um den Kalender um ein Jahr oder einen Monat zurück oder vorwärts zu verschieben. Ändern Sie bei Bedarf die Zeitzone, indem Sie die richtige Zeitzone in der Liste oben rechts auswählen.

6. Wählen Sie auf der Seite Event details (Ereignisdetails(die Option Delete (Löschen). Wenn Sie aufgefordert werden, das Löschen des Ereignisses zu bestätigen, wählen Sie Confirm (Bestätigen) aus.

Importieren und Verwalten von Ereignissen aus Drittanbieter-Kalendern

Alternativ zum Erstellen von Ereignissen direkt in der AWS Systems Manager-Konsole können Sie eine iCalendar (`.ics`)-Datei aus einer unterstützten Kalenderanwendung eines Drittanbieters importieren. Der Kalender kann sowohl importierte Ereignisse als auch Ereignisse enthalten, die Sie in Change Calendar erstellen, was eine Funktion von AWS Systems Manager ist.

Bevor Sie beginnen

Bevor Sie versuchen, eine Kalenderdatei zu importieren, überprüfen Sie die folgenden Anforderungen und Einschränkungen:

Kalenderdateiformat

Nur gültige iCalendar-Dateien (`.ics`) werden unterstützt.

Unterstützte Kalenderanbieter

Nur `.ics`-Dateien, die von den folgenden Drittanbieter-Kalenderanbietern exportiert wurden, werden unterstützt:

- Google Calendar (Exportanweisungen)
- Microsoft Outlook (Exportanweisungen)
- iCloud Calendar (Exportanweisungen)

Dateigröße

Sie können eine beliebige Anzahl gültiger `.ics`-Dateien importieren. Die Gesamtgröße aller importierten Dateien für jeden Kalender darf jedoch 64 KB nicht überschreiten.

> **Tip**
>
> Zum Minimieren der Größe der `.ics`-Datei, stellen Sie sicher, dass Sie nur grundlegende Details zu Ihren Kalendereinträgen exportieren. Verringern Sie bei Bedarf die Länge des Zeitraums, den Sie exportieren.

Zeitzone

Neben einem Kalendernamen, einem Kalenderanbieter und mindestens einem Ereignis sollte Ihre exportierte `.ics`-Datei außerdem die Zeitzone für den Kalender angeben. Wenn dies nicht der Fall ist oder ein Problem bei der Identifizierung der Zeitzone auftritt, werden Sie nach dem Importieren der Datei aufgefordert, eine anzugeben.

Wiederkehrende Ereigniseinschränkung

Ihre exportierte `.ics`-Datei kann wiederkehrende Ereignisse enthalten. Wenn jedoch ein oder mehrere Vorkommen eines wiederkehrenden Ereignisses im Quellkalender gelöscht wurden, schlägt der Import fehl.

Themen

- Importieren von Ereignissen von Drittanbietern-Kalenderanbietern

- Aktualisieren aller Ereignisse von einem Drittanbieter-Kalenderanbieter
- Löschen aller Ereignisse, die aus einem Kalender eines Drittanbieters importiert wurden

Importieren von Ereignissen von Drittanbietern-Kalenderanbietern

Gehen Sie wie folgt vor, um eine iCalendar (`.ics`)-Datei aus einer unterstützten Kalenderanwendung eines Drittanbieters zu importieren. Die in der Datei enthaltenen Ereignisse werden in die Regeln für Ihren offenen oder geschlossenen Kalender integriert. Sie können eine Datei in einen neuen Kalender importieren, den Sie mit Change Calendar erstellen (eine Funktion von AWS Systems Manager), oder in einen vorhandenen Kalender.

Nach dem Importieren der `.ics`-Datei, können Sie einzelne Ereignisse aus dieser mit der Change Calendar-Schnittstelle entfernen. Weitere Informationen finden Sie unter Löschen eines Change Calendar-Ereignisses. Sie können auch alle Ereignisse aus dem Quellkalender löschen, indem Sie die `.ics`-Datei löschen. Weitere Informationen finden Sie unter Löschen aller Ereignisse, die aus einem Kalender eines Drittanbieters importiert wurden.

So importieren Sie Ereignisse von Drittanbietern-Kalenderanbietern

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Um mit einem neuen Kalender zu beginnen, wählen Sie Kalender erstellen. Wählen Sie im Bereich Kalender Importieren Datei auswählen. Informationen zu anderen Schritten zum Erstellen eines neuen Kalenders finden Sie unter Erstellen eines Änderungskalenders.

   –oder–

   Um Ereignisse von Drittanbietern in einen vorhandenen Kalender zu importieren, wählen Sie den Namen eines vorhandenen Kalenders aus, um diesen zu öffnen.
4. Wählen Sie Actions, Edit (Aktionen, Bearbeiten) und dann im Bereich Import calendar (Kalender importieren) die Option Choose file (Datei auswählen) aus.
5. Navigieren Sie zu der exportierten `.ics`-Datei auf Ihrem lokalen Computer und wählen Sie sie aus.
6. Wenn Sie dazu aufgefordert werden, wählen Sie Select a time zone (Zeitzone auswählen), um zu bestimmen, welche Zeitzone für den Kalender gelten soll.
7. Wählen Sie Save (Speichern).

Aktualisieren aller Ereignisse von einem Drittanbieter-Kalenderanbieter

Wenn mehrere Ereignisse zu Ihrem Quellkalender hinzugefügt oder aus diesem entfernt werden, nachdem Sie dessen iCalendar `.ics`-Datei importiert haben, können Sie diese Änderungen in Change Calendar widerspiegeln. Exportieren Sie zunächst den Quellkalender neu und importieren Sie die neue Datei in Change Calendar, was eine Funktion von AWS Systems Manager ist. Ereignisse in Ihrem Änderungskalender werden aktualisiert, um den Inhalt der neueren Datei wiederzugeben.

So aktualisieren Sie alle Ereignisse eines Drittanbieter-Kalenderanbieters

1. Fügen Sie in Ihrem Kalender eines Drittanbieters Ereignisse hinzu oder entfernen Sie sie, wenn Sie möchten, dass sie in Change Calendar wiedergegeben werden, und exportieren Sie den Kalender dann erneut in eine neue `.ics`-Datei.
2. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Change Calendar aus.
4. Wählen Sie aus der Liste der Kalender den Kalendernamen aus der Liste aus.
5. Wählen Sie Datei auswählen, navigieren Sie zu der `.ics`-Ersatzdatei und wählen Sie diese aus.
6. Als Reaktion auf die Benachrichtigung über das Überschreiben der vorhandenen Datei wählen Sie Confirm (Bestätigen).

Löschen aller Ereignisse, die aus einem Kalender eines Drittanbieters importiert wurden

Wenn Sie nicht mehr möchten, dass eines der Ereignisse, das Sie von einem Drittanbieter importiert haben, in Ihren Kalender aufgenommen wird, können Sie die importierte iCalendar `.ics`-Datei löschen.

So löschen Sie alle Ereignisse, die aus einem Kalender eines Drittanbieters importiert wurden

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie aus der Liste der Kalender den Kalendernamen aus der Liste aus.
4. Suchen Sie den Namen des importierten Kalenders im Bereich Kalender importieren, unter Meine importierten Kalender und wählen Sie die Schaltfläche X in seiner Registerkarte.
5. Wählen Sie Save (Speichern).

Aktualisieren eines Änderungskalenders

Sie können die Beschreibung eines Änderungskalenders aktualisieren, aber nicht seinen Namen. Obwohl Sie den Standardstatus eines Kalenders ändern können, beachten Sie, dass dies das Verhalten von Änderungsaktionen bei Ereignissen, die mit dem Kalender verbunden sind, umkehrt. Wenn Sie z. B. den Status eines Kalenders von Standardmäßig geöffnet auf Standardmäßig geschlossen ändern, können unerwünschte Änderungen während der Ereigniszeiträume vorgenommen werden, wenn die Benutzer, die die verknüpften Ereignisse erstellt haben, keine Änderungen erwarten.

Wenn Sie einen Änderungskalender aktualisieren, bearbeiten Sie das Change Calendar-Dokument, das Sie beim Erstellen des Eintrags erstellt haben. Change Calendar ist eine Funktion von AWS Systems Manager.

So aktualisieren Sie einen Änderungskalender

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalenders aus, den Sie aktualisieren möchten.
4. Wählen Sie auf der Detailseite des Kalenders Actions, Edit (Aktionen, Bearbeiten) aus.
5. In Description (Beschreibung) können Sie den Beschreibungstext ändern. Sie können den Namen eines Änderungskalenders nicht bearbeiten.
6. Um den Kalenderstatus zu ändern, wählen Sie in Calendar type (Kalendertyp) einen anderen Wert. Beachten Sie, dass dies das Verhalten von Änderungsaktionen bei Ereignissen, die mit dem Kalender verbunden sind, umkehrt. Bevor Sie den Kalendertyp ändern, sollten Sie sich bei anderen Change Calendar-Benutzern vergewissern, dass die Änderung des Kalendertyps keine unerwünschten Änderungen während der von ihnen erstellten Ereignisse zulässt.

   - Open by default (Standardmäßig geöffnet) - Der Kalender ist geöffnet (Automatisierungsaktionen können bis zum Start eines Ereignisses ausgeführt werden) und dann für die Dauer eines zugehörigen Ereignisses geschlossen.
   - Closed by default (Standardmäßig geschlossen) - Der Kalender ist geschlossen (Automatisierungsaktionen können erst ab dem Beginn eines Ereignisses ausgeführt werden), aber für die Dauer eines zugehörigen Ereignisses geöffnet.
7. Wählen Sie Save (Speichern).

Ihr Kalender kann keine Aktionen verhindern oder zulassen, bis Sie mindestens ein Ereignis hinzufügen. Weitere Informationen zum Hinzufügen eines Ereignisses finden Sie unter Erstellen eines Change Calendar-Ereignisses.

Freigeben eines Änderungskalenders

Mithilfe der AWS Systems Manager Konsole können Sie einen Kalender mit einer Funktion AWS-Konten von AWS Systems Manager mit anderen teilen. Change Calendar Wenn Sie einen Kalender freigeben, ist der Kalender für Benutzer im freigegebenen Konto schreibgeschützt. Wartungsfenster, State Manager Verknüpfungen und Automatisierungen werden nicht gemeinsam genutzt.

So geben Sie einen Änderungskalender frei

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalenders aus, den Sie freigeben möchten.
4. Wählen Sie auf der Detailseite des Kalenders die Registerkarte Sharing (Freigabe) aus.
5. Wählen Sie Actions, Share (Aktionen, Freigeben) aus.
6. Geben Sie im Feld Kalender teilen unter Konto-ID die ID-Nummer eines gültigen AWS-Konto Benutzers ein und wählen Sie dann Teilen aus.

   Benutzer des freigegebenen genutzten Kontos können den Änderungskalender lesen, aber keine Änderungen vornehmen.

Einen Änderungskalender löschen

Sie können einen Kalender in Change Calendar löschen, eine Funktion von AWS Systems Manager, indem Sie entweder die Systems Manager-Konsole oder die AWS Command Line Interface(AWS CLI) verwenden. Durch das Löschen eines Änderungskalenders werden alle zugehörigen Ereignisse gelöscht.

So löschen Sie einen Änderungskalender

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalenders aus, den Sie löschen möchten.
4. Wählen Sie auf der Detailseite des Kalenders Actions, Delete (Aktionen, Löschen) aus. Wenn Sie aufgefordert werden, zu bestätigen, dass Sie den Kalender löschen möchten, wählen Sie Delete (Löschen).

Abrufen des Status eines Änderungskalenders

Sie können den Gesamtzustand eines Kalenders oder den Zustand eines Kalenders zu einem bestimmten Zeitpunkt in Change Calendar, einer Funktion in AWS Systems Manager, abrufen. Sie können auch den nächsten Zeitpunkt anzeigen, an dem der Kalenderzustand von `OPEN` auf `CLOSED` oder umgekehrt wechselt.

Diese Aufgabe können Sie nur mit der `GetCalendarState`-API-Operation ausführen. Die Prozedur in diesem Abschnitt verwendet die AWS Command Line Interface (AWS CLI).

So rufen Sie den Status eines Änderungskalenders ab

- Führen Sie den folgenden Befehl aus, um den Status eines oder mehrerer Kalender zu einer bestimmten Zeit anzuzeigen. Der Parameter `--calendar-names` ist erforderlich, `--at-time` ist jedoch optional. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

  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"
  ```

  Im Folgenden wird ein Beispiel gezeigt.

  ```
  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"
```

Im Folgenden wird ein Beispiel gezeigt.

```
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"
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "State": "OPEN",
    "AtTime": "2020-07-30T16:18:18Z",
    "NextTransitionTime": "2020-07-31T00:00:00Z"
}
```

Die Ergebnisse zeigen den Status des Kalenders (unabhängig davon, ob der Kalender vom Typ `DEFAULT_OPEN` oder `DEFAULT_CLOSED` ist) für die angegebenen Kalendereinträge, die Ihrem Konto gehören oder für dieses freigegeben sind, zu dem Zeitpunkt an, der als Wert von `--at-time` angegeben ist, sowie den Zeitpunkt des nächsten Übergangs. Wenn Sie den Parameter `--at-time` nicht hinzufügen, wird die aktuelle Zeit verwendet.

> **Note**
>
> Wenn Sie mehr als einen Kalender in einer Anforderung angeben, gibt der Befehl den Status von `OPEN` nur, wenn alle Kalender in der Anforderung geöffnet sind. Wenn ein

oder mehrere Kalender in der Anforderung geschlossen sind, lautet der zurückgegebene Status `CLOSED`.

Hinzufügen von Change Calendar-Abhängigkeiten zu Automation-Runbooks

Um Automation-Aktionen von Change Calendar auszuführen, eine Funktion von AWS Systems Manager, fügen Sie einen Schritt in ein Automation-Runbook ein, das die Aktion `aws:assertAwsResourceProperty` verwendet. Konfigurieren Sie die Aktion zur Ausführung von `GetCalendarState`, um zu überprüfen, ob sich ein bestimmter Kalendereintrag in dem gewünschten Zustand befindet (`OPEN` oder `CLOSED`). Das Automation-Runbook darf nur dann mit dem nächsten Schritt fortfahren, wenn der Kalenderstatus `OPEN` ist. Im Folgenden wird ein YAML-basierter Beispielausschnitt eines Automation-Runbooks gezeigt, das nicht zum nächsten Schritt `LaunchInstance` weitergehen kann, es sei denn, der Kalenderstatus entspricht `OPEN` (dem in `DesiredValues` festgelegten Status).

Im Folgenden wird ein Beispiel gezeigt.

```
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
...
```

Fehlerbehebung für Change Calendar

Verwenden Sie die folgenden Informationen bei der Behebung von Problemen mit Change Calendar, eine Funktion von AWS Systems Manager.

Themen

- Fehler ‚Calendar import failed' (Importieren des Kalenders fehlgeschlagen)

Fehler ‚Calendar import failed' (Importieren des Kalenders fehlgeschlagen)

Problem: Beim Importieren einer iCalendar (`.ics`)-Datei meldet das System, dass der Kalenderimport fehlgeschlagen ist.

- Lösung 1— Stellen Sie sicher, dass Sie eine Datei importieren, die von einem unterstützten Drittanbieter-Kalenderanbieter exportiert wurde. Unterstütze Anbieter sind:
  - Google Calendar (Exportanweisungen)
  - Microsoft Outlook (Exportanweisungen)
  - iCloud Calendar (Exportanweisungen)
- Lösung 2— Wenn der Quellkalender wiederkehrende Ereignisse enthält, stellen Sie sicher, dass keine einzelnen Ereignisse des Ereignisses abgebrochen oder gelöscht wurden. Derzeit unterstützt Change Calendar das Importieren von wiederkehrenden Ereignissen mit individuellen Stornierungen nicht. Um das Problem zu beheben, entfernen Sie das wiederkehrende Ereignis aus dem Quellkalender, exportieren Sie den Kalender erneut und importieren Sie ihn erneut in Change Calendar und fügen Sie dann das wiederkehrende Ereignis mithilfe der Change Calendar-Schnittstelle hinzu. Weitere Informationen finden Sie unter Erstellen eines Change Calendar-Ereignisses.
- Lösung 3 – Stellen Sie sicher, dass der Quellkalender mindestens ein Ereignis enthält. Uploads von `.ics`-Dateien, die keine Ereignisse enthalten, werden fehlschlagen.
- Lösungs 4 – Wenn das System meldet, dass der Import fehlgeschlagen ist, weil die `.ics`-Datei zu groß ist, stellen Sie sicher, dass Sie nur grundlegende Details zu Ihren Kalendereinträgen exportieren. Verringern Sie bei Bedarf die Länge des Zeitraums, den Sie exportieren.
- Lösung 5 – Wenn Change Calendar die Zeitzone des exportierten Kalenders nicht ermitteln kann, wenn Sie versuchen, ihn aus der Registerkarte Ereignisse zu importieren, wird möglicherweise die folgende Meldung angezeigt: „Der Import des Kalenders ist fehlgeschlagen" (Calendar import failed). Change Calendar konnte keine gültige Zeitzone finden. Sie können den Kalender aus dem Menü Bearbeiten importieren." Wählen Sie in diesem Fall die Option Actions, Edit (Aktionen,

bearbeiten) und versuchen Sie dann, die Datei von der Seite Edit calendar (Kalender bearbeiten) zu importieren.

- Lösungs 6 – Bearbeiten Sie die `.ics`-Datei nicht vor dem Import. Der Versuch, den Inhalt der Datei zu ändern, kann die Kalenderdaten beschädigen. Wenn Sie die Datei vor dem Import verändert haben, exportieren Sie den Kalender erneut aus dem Quellkalender, und führen Sie einen erneuten Upload durch.

AWS Systems Manager Maintenance Windows

Maintenance Windows, eine Funktion von AWS Systems Manager, hilft Ihnen dabei, einen Zeitplan für die Ausführung potenziell störender Aktionen auf Ihren Knoten zu definieren, z. B. das Patchen eines Betriebssystems, das Aktualisieren von Treibern oder das Installieren von Software oder Patches.

Mit Maintenance Windows können Sie Aktionen für zahlreiche andere AWS Ressourcentypen planen, z. B. Amazon Simple Storage Service (Amazon S3) -Buckets, Amazon Simple Queue Service (Amazon SQS) -Warteschlangen, AWS Key Management Service (AWS KMS) -Schlüssel und vieles mehr.

Eine vollständige Liste der unterstützten Ressourcentypen, die Sie in ein Wartungsfensterziel aufnehmen können, finden Sie unter Ressourcen, die Sie mit verwenden können AWS Resource Groups und Tag-Editor im AWS Resource Groups Benutzerhandbuch. Um mit Maintenance Windows zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Maintenance Windows aus.

> **Note**
>
> State Manager und Maintenance Windows können ähnliche Arten von Updates für Ihre verwalteten Knoten ausführen. Welche Option Sie wählen, hängt davon ab, ob Sie die System-Compliance automatisieren oder zeitkritische Aufgaben mit hoher Priorität während der von Ihnen angegebenen Zeiträume ausführen müssen.
> Weitere Informationen finden Sie unter Auswahl zwischen State Manager und Maintenance Windows.

Jedes Wartungsfenster hat einen Zeitplan, eine maximale Dauer, eine Reihe registrierter Ziele (die verwalteten Knoten oder andere AWS Ressourcen, auf die reagiert wird) und eine Reihe

registrierter Aufgaben. Sie können Tags zu Ihren Wartungsfenstern hinzufügen, wenn Sie sie erstellen oder aktualisieren. Tags sind Schlüssel, die das Identifizieren und Sortieren der Ressourcen in Ihrer Organisation ermöglichen. Sie können auch Daten angeben, vor oder nach denen ein Wartungsfenster nicht ausgeführt werden soll und Sie können die internationale Zeitzone angeben, auf der der Zeitplan des Wartungsfensters basieren soll.

Eine Beschreibung des Verhältnisses zwischen den verschiedenen zeitplanbezogenen Optionen für Wartungsfenster finden Sie unter Wartungsfenster-Optionen für Planung und aktive Zeiträume.

Weitere Informationen zum Arbeiten mit der `--schedule`-Option finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Unterstützte Aufgabentypen

Mit Wartungsfenstern können Sie vier Arten von Aufgaben ausführen:

- Befehle in Run Command, eine Funktion von Systems Manager

  Mehr über Run Command erfahren Sie unter AWS Systems Manager Run Command.
- Workflows in Automation, eine Funktion von Systems Manager

  Weitere Informationen über Automation-Workflows finden Sie unter AWS Systems Manager-Automatisierung.
- Funktionen in AWS Lambda

  Weitere Informationen über Lambda-Funktionen finden Sie unter Erste Schritte mit Lambda im AWS Lambda -Entwicklerhandbuch.
- Aufgaben in AWS Step Functions

> **Note**
>
> Aufgaben im Wartungsfenster unterstützen nur Step Functions Standard-State-Machine-Workflows. Sie unterstützen keine Express-State-Machine-Workflows. Informationen zu Workflowtypen für Zustandsmaschinen finden Sie unter Standard- und Express-Workflows im AWS Step Functions Entwicklerhandbuch.

Weitere Informationen zu Step Functions finden Sie im AWS Step Functions Developer Guide.

> **Note**
>
> Ein oder mehrere Ziele für Wartungsfenster Run Command-Typ-Aufgaben müssen angegeben werden. Je nach Aufgabe sind Ziele für andere Aufgabentypen im Wartungsfenster (Automatisierung AWS Lambda, und AWS Step Functions) optional. Weitere Informationen zur Ausführung von Aufgaben, die keine Ziele angeben, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

Das bedeutet, dass Sie Wartungsfenster verwenden können, um Aufgaben wie die folgenden an den ausgewählten Zielen auszuführen.

- Installieren oder Aktualisieren von Anwendungen.
- Anwenden von Patches.
- Installieren oder Aktualisieren von SSM Agent
- Führen Sie PowerShell Befehle und Linux-Shell-Skripts mithilfe einer Systems Manager Run Command Manager-Task aus.
- Erstellen von Amazon Machine Images (AMIs), Bootstrappen von Software und Konfigurieren von Knoten mit einer Systems-Manager-Automatisierungs-Aufgabe.
- Führen Sie AWS Lambda Funktionen aus, die zusätzliche Aktionen aufrufen, z. B. das Scannen Ihrer Knoten nach Patch-Updates.
- Führen Sie AWS Step Functions Zustandsmaschinen aus, um Aufgaben wie das Entfernen eines Knotens aus einer Elastic Load Balancing Balancing-Umgebung, das Patchen des Knotens und das anschließende Hinzufügen des Knotens wieder zur Elastic Load Balancing Balancing-Umgebung auszuführen.
- Zielknoten, die offline sind, indem Sie eine AWS Ressourcengruppe als Ziel angeben.

EventBridge Unterstützung

Diese Systems Manager Manager-Funktion wird in den EventBridge Amazon-Regeln als Ereignistyp unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager.

Inhalt

- Einrichten von Maintenance Windows

- Arbeiten mit Wartungsfenstern (Konsole)
- Systems Manager Maintenance Windows-Tutorials (AWS CLI)
- Anleitungen zu Wartungsfenstern
- Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben
- Wartungsfenster-Optionen für Planung und aktive Zeiträume
- Wartungsfenster-Tasks ohne Ziele registrieren
- Fehlerbehebung bei Wartungsfenstern

Einrichten von Maintenance Windows

Bevor Benutzer in Ihrem AWS-Konto System Wartungsfensteraufgaben mit Maintenance Windows einer Funktion von erstellen und planen können AWS Systems Manager, müssen ihnen die erforderlichen Berechtigungen erteilt werden.

Bevor Sie beginnen

Um die Aufgaben in diesem Abschnitt abzuschließen, müssen Sie eine oder beide der folgenden Ressourcen bereits eingerichtet haben:

- Einer IAM-Entität (Benutzer, Rolle oder Gruppe) zugewiesene Berechtigungen. Diese Entitäten sollten bereits über allgemeine Berechtigungen für die Arbeit mit Wartungsfenstern verfügen. Weisen Sie dazu den Benutzern oder Gruppen die IAM-Richtlinie `AmazonSSMFullAccess` oder eine andere IAM-Richtlinie zu, die einen kleineren Satz an Zugriffsberechtigungen für Systems Manager bereitstellt, der Aufgaben im Wartungsfenster abdeckt.
- (Optional) Für Wartungsfenster, die Run Command-Aufgaben ausführen, können Sie Statusbenachrichtigungen von Amazon Simple Notification Service (Amazon SNS) senden. Run Command ist eine Funktion von Systems Manager. Wenn Sie diese Option verwenden möchten, konfigurieren Sie das Amazon-SNS-Thema, bevor Sie diese Einrichtungs-Aufgaben ausführen. Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Systems Manager, einschließlich Informationen zum Erstellen einer IAM-Rolle zum Senden von SNS-Benachrichtigungen, finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

Übersicht über Einrichtungs-Aufgaben

Um die Berechtigungen zu erteilen, die Benutzer zum Registrieren von Wartungsfenstern benötigen, führt ein Administrator die folgenden Aufgaben aus. (Vollständige Anweisungen finden Sie in Konfigurieren Sie mit der Konsole Berechtigungen für Wartungsfenster)

Aufgabe 1: Erstellen Sie eine Richtlinie zur Verwendung mit der Rolle des benutzerdefinierten Wartungsfensters

Wartungsfenster-Aufgaben erfordern eine IAM-Rolle, um die Berechtigungen bereitzustellen, die für die Ausführung für die Zielressourcen erforderlich sind. Die Arten von Aufgaben, die Sie ausführen, und Ihre anderen betrieblichen Anforderungen bestimmen den Inhalt dieser Richtlinie.

Wir bieten eine Basisrichtlinie an, die Sie im Thema Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben anpassen können.

Aufgabe 2: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben

Die Richtlinie, die Sie in Aufgabe 1 erstellen, ist an die Wartungsfenster-Rolle angehängt, die Sie in Aufgabe 2 erstellen. Wenn Benutzer eine Wartungsfenster-Aufgabe registrieren, geben sie diese benutzerdefinierte Service-Rolle als Teil der Aufgabenkonfiguration an. Die Berechtigungen in dieser Rolle ermöglichen es Systems Manager, Wartungsfenster-Aufgaben in Ihrem Namen auszuführen.

> **Important**
>
> Bisher bot Ihnen die Systems Manager Manager-Konsole die Möglichkeit, die AWS verwaltete, mit dem IAM-Dienst verknüpfte Rolle auszuwählen`AWSServiceRoleForAmazonSSM`, die Sie als Wartungsrolle für Ihre Aufgaben verwenden möchten. Die Verwendung dieser Rolle und der zugehörigen Richtlinie, `AmazonSSMServiceRolePolicy`, für Wartungsfenster-Aufgaben wird nicht mehr empfohlen. Wenn Sie diese Rolle jetzt für Wartungsfenster-Aufgaben verwenden, empfehlen wir Ihnen, sie nicht mehr zu verwenden. Erstellen Sie stattdessen Ihre eigene IAM-Rolle, die die Kommunikation zwischen Systems Manager und anderen AWS-Services ermöglicht, wenn Ihre Wartungsfenster-Aufgaben ausgeführt werden.

Aufgabe 3: Benutzern, die Wartungsfenster-Aufgaben registrieren, Berechtigungen zur Verwendung der Service-Rolle erteilen

Wenn Sie Benutzern Berechtigungen für den Zugriff auf die Rolle des benutzerdefinierten Wartungsfensters erteilen, können sie sie mit ihren Wartungsfenstern verwenden. Dies gilt

zusätzlich zu den Berechtigungen, die Sie ihnen bereits erteilt haben, um mit den Systems Manager Manager-API-Befehlen für diese Maintenance Windows Funktion zu arbeiten. Diese Rolle vermittelt, dass Berechtigungen zum Ausführen einer Wartungsfenster-Aufgabe erforderlich sind. Infolgedessen kann ein Benutzer einem Wartungsfenster mithilfe Ihrer benutzerdefinierten Servicerolle keine Aufgaben zuweisen, ohne diese IAM-Berechtigungen übergeben zu können.

Aufgabe 4: (Optional) Verweigern Sie explizit Berechtigungen für Benutzer, die keine Wartungsfenster-Aufgaben registrieren dürfen

Sie können den Benutzern in Ihrem Bereich, die Sie nicht möchten AWS-Konto , die `ssm:RegisterTaskWithMaintenanceWindow` Erlaubnis verweigern, Aufgaben in Wartungsfenstern zu registrieren. Dies bietet eine zusätzliche Verhinderungsebene für Benutzer, die keine Wartungsfenster-Aufgaben registrieren sollten.

Themen

- Konfigurieren Sie mit der Konsole Berechtigungen für Wartungsfenster

Konfigurieren Sie mit der Konsole Berechtigungen für Wartungsfenster

Im Folgenden wird beschrieben, wie Sie die erforderlichen Rollen und Berechtigungen für Wartungsfenster mit der AWS Systems Manager-Konsole erstellen.

Themen

- Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben
- Aufgabe 2: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster (Konsole)
- Aufgabe 3: Konfigurieren von Berechtigungen für Benutzer, die Wartungsfenster-Aufgaben registrieren dürfen (Konsole)
- Aufgabe 4: Konfigurieren von Berechtigungen für Benutzer, die keine Aufgaben für das Wartungsfenster registrieren können

Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben

Sie können die folgende Richtlinie im JSON-Format verwenden, um die Richtlinie zu erstellen, die mit der Wartungsfenster-Rolle verwendet werden soll. Sie hängen diese Richtlinie an die Rolle an, die Sie später in Aufgabe 2: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster (Konsole) erstellen.

> **Important**
>
> Abhängig von den Aufgaben und Arten von Aufgaben, die Ihre Wartungsfenster ausführen, benötigen Sie möglicherweise nicht alle Berechtigungen in dieser Richtlinie und müssen möglicherweise zusätzliche Berechtigungen einschließen.

Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies und dann Create Policy.
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie die Standardinhalte durch folgenden Inhalt:

```
{
    "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": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
```

```
}
```

5. Ändern Sie den JSON-Inhalt nach Bedarf für die Wartungsaufgaben, die Sie in Ihrem Konto ausführen. Die Änderungen, die Sie vornehmen, beziehen sich auf Ihre geplanten Abläufe.

   Zum Beispiel:

   - Sie können Amazon-Ressourcenname (ARN) für bestimmte Funktionen und Zustandsmaschinen angeben, anstatt Platzhalter-Kennzeichner (*) zu verwenden.
   - Wenn Sie keine AWS Step Functions-Aufgaben ausführen möchten, können Sie die `states`-Berechtigungen und (ARNs) entfernen.
   - Wenn Sie keine AWS Lambda-Aufgaben ausführen möchten, können Sie die `lambda`-Berechtigungen und ARNs entfernen.
   - Wenn Sie keine Automatisierungs-Aufgaben ausführen möchten, können Sie die `ssm:GetAutomationExecution`- und `ssm:StartAutomationExecution`-Berechtigungen entfernen.
   - Fügen Sie zusätzliche Berechtigungen hinzu, die möglicherweise für die Ausführung der Aufgaben erforderlich sind. Manche Automatisierungsaktionen basieren z. B. auf AWS CloudFormation-Stacks. Aus diesem Grund sind die Berechtigungen `cloudformation:CreateStack`, `cloudformation:DescribeStacks` und `cloudformation:DeleteStack` erforderlich.

     Als weiteres Beispiel benötigt das Automation-Runbook `AWS-CopySnapshot` Berechtigungen zum Erstellen eines Amazon Elastic Block Store (Amazon EBS)-Snapshots. Daher benötigt die Servicerolle die Berechtigung `ec2:CreateSnapshot`.

     Informationen zu den Rollenberechtigungen, die von Automation-Runbooks benötigt werden, finden Sie in den Runbook-Beschreibungen in der Referenz zum AWS Systems Manager-Automation-Runbook..

6. Nachdem Sie die Richtlinienüberarbeitungen abgeschlossen haben, wählen Sie Next: Tags (Weiter: Tags).
7. (Optional) Fügen Sie ein oder mehrere Tag (Markierung)-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern, und wählen Sie dann Next: Review (Nächster Schritt: Prüfen) aus.
8. Geben Sie für Name einen Namen ein, der dies als Richtlinie identifiziert, die von der von Ihnen erstellten Maintenance Windows-Servicerolle verwendet wird. Beispiel: **`my-maintenance-window-role-policy`**.

9. Wählen Sie Create policy (Richtlinie erstellen) und notieren Sie sich den Namen, den Sie für die Richtlinie angegeben haben. Sie beziehen sich im nächsten Verfahren, Aufgabe 2: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster (Konsole), darauf.

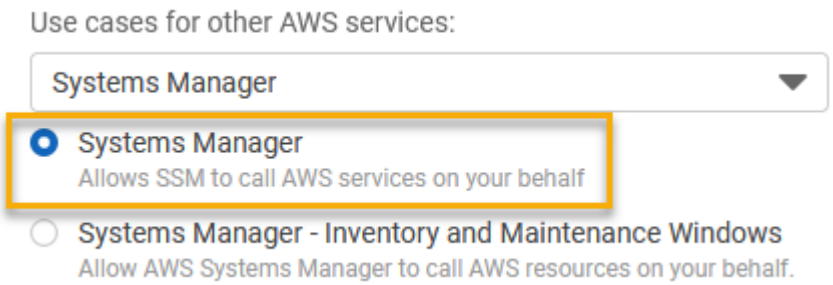
Aufgabe 2: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster (Konsole)

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte Servicerolle für Maintenance Windows zu erstellen, damit Systems Manager Maintenance Windows-Aufgaben in Ihrem Namen ausführen kann. Sie fügen die Richtlinie, die Sie in der vorherigen Aufgabe erstellt haben, an die von Ihnen erstellte benutzerdefinierte Servicerolle an.

> **Important**
>
> Zuvor bot Ihnen die Systems-Manager-Konsole die Möglichkeit, die von AWS verwaltete serviceverknüpfte IAM-Rolle `AWSServiceRoleForAmazonSSM` als Wartungsrolle für Ihre Aufgaben zu verwenden. Die Verwendung dieser Rolle und der zugehörigen Richtlinie, `AmazonSSMServiceRolePolicy`, für Wartungsfenster-Aufgaben wird nicht mehr empfohlen. Wenn Sie diese Rolle jetzt für Wartungsfenster-Aufgaben verwenden, empfehlen wir Ihnen, sie nicht mehr zu verwenden. Erstellen Sie stattdessen Ihre eigene IAM-Rolle, die die Kommunikation zwischen Systems Manager und anderen AWS-Services ermöglicht, wenn Ihre Wartungsfenster-Aufgaben ausgeführt werden.

So erstellen Sie eine benutzerdefinierte Servicerolle (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
3. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die folgenden Optionen:
    1. Wählen Sie unter Trusted entity type (Typ der vertrauenswürdigen Entität) die Option AWS-Service
    2. Für Anwendungsfälle für andere AWS-Services, wählen Sie Systems Manager
    3. Wählen Sie Systems Manager, wie im folgenden Image gezeigt.



4. Wählen Sie Next (Weiter).
5. Geben Sie im Suchfeld den Namen der Richtlinie ein, die Sie in Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben erstellt haben, aktivieren Sie das Kontrollkästchen neben dem Namen und wählen Sie dann Next (Weiter).
6. Geben Sie unter Role name (Rollenname) einen Namen ein, der diese Rolle als Maintenance Windows-Rolle identifiziert. Beispiel: **`my-maintenance-window-role`**.
7. (Optional) Ändern der Standardrollenbeschreibung, um den Zweck dieser Rolle anzuzeigen. Beispiel: **`Performs maintenance window tasks on your behalf`**.
8. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, zu verfolgen oder zu steuern und wählen Sie dann Next: Review (Weiter: Prüfen) aus.
9. Wählen Sie Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Roles (Rollen) zurück.
10. Wählen Sie den Namen der Rolle aus, die Sie gerade erstellt haben.
11. Wählen Sie die Registerkarte Trust relationships (Vertrauensstellungen) aus, und überprüfen Sie dann, ob die folgende Richtlinie im Feld Trusted entities (Vertrauenswürdige Entitäten) angezeigt wird.

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

```
}
```

12. Kopieren oder Notieren Sie sich Rollenname und ARN-Wert im Übersicht-Bereich. Benutzer in Ihrem Konto geben diese Informationen an, wenn sie Wartungsfenster erstellen.

Aufgabe 3: Konfigurieren von Berechtigungen für Benutzer, die Wartungsfenster-Aufgaben registrieren dürfen (Konsole)

Wenn Sie eine Aufgabe mit einem Wartungsfenster registrieren, geben Sie entweder eine benutzerdefinierte Servicerolle oder eine Systems Manager Service-verknüpfte Rolle für die Ausführung der eigentlichen Aufgaben-Vorgänge an. Hierbei handelt es sich um die Rolle, die vom Service angenommen wird, wenn Aufgaben in Ihrem Namen ausgeführt werden. Um die Aufgabe selbst zu registrieren, weisen Sie zuvor die IAM PassRole-Richtlinie einer IAM-Entität (z. B. einem Benutzer oder einer Gruppe) zu. Dadurch kann die IAM Entität (Benutzer oder Gruppe) als Teil der Registrierung dieser Aufgaben im Wartungsfenster die Rolle angeben, die beim Ausführen der Aufgaben verwendet werden soll. Weitere Informationen finden Sie unter Erteilen von Berechtigungen, mit denen ein Benutzer eine Rolle an einen AWS-Service übergeben kann im IAM-Benutzerhandbuch.

So konfigurieren Sie die Berechtigungen für Benutzer, die Wartungsfensteraufgaben registrieren dürfen

Wenn eine IAM-Entität (Benutzer, Rolle oder Gruppe) mit Administratorberechtigungen eingerichtet ist, hat der Benutzer oder die Rolle Zugriff auf Wartungsfenster. Für Entitäten ohne Administratorberechtigungen muss ein Administrator der IAM-Entität die folgenden Berechtigungen gewähren. Dies sind die Mindestberechtigungen, die erforderlich sind, um Aufgaben in einem Wartungsfenster zu registrieren:

- Die von `AmazonSSMFullAccess` verwaltete Richtlinie oder eine Richtlinie, die vergleichbare Berechtigungen bereitstellt.
- Die folgenden `iam:PassRole`- und `iam:ListRoles`-Berechtigungen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::account-id:role/my-maintenance-window-role"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::account-id:role/"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::account-id:role/aws-service-role/
ssm.amazonaws.com/"
        }
    ]
}
```

my-maintenance-window-role repräsentiert den Namen der benutzerdefinierten Wartungsfensterrolle, die Sie zuvor erstellt haben.

account-id repräsentiert die ID Ihres AWS-Konto. Durch das Hinzufügen dieser Berechtigung für die Ressource `arn:aws:iam::account-id:role/` können Benutzer Kundenrollen in der Konsole anzeigen und auswählen, wenn sie eine Wartungsfensteraufgabe erstellen. Durch das Hinzufügen dieser Berechtigung für `arn:aws:iam::account-id:role/aws-service-role/ssm.amazonaws.com/` können Benutzer die mit dem Systems Manager-Service verknüpfte Rolle in der Konsole auswählen, wenn sie eine Wartungsfensteraufgabe erstellen.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center-Benutzerhandbuch.

- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter Erstellen einer Rolle für einen externen Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch.

- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter Erstellen einer Rolle für einen IAM-Benutzer im IAM-Benutzerhandbuch.

- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Konfigurieren von Berechtigungen für Gruppen, die Wartungsfensteraufgaben registrieren dürfen (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich auf Benutzergruppen.
3. Wählen Sie in der Liste der Gruppen den Namen der Gruppe aus, der Sie die `iam:PassRole`-Berechtigung zuweisen möchten.
4. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Optionen Add permissions, Create Inline Policy (Berechtigungen hinzufügen, Inline-Richtlinie erstellen) und anschließend die Registerkarte JSON aus.
5. Ersetzen Sie den Standardinhalt des Felds durch den folgenden Inhalt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::account-id:role/my-maintenance-window-role"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::account-id:role/"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::account-id:role/aws-service-role/
ssm.amazonaws.com/"
        }
    ]
}
```

my-maintenance-window-role repräsentiert den Namen der benutzerdefinierten Wartungsfensterrolle, die Sie zuvor erstellt haben.

account-id repräsentiert die ID Ihres AWS-Konto. Durch das Hinzufügen dieser Berechtigung für die Ressource `arn:aws:iam::`*account-id*`:role/` können Benutzer Kundenrollen in der Konsole anzeigen und auswählen, wenn sie eine Wartungsfensteraufgabe erstellen. Durch das Hinzufügen dieser Berechtigung für `arn:aws:iam::`*account-id*`:role/aws-service-role/ssm.amazonaws.com/` können Benutzer die mit dem Systems Manager-Service verknüpfte Rolle in der Konsole auswählen, wenn sie eine Wartungsfensteraufgabe erstellen.

6. Wählen Sie Review policy (Richtlinie prüfen).
7. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) einen Namen in das Feld Name ein, um diese `PassRole`-Richtlinie zu identifizieren (beispielsweise **`my-group-iam-passrole-policy`**) und wählen Sie dann Create Policy (Richtlinie erstellen) aus.

Aufgabe 4: Konfigurieren von Berechtigungen für Benutzer, die keine Aufgaben für das Wartungsfenster registrieren können

Je nachdem, ob Sie die `ssm:RegisterTaskWithMaintenanceWindow`-Berechtigung für einen einzelnen Benutzer oder eine Gruppe verweigern, verwenden Sie eines der folgenden Verfahren, um zu verhindern, dass Benutzer Aufgaben mit einem Wartungsfenster registrieren können.

So konfigurieren Sie Berechtigungen für Benutzer, die keine Wartungsfensteraufgaben registrieren dürfen

- Ein Administrator muss der IAM-Entität die folgenden Einschränkungen hinzufügen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:RegisterTaskWithMaintenanceWindow",
            "Resource": "*"
        }
    ]
}
```

Konfigurieren von Berechtigungen für Gruppen, die Wartungsfensteraufgaben registrieren dürfen (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich auf Benutzergruppen.
3. Wählen Sie in der Liste der Gruppen den Namen der Gruppe aus, der Sie die `ssm:RegisterTaskWithMaintenanceWindow`-Berechtigung verweigern möchten.
4. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Optionen Add permissions, Create Inline Policy (Berechtigungen hinzufügen, Inline-Richtlinie erstellen) aus.
5. Wählen Sie die Registerkarte JSON und ersetzen Sie den Standardinhalt des Feldes durch den folgenden Text.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:RegisterTaskWithMaintenanceWindow",
            "Resource": "*"
        }
    ]
}
```

6. Wählen Sie Review policy (Richtlinie prüfen).
7. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) bei Name einen Namen ein, um diese Richtlinie zu identifizieren (beispielsweise **`my-groups-deny-mw-tasks-policy`**) und wählen Sie dann Create Policy (Richtlinie erstellen) aus.

Arbeiten mit Wartungsfenstern (Konsole)

In diesem Abschnitt wird beschrieben, wie Sie Wartungsfenster mithilfe der AWS Systems Manager-Konsole erstellen, konfigurieren, aktualisieren und löschen. Dieser Abschnitt enthält auch Informationen zum Verwalten der Ziele und Aufgaben eines Wartungsfensters.

> **Important**
>
> Wir empfehlen, dass Sie Wartungsfenster anfänglich in einer Testumgebung erstellen und konfigurieren.

Bevor Sie beginnen

Bevor Sie ein Wartungsfenster erstellen, müssen Sie den Zugriff auf Maintenance Windows, eine Funktion von AWS Systems Manager, konfigurieren. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

Themen

- Erstellen eines Wartungsfensters (Konsole)
- Zuweisen von Zielen zu einem Wartungsfenster (Konsole)
- Zuweisen von Aufgaben zu einem Wartungsfenster (Konsole)
- Ein Wartungsfenster deaktivieren oder aktivieren
- Aktualisieren oder Löschen von Wartungsfenster-Ressourcen (Konsole)

Erstellen eines Wartungsfensters (Konsole)

In diesem Verfahren erstellen Sie ein Wartungsfenster in Maintenance Windows, eine-Funktion von AWS Systems Manager. Sie können die grundlegenden Optionen, wie Name, Zeitplan und Dauer, festlegen. In späteren Schritten wählen Sie die Ziele oder Ressourcen aus, die damit aktualisiert werden sollen, sowie die Aufgaben, die während der Ausführung des Wartungsfensters ausgeführt werden.

> **Note**
>
> Eine Beschreibung des Verhältnisses zwischen den verschiedenen zeitplanbezogenen Optionen für Wartungsfenster finden Sie unter Wartungsfenster-Optionen für Planung und aktive Zeiträume.
> Weitere Informationen zum Arbeiten mit der `--schedule`-Option finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

So erstellen Sie ein Wartungsfenster (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus.
4. Geben Sie im Feld Name einen aussagekräftigen Namen ein, an dem Sie dieses Wartungsfenster erkennen können.
5. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein, um anzugeben, wie dieses Wartungsfenster verwendet werden soll.
6. Wenn eine Wartungsfenster-Aufgabe auf verwalteten Knoten ausgeführt werden soll, obwohl diese Knoten nicht als Ziele registriert wurden, wählen Sie Allow unregistered targets (Nicht registrierte Ziele erlauben) aus.

   Falls Sie diese Option wählen, können Sie die nicht registrierten Knoten (nach Knoten-ID) auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.

   Sollten Sie diese Option nicht wählen, müssen Sie die zuvor registrierten Ziele auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.
7. Geben Sie mithilfe einer der drei Planungsoptionen einen Zeitplan für das Wartungsfenster an.

   Weitere Informationen zum Erstellen von CRON-/Rate-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.
8. Geben Sie unter Duration (Dauer) die Anzahl der Stunden ein, die das Wartungsfenster ausgeführt wird. Der Wert, den Sie angeben, bestimmt die spezifische Endzeit für das Wartungsfenster basierend auf dem Zeitpunkt, an dem es beginnt. Nach der resultierenden Endzeit dürfen keine Wartungsfenster-Aufgaben gestartet werden, abzüglich der Anzahl der Stunden, die Sie für Stop initiating tasks (Initiieren von Aufgaben beenden) im nächsten Schritt angeben.

   Beispiel: Wenn das Wartungsfenster um 15:00 Uhr beginnt, die Dauer drei Stunden beträgt und der Wert Stop initiating tasks (Initiieren von Aufgaben beenden) eine Stunde beträgt, können nach 17:00 Uhr keine Wartungsfenster-Aufgaben gestartet werden.
9. Geben Sie unter Stop initiating tasks (Initiieren von Aufgaben beenden) die Anzahl der Stunden für den Zeitpunkt vor dem Ende des Wartungsfensters an, ab dem vom System keine neuen auszuführenden Aufgaben mehr geplant werden sollen.

10. (Optional) Geben Sie unter Window start date (Startzeit des Fensters) ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an, zu dem bzw. der das Wartungsfenster aktiviert werden soll. Auf diese Weise können Sie die Aktivierung des Wartungsfensters bis zum angegebenen künftigen Zeitpunkt verzögern.

    > **Note**
    >
    > Sie können kein Startdatum und keine Startzeit angeben, die in der Vergangenheit liegen.

11. (Optional) Geben Sie unter Window end date (Enddatum des Fensters) ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an, zu dem bzw. der das Wartungsfenster deaktiviert werden soll. Auf diese Weise können Sie ein in der Zukunft liegendes Datum sowie eine Uhrzeit festlegen, nach dem das Wartungsfenster nicht mehr ausgeführt wird.
12. (Optional) Geben Sie unter Schedule time zone (Zeitzone des Zeitplans) die Zeitzone im IANA-Format (Internet Assigned Numbers Authority) an, die als Grundlage für die Ausführung der geplanten Wartungsfenster verwendet werden soll. Zum Beispiel: "America/Los_Angeles", "etc/UTC", oder "Asia/Seoul".

    Weitere Informationen zu gültigen Formaten finden Sie unter Time Zone Database (Zeitzonendatenbank) auf der IANA-Website.
13. (Optional) Geben Sie unter Schedule offset (Zeitplanversatz) die Anzahl der Tage an, die nach dem durch einen Cron- oder Rate-Ausdruck angegebenen Datum und der angegebenen Uhrzeit gewartet werden soll, bevor das Wartungsfenster ausgeführt wird. Sie können ein bis sechs Tage angeben.

    > **Note**
    >
    > Diese Option ist nur verfügbar, wenn Sie einen Zeitplan durch manuelle Eingabe eines Cron- oder Rate-Ausdrucks angegeben haben.

14. (Optional) Weisen Sie im Abschnitt Manage tags (Tags verwalten) dem Wartungsfenster ein oder mehrere Tag-Schlüsselname-Wert-Paare zu.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise ein Wartungsfenster mit Tags versehen, um die Aufgabentypen, die darüber ausgeführt werden, die Arten der Ziele sowie die Umgebung,

in der es ausgeführt wird, zu identifizieren. In diesem Fall könnten Sie z.B. die folgenden Schlüsselname-Wert-Paare angeben:

- `Key=TaskType,Value=AgentUpdate`
- `Key=OS,Value=Windows`
- `Key=Environment,Value=Production`

15. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus. Das System leitet Sie zur Seite „Maintenance Window“ (Wartungsfenster) zurück. Der Status des soeben erstellten Wartungsfensters lautet Enabled (Aktiviert).

Zuweisen von Zielen zu einem Wartungsfenster (Konsole)

In diesem Verfahren registrieren Sie ein Ziel für ein Wartungsfenster. Mit anderen Worten: Geben Sie an, für welche Ressourcen das Wartungsfenster Aktionen durchführt.

> **Note**
>
> Wenn eine einzelne Wartungsfenster-Aufgabe mit mehreren Zielen registriert ist, werden ihre Aufrufe sequenziell und nicht parallel ausgeführt. Wenn Ihre Aufgabe gleichzeitig auf mehreren Zielen ausgeführt werden muss, registrieren Sie eine Aufgabe für jedes Ziel einzeln, und weisen Sie jeder Aufgabe dieselbe Prioritätsstufe zu.

So weisen Sie einem Wartungsfenster Ziele zu (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie in der Wartungsfensterliste die Wartungsfenster aus, dem Ziele hinzugefügt werden sollen.
4. Wählen Sie Actions (Aktionen) und anschließend Register targets (Ziele registrieren) aus.
5. (Optional) Geben Sie im Feld Target Name (Zielname) einen Namen für die Ziele ein.
6. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein.
7. (Optional) Geben Sie für Eigentümerinformationen Informationen an, die in jedes EventBridge Amazon-Ereignis aufgenommen werden sollen, das während der Ausführung von Aufgaben für diese Ziele in diesem Wartungsfenster ausgelöst wird.

Informationen EventBridge zur Überwachung von Systems Manager Manager-Ereignissen finden Sie unter[Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge](#).

8. Wählen Sie im Bereich Targets (Ziele) eine der in der folgenden Tabelle beschriebenen Optionen.

Option	Beschreibung
Specify instance tags (Instance-Tags angeben)	Geben Sie unter Specify instance tags (Instance-Tags angeben) einen oder mehrere Tag-Schlüssel und (optional) Werte an, die den verwalteten Knoten in Ihrem Konto hinzugefügt wurden oder werden. Wenn das Wartungsfenster ausgeführt wird, versucht das Programm, Aufgaben auf allen verwaltet en Knoten auszuführen, denen diese Tags hinzugefügt wurden. Wenn Sie mehr als einen Tag-Schlüssel angeben, muss ein Knoten mit allen Tag-Schlüsseln und -Werten markiert werden, die Sie für die Aufnahme in die Zielgruppe angeben.
Choose instances manually (Instances manuell auswählen)	Aktivieren Sie in der Liste das Kontrollk ästchen für jeden Knoten, den Sie für das Wartungsfenster-Ziel aufnehmen möchten. Die Liste enthält alle Knoten in Ihrem Konto, die für die Verwendung mit Systems Manager konfiguriert sind. Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembe handlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

Option	Beschreibung
	Weitere Informationen zu Edge-Geräten und On-Premises-Servern und virtuelle n Maschinen (VMs) finden Sie unter Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen

Option	Beschreibung
Eine Ressourcengruppe auswählen	Wählen Sie für Resource group (Ressourcengruppe) den Namen einer vorhandenen Ressourcengruppe in Ihrem Konto aus der Liste aus. Weitere Informationen zum Erstellen von und Arbeiten mit Ressourcengruppen finden Sie unter den folgenden Themen: • Was sind Ressourcengruppen? im AWS Resource Groups -Benutzerhandbuch • Ressourcengruppen und Tagging für AWS im AWS News Blog (Optional) Wählen Sie unter Resource types (Ressourcentypen) bis zu fünf verfügbare Ressourcentypen aus oder wählen Sie All resource types (Alle Ressourcentypen) aus. Wenn die Aufgaben, die Sie dem Wartungsfenster zugeordnet haben, für einen der dem Ziel hinzugefügten Ressourcentypen nicht geeignet sind, meldet das System möglicherweise einen Fehler. Auch wenn ein solcher Fehler gemeldet wird, werden Aufgaben, für die ein unterstützter Ressourcentyp gefunden wurde, dennoch ausgeführt. Nehmen Sie beispielsweise an, Sie fügen diesem Ziel die folgenden Ressourcentypen hinzu: • `AWS::S3::Bucket` • `AWS::DynamoDB::Table` • `AWS::EC2::Instance`

Option	Beschreibung
	Wenn Sie dem Wartungsfenster später Aufgaben hinzufügen, nehmen Sie nur Aufgaben auf, die Aktionen für Knoten durchführen, wie z. B. das Anwenden einer Patch-Baseline oder das Neustarte n eines Knotens. Möglicherweise wird im Protokoll Wartungsfensterprotokoll ein Fehler gemeldet, dass keine Amazon Simple Storage Service (Amazon S3)-Buckets oder Amazon DynamoDB-Tabellen gefunden wurden. Das Wartungsfenster führt jedoch weiterhin Aufgaben auf den Knoten in Ihrer Ressourcengruppe aus.

9. Wählen Sie Register target.

Wenn Sie diesem Wartungsfenster mehrere Ziele zuweisen möchten, wählen Sie die Registerkarte Targets (Ziele) und anschließend Register target (Ziel registrieren) aus. Mit dieser Option können Sie eine andere Auswahlmethode festlegen. Wenn Sie beispielsweise zuvor Ziel-Knoten nach Knoten-ID ausgewählt haben, können Sie neue Ziele und Ziel-Knoten registrieren, indem Sie für verwaltete Knoten Tags angeben oder Ressourcentypen aus einer Ressourcengruppe auswählen.

Zuweisen von Aufgaben zu einem Wartungsfenster (Konsole)

In diesem Verfahren fügen Sie eine Aufgabe zu einem Wartungsfenster hinzu. Aufgaben sind die Aktionen, die während der Ausführung eines Wartungsfensters durchgeführt werden.

Die folgenden vier Aufgabentypen können zu einem Wartungsfenster hinzugefügt werden:

- AWS Systems Manager Run Command-Befehle
- Systems Manager Automation-Workflows
- AWS Step Functions Aufgaben
- AWS Lambda Funktionen

> **Important**
>
> Die IAM-Richtlinie für Maintenance Windows erfordert, dass Sie den Namen von Lambda-Funktionen (oder Aliasen) das Präfix `SSM` hinzufügen. Bevor Sie mit der Registrierung dieser Art von Aufgabe fortfahren, aktualisieren Sie ihren Namen so, dass er AWS Lambda einschließt`SSM`. Beispiel: Wenn Ihr Lambda-Funktionsname `MyLambdaFunction` lautet, ändern Sie ihn in `SSMMyLambdaFunction`.

So weisen Sie einem Wartungsfenster Aufgaben zu

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie ein Wartungsfenster aus der Wartungsfensterliste aus.
4. Wählen Sie Actions (Aktionen) und anschließend die Option für den Aufgabentyp aus, den Sie für das Wartungsfenster registrieren möchten.

   - Register Run command task (Run Command-Aufgabe registrieren)
   - Register Automation task (Automatisierungsaufgabe registrieren)
   - Register Lambda task (Lambda-Aufgabe registrieren)
   - Register Step Functions task (Step Functions-Aufgabe registrieren)

   > **Note**
   >
   > Aufgaben im Wartungsfenster unterstützen nur Step Functions Standard-State-Machine-Workflows. Sie unterstützen keine Express-State-Machine-Workflows. Informationen zu Workflowtypen für Zustandsmaschinen finden Sie unter Standard- und Express-Workflows im AWS Step Functions Entwicklerhandbuch.

5. (Optional) Geben Sie unter Name einen Namen für die Aufgabe ein.
6. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein.
7. Wählen Sie für New task invocation cutoff (Cutoff für den Aufruf neuer Aufgaben) Enabled (Aktiviert), wenn Sie nicht möchten, dass neue Aufgabenaufrufe nach Erreichen der Grenzzeit des Wartungsfensters gestartet werden.

Wenn diese Option nicht aktiviert ist, wird die Aufgabe weiter ausgeführt, wenn die Grenzzeit erreicht ist, und startet neue Aufgabenaufrufe bis zum Abschluss.

> **Note**
>
> Der Status für Aufgaben, die beim Aktivieren dieser Option nicht abgeschlossen sind, lautet `TIMED_OUT`.

8. Folgen Sie für diesen Schritt den Unterschritten für den ausgewählten Aufgabentyp.

   Run Command

   1. Wählen Sie in der Liste Befehlsdokument das Systems Manager Manager-Befehlsdokument (SSM-Dokument) aus, das die auszuführenden Aufgaben definiert.
   2. Wählen Sie für Document version (Dokumentversion) die zu verwendende Dokumentversion aus.
   3. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.

   Automation

   1. Wählen Sie in der Liste der Automatisierungsdokumente das Automatisierungs-Runbook aus, das die auszuführenden Aufgaben definiert.
   2. Wählen Sie für Document version (Dokumentversion) die zu verwendende Runbook-Version aus.
   3. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.

   Lambda

   1. Wählen Sie im Bereich Lambda-Parameter eine Lambda-Funktion aus der Liste aus.
   2. (Optional) Geben Sie einzubeziehende Inhalte für Payload (Nutzlast), Client Context (Client-Kontext) oder Qualifier (Qualifizierer) an.

> **Note**
>
> In einigen Fällen können Sie einen Pseudo-Parameter als Teil Ihres Werts verwenden. `Payload` Wenn dann die Wartungsfensteraufgabe ausgeführt wird, werden anstelle der Platzhalter für Pseudo-Parameter die richtigen Werte übergeben. Weitere Informationen finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben.

3. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.

Step Functions

1. Wählen Sie im Parameterbereich Step Functions eine Zustandsmaschine aus der Liste aus.
2. (Optional) Geben Sie einen Namen für die Ausführung des Zustandsautomaten und einzubeziehende Inhalte für Input (Eingabe) an.

> **Note**
>
> In einigen Fällen können Sie einen Pseudo-Parameter als Teil Ihres `Input` Werts verwenden. Wenn dann die Wartungsfensteraufgabe ausgeführt wird, werden anstelle der Platzhalter für Pseudo-Parameter die richtigen Werte übergeben. Weitere Informationen finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben.

3. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.

9. Wählen Sie im Bereich Targets (Ziele) eine der folgenden Optionen aus:

- Auswahl registrierter Zielgruppen: Wählen Sie ein oder mehrere Wartungsfensterziele aus, die Sie im aktuellen Wartungsfenster registriert haben.
- Auswählen von nicht registrierten Zielen: Wählen Sie nacheinander verfügbare Ressourcen als Ziele für den Vorgang aus.

Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

- Aufgabenziel nicht erforderlich: Ziele für die Aufgabe werden möglicherweise bereits in anderen Funktionen für alle Run Command-Typ-Aufgaben angegeben.

  Geben Sie ein oder mehrere Ziele für Wartungsfenster Run Command-Typ-Aufgaben an. Je nach Aufgabe sind Ziele für andere Aufgabentypen im Wartungsfenster (Automatisierung AWS Lambda, und AWS Step Functions) optional. Weitere Informationen zur Ausführung von Aufgaben, die keine Ziele angeben, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

  > **Note**
  >
  > In vielen Fällen müssen Sie ein Ziel für eine Automatisierungsaufgabe nicht explizit angeben. Angenommen, Sie erstellen beispielsweise eine Automation-Aufgabe, um eine Amazon Machine Image (AMI) für Linux mit dem `AWS-UpdateLinuxAmi`-Runbook zu aktualisieren. Wenn die Aufgabe ausgeführt wird, wird AMI mit den neuesten verfügbaren Linux-Verteilungspaketen und Amazon-Software aktualisiert. Neue Instances, die aus der AMI erstellt wurden, haben diese Updates bereits installiert. Da die ID des AMI in den Eingabeparametern für das Runbook angegeben ist, muss in der Wartungsfenster-Aufgabe kein Ziel erneut angegeben werden.

10. Nur Automatisierungsaufgaben:

    Geben Sie im Bereich Input parameters (Eingabeparameter) Werte für erforderliche oder optionale Parameter an, die zum Ausführen der Aufgabe notwendig sind.

    > **Note**
    >
    > In einigen Fällen können Sie einen Pseudo-Parameter für bestimmte Eingabeparameterwerte verwenden. Wenn dann die Wartungsfensteraufgabe ausgeführt wird, werden anstelle der Platzhalter für Pseudo-Parameter die richtigen Werte übergeben. Weitere Informationen finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben.

11. Für Rate control (Ratenregelung):

- Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

  > **Note**
  >
  > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

12. (Optional) Wählen Sie für die IAM-Servicerolle eine Rolle aus, die Systems Manager bei der Ausführung einer Wartungsfensteraufgabe übernehmen soll.

    Wenn Sie keinen ARN für eine Servicerolle angeben, verwendet Systems Manager eine dienstverknüpfte Rolle in Ihrem Konto. Wenn in Ihrem Konto keine geeignete serviceverknüpfte Rolle für Systems Manager vorhanden ist, wird sie erstellt, wenn die Aufgabe erfolgreich registriert wurde.

    > **Note**
    >
    > Um die Sicherheit zu verbessern, empfehlen wir dringend, eine benutzerdefinierte Richtlinie und eine benutzerdefinierte Servicerolle für die Ausführung Ihrer Aufgaben im Wartungsfenster zu erstellen. Die Richtlinie kann so gestaltet werden, dass sie nur die Berechtigungen gewährt, die für Ihre speziellen Wartungsfensteraufgaben erforderlich sind. Weitere Informationen finden Sie unter Konfigurieren Sie mit der Konsole Berechtigungen für Wartungsfenster.

13. Run Commandnur Aufgaben:

    (Optional) Gehen Sie für Output options (Ausgabeoptionen) wie folgt vor:

- Aktivieren Sie das Kontrollkästchen Enable writing to S3 (Schreiben in S3 aktivieren), um die Befehlsausgabe in einer Datei zu speichern. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.
- Aktivieren Sie das Kontrollkästchen CloudWatch Ausgabe, um die vollständige Ausgabe in Amazon CloudWatch Logs zu schreiben. Geben Sie den Namen einer CloudWatch Logs-Protokollgruppe ein.

> **Note**
>
> Die Berechtigungen, die das Schreiben von Daten in einen S3-Bucket oder in CloudWatch Logs ermöglichen, entsprechen denen des Instanzprofils, das dem Knoten zugewiesen ist, und nicht denen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen. Wenn sich der angegebene S3-Bucket oder die angegebene Protokollgruppe in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das dem Knoten zugeordnete Instanzprofil über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

14. Run Commandnur Aufgaben:

    Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

15. Run CommandNur Aufgaben:

    Geben Sie im Abschnitt Parameters (Parameter) die Parameter für das Dokument an.

> **Note**
>
> In einigen Fällen können Sie einen Pseudo-Parameter für bestimmte Eingabeparameterwerte verwenden. Wenn dann die Wartungsfensteraufgabe ausgeführt wird, werden anstelle der Platzhalter für Pseudo-Parameter die richtigen Werte

übergeben. Weitere Informationen finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben.

16. Run Commandund nur Automatisierungsaufgaben:

    (Optional) Wählen Sie im CloudWatch Alarmbereich unter Alarmname einen vorhandenen CloudWatch Alarm aus, der auf Ihre zu überwachende Aufgabe angewendet werden soll.

    Wenn der Alarm aktiviert wird, wird die Aufgabe gestoppt.

    > **Note**
    >
    > Um Ihrer Aufgabe einen CloudWatch Alarm zuzuweisen, muss der IAM-Principal, der die Aufgabe ausführt, über die entsprechenden Berechtigungen verfügen`iam:createServiceLinkedRole`. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden.

17. Wählen Sie je nach Aufgabentyp eine der folgenden Optionen aus:

    - Register Run command task (Run Command-Aufgabe registrieren)
    - Register Automation task (Automatisierungsaufgabe registrieren)
    - Register Lambda task (Lambda-Aufgabe registrieren)
    - Register Step Functions task (Step Functions-Aufgabe registrieren)

Ein Wartungsfenster deaktivieren oder aktivieren

Sie können ein Wartungsfenster in Maintenance Windows deaktivieren oder aktivieren, eine Funktion von AWS Systems Manager. Sie können jeweils ein Wartungsfenster auswählen, um die Ausführung des Wartungsfensters entweder zu deaktivieren oder zu aktivieren. Sie können auch mehrere oder alle Wartungsfenster zum Aktivieren und Deaktivieren auswählen.

In diesem Abschnitt wird beschrieben, wie Sie ein Wartungsfenster mit Hilfe der Systems-Manager-Konsole deaktivieren oder aktivieren können. Beispiele dafür, wie Sie dies mithilfe von AWS Command Line Interface (AWS CLI) tun können, finden Sie unterTutorial: Aktualisieren eines Wartungsfensters (AWS CLI).

Themen

- Dektivieren eines Wartungsfensters (Konsole)

- Aktivieren eines Wartungsfensters (Konsole)

Dektivieren eines Wartungsfensters (Konsole)

Sie können ein Wartungsfenster deaktivieren, um eine Aufgabe für einen bestimmten Zeitraum anzuhalten, so dass sie später wieder aktiviert werden kann.

Um ein Wartungsfenster zu deaktivieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie mit Hilfe des Kontrollkästchens neben dem Wartungsfenster, das Sie deaktivieren möchten, ein oder mehrere Wartungsfenster aus.
4. Wählen Sie Wartungsfenster deaktivieren im Menü Aktionen. Sie werden aufgefordert, Ihre Aktionen zu bestätigen.

Aktivieren eines Wartungsfensters (Konsole)

Sie können ein Wartungsfenster aktivieren, um eine Aufgabe wieder aufzunehmen.

> **Note**
>
> Wenn für das Wartungsfenster ein Tarif verwendet wird und das Startdatum derzeit auf ein Datum und eine vergangene Uhrzeit festgelegt ist, werden das aktuelle Datum und die aktuelle Uhrzeit als Startdatum für das Wartungsfenster verwendet. Sie können das Startdatum des Wartungsfensters vor oder nach der Aktivierung ändern. Weitere Informationen finden Sie unter Aktualisieren oder Löschen von Wartungsfenster-Ressourcen (Konsole).

Ein Wartungsfenster aktivieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Markieren Sie das Kontrollkästchen neben dem Wartungsfenster, um es zu aktivieren.

4. Wählen Sie Aktionen, Wartungsfenster aktivieren. Sie werden aufgefordert, Ihre Aktionen zu bestätigen.

Aktualisieren oder Löschen von Wartungsfenster-Ressourcen (Konsole)

Sie können ein Wartungsfenster in Maintenance Windows, eine Funktion von AWS Systems Manager, aktualisieren oder löschen. Sie können auch die Ziele oder Aufgaben eines Wartungsfensters aktualisieren oder löschen. Wenn Sie die Details eines Wartungsfensters bearbeiten, können Sie den Zeitplan, die Ziele und die Aufgaben ändern. Sie können auch Namen und Beschreibungen für Fenster, Ziele und Aufgaben angeben. Auf diese Weise erhalten Sie einen besseren Eindruck des Zwecks und vereinfachen die Verwaltung Ihrer Fensterwarteschlange.

In diesem Abschnitt wird beschrieben, wie Sie ein Wartungsfenster, Ziele und Aufgaben über die Systems Manager-Konsole aktualisieren oder löschen. Beispiele dafür, wie dies mit der AWS Command Line Interface (AWS CLI) möglich ist, finden Sie unter Tutorial: Aktualisieren eines Wartungsfensters (AWS CLI).

Themen

- Aktualisieren oder Löschen eines Wartungsfensters (Konsole)
- Aktualisieren oder Abmelden von Wartungsfenster-Zielen (Konsole)
- Aktualisieren oder Abmelden von Wartungsfenster-Aufgaben (Konsole)

Aktualisieren oder Löschen eines Wartungsfensters (Konsole)

Sie können ein Wartungsfenster aktualisieren, um den Namen, die Beschreibung und den Zeitplan des Wartungsfensters zu ändern und festzulegen, ob das Wartungsfenster nicht registrierte Ziele zulassen soll.

So aktualisieren oder löschen Sie ein Wartungsfenster

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie die Schaltfläche neben dem Wartungsfenster aus, das Sie aktualisieren oder löschen möchten, und führen Sie dann einen der folgenden Schritte aus:

   - Wählen Sie Delete (Löschen). Sie werden aufgefordert, Ihre Aktionen zu bestätigen.

- Wählen Sie Edit (Bearbeiten) aus. Ändern Sie auf der Seite Edit maintenance window (Wartungsfenster bearbeiten) die Werte und Optionen nach Bedarf und wählen Sie dann Save changes (Änderungen speichern) aus.

  Weitere Informationen zu den Konfigurationsoptionen, die Sie ausführen können, finden Sie unter Erstellen eines Wartungsfensters (Konsole).

Aktualisieren oder Abmelden von Wartungsfenster-Zielen (Konsole)

Sie können die Ziele eines Wartungsfensters aktualisieren oder abmelden. Wenn Sie das Ziel eines Wartungsfensters aktualisieren möchten, können Sie einen neuen Zielnamen, eine Beschreibung und einen Eigentümer angeben. Sie können auch verschiedene Ziele auswählen.

So aktualisieren oder löschen Sie die Ziele eines Wartungsfensters

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie den Namen des zu aktualisierenden Wartungsfensters und danach die Registerkarte Targets (Ziele) aus und führen Sie dann einen der folgenden Schritte aus:

   - Um Ziele zu aktualisieren, klicken Sie auf die Schaltfläche neben dem zu aktualisierenden Ziel und wählen Sie dann Edit (Bearbeiten) aus.
   - Um Ziele abzumelden, klicken Sie auf die Schaltfläche neben dem abzumeldenden Ziel und wählen Sie dann Deregister target (Ziel abmelden) aus. Wählen Sie im Dialogfenster Deregister maintenance windows target (Deregistrierung des Wartungsfensterziels) die Option Deregistrierung.

Aktualisieren oder Abmelden von Wartungsfenster-Aufgaben (Konsole)

Sie können die Aufgaben eines Wartungsfensters aktualisieren oder abmelden. Wenn Sie eine Aktualisierung durchführen möchten, können Sie einen neuen Aufgabennamen, eine Beschreibung und einen Eigentümer angeben. Bei Run Command und Automation-Aufgaben können Sie ein anderes SSM-Dokument für die Aufgaben wählen. Sie können allerdings den Typ einer Aufgabe nicht durch Bearbeitung ändern. Beispiel: Wenn Sie eine Automatisierungsaufgabe erstellt haben, können Sie die Aufgabe nicht bearbeiten und in eine Run Command-Aufgabe ändern.

So aktualisieren oder löschen Sie die Aufgaben eines Wartungsfensters (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie den Namen des zu aktualisierenden Wartungsfensters aus.
4. Wählen Sie die Registerkarte Tasks (Aufgaben) aus und klicken Sie dann auf die Schaltfläche neben der zu aktualisierenden Aufgabe.
5. Führen Sie eine der folgenden Aktionen aus:
   - Um eine Aufgabe abzumelden, wählen Sie Deregister task (Aufgabe abmelden) aus.
   - Um die Aufgabe zu bearbeiten, wählen Sie Edit (Bearbeiten) aus. Ändern Sie die Werte und Optionen wie gewünscht und wählen Sie dann Edit Task (Aufgabe bearbeiten) aus.

Systems Manager Maintenance Windows-Tutorials (AWS CLI)

Dieser Abschnitt enthält Tutorials, in denen Sie lernen, wie Sie das AWS Command Line Interface (AWS CLI) für folgende Zwecke verwenden können:

- Erstellen und Konfigurieren eines Wartungsfensters
- Anzeigen von Informationen zu Wartungsfenstern
- Anzeigen von Informationen über Wartungsfenster-Aufgaben und Aufgabenausführungen
- Aktualisieren eines Wartungsfensters
- Löschen eines Wartungsfensters

Erfüllen der Voraussetzungen

Erfüllen Sie die folgenden Voraussetzungen, bevor Sie diese Tutorials ausführen.

- Konfigurieren Sie das AWS CLI auf Ihrem lokalen Computer — Bevor Sie AWS CLI Befehle ausführen können, müssen Sie die CLI auf Ihrem lokalen Computer installieren und konfigurieren. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.
- Überprüfen Sie die Rollen und Berechtigungen für das Wartungsfenster — Ein AWS Administrator in Ihrem Konto muss Ihnen die AWS Identity and Access Management (IAM-) Berechtigungen

gewähren, die Sie für die Verwaltung von Wartungsfenstern mithilfe der CLI benötigen. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

- Erstellen oder Konfigurieren einer mit Systems Manager kompatiblen Instance Sie benötigen mindestens eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance, die für die Verwendung mit Systems Manager konfiguriert ist, um die Tutorials abzuschließen. Dies bedeutet, dass SSM Agent auf der Instance installiert ist und der Instance ein IAM-Instance-Profil für Systems Manager zugeordnet ist.

  Wir empfehlen, eine Instance von einer AWS verwalteten Instanz Amazon Machine Image (AMI) aus zu starten, wobei der Agent vorinstalliert ist. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

  Weitere Informationen zum Installieren von SSM Agent auf einer Instance finden Sie in den folgenden Themen:

  - Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Windows Server
  - Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux

  Informationen zur Konfiguration von IAM-Berechtigungen für Systems Manager für Ihre Instance finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.
- Erstellen zusätzlicher Ressourcen nach Bedarf: Run Command, eine Funktion von Systems Manager, enthält viele Aufgaben, für die Sie keine anderen Ressourcen erstellen müssen als die, die in diesem Thema über die Voraussetzungen aufgeführt sind. Aus diesem Grund stellen wir Ihnen eine einfache Run Command-Aufgabe bereit, die Sie verwenden können, wenn Sie sich das erste Mal durch die Tutorials arbeiten. Sie benötigen außerdem eine EC2-Instance, die für die Verwendung mit Systems Manager konfiguriert ist, wie weiter oben in diesem Thema beschrieben. Nachdem Sie diese Instance konfiguriert haben, können Sie eine einfache Run Command-Aufgabe registrieren.

  Die Systems-Manager-Funktion Maintenance Windows unterstützt die Ausführung von vier Arten von Aufgaben:

  - Run Command-Befehle
  - Systems Manager Automation-Workflows
  - AWS Lambda Funktionen
  - AWS Step Functions Aufgaben

  Wenn eine Wartungsfensteraufgabe, die Sie ausführen möchten, zusätzliche Ressourcen erfordert, gilt im Allgemeinen, dass Sie diese zuerst erstellen sollten. Wenn Sie beispielsweise

ein Wartungsfenster wünschen, in dem eine AWS Lambda Funktion ausgeführt wird, erstellen Sie die Lambda-Funktion, bevor Sie beginnen. Erstellen Sie für eine Run Command Aufgabe den S3-Bucket, in dem Sie die Befehlsausgabe speichern können (falls Sie dies planen), und so weiter.

Verfolgen der Ressourcen-IDs

Behalten Sie bei der Ausführung der Aufgaben in diesem AWS CLI Tutorial die Ressourcen-IDs im Auge, die durch die von Ihnen ausgeführten Befehle generiert wurden. Sie können viele davon als Eingabe für nachfolgende Befehle verwenden. Wenn Sie beispielsweise das Wartungsfenster erstellen, stellt das System eine Wartungsfenster-ID im folgenden Format für Sie bereit.

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

Notieren Sie sich die folgenden systemgenerierten IDs, da sie in den Tutorials in diesem Abschnitt verwendet werden:

- `WindowId`
- `WindowTargetId`
- `WindowTaskId`
- `WindowExecutionId`
- `TaskExecutionId`
- `InvocationId`
- `ExecutionId`

Sie benötigen außerdem die ID der EC2-Instance, die Sie im Tutorial verwenden möchten. Zum Beispiel: `i-02573cafcfEXAMPLE`

Tutorials

- Tutorial: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI)
- Tutorial: Anzeigen von Informationen zu Wartungsfenstern (AWS CLI)
- Tutorial: Anzeigen von Informationen über Aufgaben und Aufgabenausführungen (AWS CLI)
- Tutorial: Aktualisieren eines Wartungsfensters (AWS CLI)
- Tutorial: Löschen eines Wartungsfensters (AWS CLI)

Tutorial: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI)

In diesem Tutorial wird gezeigt, wie Sie die AWS Command Line Interface (AWS CLI) zur Erstellung und Konfiguration eines Wartungsfensters, seiner Ziele und seiner Aufgaben verwenden können. Der Hauptpfad durch das Tutorial besteht aus einfachen Schritten. Sie erstellen ein einziges Wartungsfenster, identifizieren ein einziges Ziel und richten eine einfache Aufgabe ein, die im Wartungsfenster ausgeführt werden soll. Entlang des Pfades stellen wir Informationen bereit, die Sie beim Ausprobieren komplexerer Szenarien unterstützen.

Wenn Sie die Schritte in diesem Tutorial ausführen, ersetzen Sie die Werte in kursiv-*`rotem`* Text durch Ihren eigenen Optionen und IDs. Ersetzen Sie z. B. die Wartungsfenster-ID *`MW-0C50858D01Beispiel`* und die Instance-ID *`i-02573CafcfBeispiel`* mit IDs der Ressourcen, die Sie erstellen.

Inhalt

- Schritt 1: Erstellen des Wartungsfensters (AWS CLI)
- Schritt 2: Registrieren eines Ziel-Knotens mit dem Wartungsfenster (AWS CLI)
- Schritt 3: Registrieren einer Aufgaben für das Wartungsfenster (AWS CLI)

Schritt 1: Erstellen des Wartungsfensters (AWS CLI)

In diesem Schritt erstellen Sie ein Wartungsfenster und geben die grundlegenden Optionen, wie z. B. Name, Zeitplan und Dauer, an. In späteren Schritten, wählen Sie die Instance aus, die aktualisiert werden soll, und legen die Aufgabe fest, die ausgeführt wird.

In unserem Beispiel erstellen Sie ein Wartungsfenster, das alle fünf Minuten ausgeführt wird. Normalerweise würden Sie ein Wartungsfenster nicht so häufig ausführen. Mit dieser Rate werden Ihre Ergebnisse des Tutorials schneller ersichtlich. Wir zeigen Ihnen, wie Sie zu einer weniger häufigen Rate wechseln, nachdem die Aufgabe erfolgreich ausgeführt wurde.

> **Note**
>
> Eine Beschreibung des Verhältnisses zwischen den verschiedenen zeitplanbezogenen Optionen für Wartungsfenster finden Sie unter Wartungsfenster-Optionen für Planung und aktive Zeiträume.
> Weitere Informationen zum Arbeiten mit der `--schedule`-Option finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

So erstellen Sie ein Wartungsfenster (AWS CLI)

1. Öffnen Sie die AWS Command Line Interface (AWS CLI) und führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus, um ein Wartungsfenster zu erstellen, das sich folgendermaßen verhält:

   - Es wird (je nach Bedarf) über bis zu zwei Stunden hinweg alle fünf Minuten ausgeführt.
   - Verhindert, dass bis zu einer Stunde nach der Ausführung des Wartungsfensters neue Aufgaben gestartet werden.
   - Es ermöglicht nicht zugeordnete Ziele (Instances, die Sie nicht beim Wartungsfenster registriert haben).
   - Es gibt durch die Verwendung von benutzerdefinierten Tags an, dass sein Ersteller beabsichtigt, es in einem Tutorial zu verwenden.

   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"
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
       "WindowId":"mw-0c50858d01EXAMPLE"
   ```

```
}
```

2. Führen Sie jetzt den folgenden Befehl aus, um Details zu diesem und allen anderen Wartungsfenstern anzuzeigen, die Ihrem Konto bereits zugeordnet sind.

```
aws ssm describe-maintenance-windows
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowIdentities":[
      {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "Enabled": true,
            "Duration": 2,
            "Cutoff": 1,
            "NextExecutionTime": "2019-05-11T16:46:16.991Z"
      }
   ]
}
```

Fahren Sie fort mit Schritt 2: Registrieren eines Ziel-Knotens mit dem Wartungsfenster (AWS CLI).

Schritt 2: Registrieren eines Ziel-Knotens mit dem Wartungsfenster (AWS CLI)

In diesem Schritt registrieren Sie ein Ziel für Ihr neues Wartungsfenster. In diesem Fall geben Sie an, welcher Knoten aktualisiert werden soll, wenn das Wartungsfenster ausgeführt wird.

Ein Beispiel für die gleichzeitige Registrierung von mehr als einem Knoten mit Knoten-IDs, Beispiele zur Verwendung von Tags zur Identifizierung mehrerer Knoten und Beispiele für die Angabe von Ressourcengruppen als Ziele finden Sie unter Beispiele: Registrieren von Zielen für ein Wartungsfenster.

> **Note**
>
> Sie sollten bereits eine in diesem Schritt zu verwendende Amazon Elastic Compute Cloud (Amazon EC2)-Instance nach der Beschreibung in den Voraussetzungen für das Maintenance Windows-Tutorial erstellt haben.

So melden Sie einen Ziel-Knoten mit einem Wartungsfenster (AWS CLI) an

1. Führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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"
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
      "WindowTargetId":"e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
   }
   ```

2. Jetzt führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus, um Details zu Ihrem Wartungsfensterziel anzuzeigen.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-targets \
       --window-id "mw-0c50858d01EXAMPLE"
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-targets ^
       --window-id "mw-0c50858d01EXAMPLE"
   ```

   Das System gibt unter anderem folgende Informationen zurück

```
{
    "Targets": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-02573cafcfEXAMPLE"
                    ]
                }
            ]
        }
    ]
}
```

Fahren Sie fort mit Schritt 3: Registrieren einer Aufgaben für das Wartungsfenster (AWS CLI).

Beispiele: Registrieren von Zielen für ein Wartungsfenster

Sie können einen einzelnen Knoten mithilfe der Knoten-ID als Ziel registrieren. Die Anleitung dazu finden Sie unter Schritt 2: Registrieren eines Ziel-Knotens mit dem Wartungsfenster (AWS CLI). Darüber hinaus haben Sie die Möglichkeit, einen oder mehrere Knoten mithilfe der Befehlsformate auf dieser Seite als Ziele zu registrieren.

Im Allgemeinen gibt es zwei Methoden, um Knoten als Ziele für das Wartungsfenster zu identifizieren: durch das Festlegen einzelner Knoten und mithilfe von Ressourcen-Tags. Die Ressourcen-Tags-Methode bietet weitere Optionen, wie in den Beispielen 2 bis 3 gezeigt.

Sie können auch eine oder mehrere Ressourcengruppen als Ziel eines Wartungsfensters angeben. Eine Ressourcengruppe kann Knoten einschließen und viele andere Arten von unterstützten AWS-Ressourcen. Die Beispiele 4 und 5 demonstrieren nun, wie Sie Ihren Zielen für das Wartungsfenster Ressourcengruppen hinzufügen.

> **Note**
>
> Wenn eine einzelne Wartungsfenster-Aufgabe mit mehreren Zielen registriert ist, werden ihre Aufrufe sequenziell und nicht parallel ausgeführt. Wenn Ihre Aufgabe gleichzeitig auf

mehreren Zielen ausgeführt werden muss, registrieren Sie eine Aufgabe für jedes Ziel einzeln, und weisen Sie jeder Aufgabe dieselbe Prioritätsstufe zu.

Weitere Informationen zum Erstellen und Verwalten von Ressourcengruppen finden Sie unter Was sind Ressourcengruppen? im AWS Resource Groups-Benutzerhandbuch und unter Resource Groups and Tagging for AWS im AWS News Blog.

Für Informationen über Kontingente für Maintenance Windows, eine Fähigkeit von AWS Systems Manager, zusätzlich zu den in den folgenden Beispielen angegebenen, siehe Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz.

Beispiel 1: Registrieren mehrerer Ziele mithilfe von Knoten-IDs

Führen Sie den folgenden Befehl auf Ihrer lokalen Maschine aus, um mehrere Knoten anhand ihrer Knoten-IDs als Ziele anzumelden. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

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
```

Empfohlene Verwendung: Besonders nützlich bei der erstmaligen Registrierung einer eindeutigen Gruppe von Knoten bei einem beliebigen Wartungsfenster, wenn die Knoten nicht über ein gemeinsames Knoten-Tag verfügen.

Kontingente: Sie können insgesamt bis zu 50 Knoten für jedes Wartungsfenster-Ziel angeben.

Beispiel 2: Registrieren von Zielen mithilfe von Ressourcen-Tags, die auf Knoten angewendet werden

Führen Sie den folgenden Befehl auf Ihrer lokalen Maschine aus, um alle Knoten anzumelden, die bereits mit einem von Ihnen zugewiesenen Schlüssel-Wert-Paar markiert wurden. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

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"
```

Empfohlene Verwendung: Besonders nützlich bei der erstmaligen Registrierung einer eindeutigen Gruppe von Knoten bei einem beliebigen Wartungsfenster, wenn die Knoten schon über ein gemeinsames Knoten-Tag verfügen.

Kontingente: Sie können insgesamt bis zu fünf Schlüssel-Wert-Paare für jedes Ziel festlegen. Wenn Sie mehr als ein Schlüssel-Wert-Paar angeben, muss ein Knoten mit allen Tag-Schlüsseln und Werten gekennzeichnet werden, die Sie für die Aufnahme in die Zielgruppe angeben.

> **Note**
>
> Sie können eine Gruppe von Knoten mit dem Tag-Schlüssel `Patch Group` oder `PatchGroup` markieren und die Knoten einem gemeinsamen Schlüsselwert zuweisen, z. B. `my-patch-group`. (Sie müssen `PatchGroup` ohne Leerzeichen verwenden, wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben.) Patch Manager, eine Funktion von Systems Manager, wertet den Schlüssel `Patch Group` oder `PatchGroup` auf Knoten aus, um zu bestimmen, welche Patch-Baseline für sie gilt. Wenn Ihre Aufgabe das `AWS-RunPatchBaseline`-SSM-Dokument (oder das Legacy-`AWS-ApplyPatchBaseline`-SSM-Dokument) ausführt, können Sie das gleiche `Patch Group`- oder `PatchGroup`-Schlüssel-Wert-Paar angeben, wenn Sie Ziele für ein Wartungsfenster registrieren. Beispiel:

`--target "Key=tag:PatchGroup,Values=my-patch-group`. Auf diese Weise können Sie Patches für eine Gruppe von Knoten, die schon der gleichen Patch-Baseline zugeordnet sind, mithilfe eines Wartungsfensters aktualisieren. Weitere Informationen finden Sie unter Patch-Gruppen.

Beispiel 3: Registrieren von Zielen mithilfe einer Gruppe von Tag-Schlüsseln (ohne Tag-Werte)

Führen Sie den folgenden Befehl auf Ihrer lokalen Maschine aus, um Knoten anzumelden, denen alle ein oder mehrere Tag-Schlüssel zugeordnet wurden, unabhängig von ihren Schlüsselwerten. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

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"
```

Empfohlene Verwendung: Diese Option ist hilfreich, wenn Sie Knoten als Ziele verwenden möchten, indem Sie mehrere Tag-Schlüssel (ohne Werte) angeben und nicht nur einen Tag-Schlüssel oder ein Tag-Schlüssel-Wert-Paar.

Kontingente: Sie können insgesamt bis zu fünf Tag-Schlüssel für jedes Ziel festlegen. Wenn Sie mehr als einen Tag-Schlüssel angeben, muss ein Knoten mit allen Tag-Schlüsseln und Werten markiert werden, die Sie für die Aufnahme in die Zielgruppe angeben.

Beispiel 4: Registrieren von Zielen unter Verwendung eines Ressourcengruppennamens

Führen Sie den folgenden Befehl zum Registrieren einer bestimmten Ressourcengruppe auf Ihrem lokalen Computer aus, unabhängig von der Art der Ressourcen, die sie enthält. Ersetzen Sie *mw-0c50858d01EXAMPLE* durch Ihre eigenen Informationen. Wenn die Aufgaben, die Sie

dem Wartungsfenster zuweisen, auf eine Art von Ressource in dieser Ressourcengruppe nicht angewendet werden kann, meldet das System möglicherweise einen Fehler. Auch wenn ein solcher Fehler gemeldet wird, werden Aufgaben, für die ein unterstützter Ressourcentyp gefunden wurde, dennoch ausgeführt.

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"
```

Empfohlene Verwendung: Diese Vorgehensweise ist hilfreich, wenn Sie schnell eine Ressourcengruppe als Ziel angeben möchten, ohne auszuwerten, ob alle Ressourcentypen Ziel des Wartungsfensters sind, oder wenn Sie wissen, dass die Ressourcengruppe nur solche Ressourcentypen enthält, über denen Ihre Aufgaben Aktionen durchführen können.

Kontingente: Sie können nur eine Ressourcengruppe als Ziel angeben.

Beispiel 5: Registrieren von Zielen durch Filtern von Ressourcentypen in einer Ressourcengruppe

Führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus, um nur bestimmte Ressourcentypen zu registrieren, die einer Ressourcengruppe angehören, die Sie angeben. Ersetzen Sie *mw-0c50858d01EXAMPLE* durch Ihre eigenen Informationen. Bei dieser Vorgehensweise wird eine Aufgabe, selbst wenn Sie sie für einen Ressourcentyp hinzufügen, der der Ressourcengruppe angehört, nicht ausgeführt, wenn Sie den Filter nicht explizit auf den Ressourcentyp setzen

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"
```

Empfohlene Verwendung: Verwenden Sie diese Vorgehensweise, wenn Sie genau kontrollieren möchten, über welchen Typen von AWS-Ressourcen Ihr Wartungsfensters Aktionen ausführen kann, oder wenn Ihre Ressourcengruppe eine große Anzahl von Ressourcentypen enthält und Sie vermeiden möchten, dass unnötig viele Fehler im Wartungsfensterprotokoll aufgenommen werden.

Kontingente: Sie können nur eine Ressourcengruppe als Ziel angeben.

Schritt 3: Registrieren einer Aufgaben für das Wartungsfenster (AWS CLI)

In diesem Schritt des Tutorials registrieren Sie eine AWS Systems Manager Run Command-Aufgabe, die den `df`-Befehl auf Ihrer Amazon Elastic Compute Cloud (Amazon EC2)-Instance für Linux ausführt. Die Ergebnisse dieses Standard-Linux-Befehls zeigen, wie viel Speicherplatz frei ist und wie viel Speicherplatz auf dem Festplatten-Dateisystem Ihrer Instance belegt wird.

–oder–

Wenn Sie die Anweisungen an eine Amazon-EC2-Instance für Windows Server anstatt für Linux richten, ersetzen Sie df im folgenden Befehl durch ipconfig. Die Ausgabe dieses Befehls enthält Details über die IP-Adresse, die Subnetzmaske und das Standard-Gateway für Adapter auf der Ziel-Instance.

Wenn Sie zum Registrieren anderer Aufgabentypen oder zur Verwendung weiterer verfügbarer Run Command Systems Manager-Optionen bereit sind, finden Sie unter Beispiele: Registrieren von Aufgaben für ein Wartungsfenster weitere Informationen. Dort befinden sich weitere Informationen zu allen vier Aufgabentypen und einigen ihrer wichtigsten Optionen, die Sie beim Planen umfassenderer realer Szenarien unterstützen.

So registrieren Sie eine Aufgabe für ein Wartungsfenster

1. Führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Die Version, die von einem lokalen Windows-Computer aus ausgeführt werden soll, enthält die Escape-Zeichen („/“), die Sie zum Ausführen des Befehls über Ihr Befehlszeilen-Tool benötigen.

   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\"]}}}
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden:

   ```
   {
       "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
   }
   ```

2. Führen Sie nun den folgenden Befehl aus, um Details zu der von Ihnen erstellten Wartungsfensteraufgabe anzuzeigen.

Linux & macOS

```
aws ssm describe-maintenance-window-tasks \
    --window-id mw-0c50858d01EXAMPLE
```

Windows

```
aws ssm describe-maintenance-window-tasks ^
    --window-id mw-0c50858d01EXAMPLE
```

3. Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "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. Räumen Sie basierend auf dem von Ihnen in Schritt 1: Erstellen des Wartungsfensters (AWS CLI) angegebenen Zeitplan genügend Zeit für die Aufgabenausführung ein. Wenn Sie **`--schedule "rate(5 minutes)"`** angegeben haben, warten Sie beispielsweise fünf

Minuten. Führen Sie dann den folgenden Befehl aus, um Informationen über alle Ausführungen anzuzeigen, die für diese Aufgabe aufgetreten sind.

Linux & macOS

```
aws ssm describe-maintenance-window-executions \
    --window-id mw-0c50858d01EXAMPLE
```

Windows

```
aws ssm describe-maintenance-window-executions ^
    --window-id mw-0c50858d01EXAMPLE
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593493.096,
            "EndTime": 1557593498.611
        }
    ]
}
```

Tip

Nachdem die Aufgabe erfolgreich durchgeführt wurde, können Sie die Rate verringern, mit der das Wartungsfenster ausgeführt wird. Führen Sie beispielsweise den folgenden Befehl aus, um die Häufigkeit auf einmal pro Woche zu verringern. Ersetzen Sie *mw-0c50858d01EXAMPLE* durch Ihre eigenen Informationen.

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)"
```

Weitere Informationen zum Verwalten von Wartungsfenster-Zeitplänen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager und Wartungsfenster-Optionen für Planung und aktive Zeiträume.
Weitere Informationen zur Verwendung der AWS Command Line Interface (AWS CLI) zum Ändern eines Wartungsfensters finden Sie unter Tutorial: Aktualisieren eines Wartungsfensters (AWS CLI).

Um sich in der Ausführung von AWS CLI-Befehlen für die Ansicht weiterer Details zu Ihrer Wartungsfenster-Aufgabe und deren Ausführungen zu üben, fahren Sie mit Tutorial: Anzeigen von Informationen über Aufgaben und Aufgabenausführungen (AWS CLI) fort.

Über die Tutorial-Befehlsausgabe

Es geht über den Rahmen dieses Tutorials hinaus, mithilfe der AWS CLI die Ausgabe des Run Command-Befehls für Ihre Wartungsfensteraufgabe-Ausführungen anzuzeigen.

Sie könnten diese Daten jedoch mithilfe der AWS CLI anzeigen. (Sie können die Ausgabe auch in der Systems Manager-Konsole oder in einer in einem Amazon Simple Storage Service (Amazon S3)-Bucket gespeicherten Protokolldatei anzeigen, sofern Sie das Wartungsfenster zur Befehlsausgabe an dieser Stelle konfiguriert haben.) In diesem Fall würden Sie feststellen, dass die Ausgabe des Befehls df auf einer EC2-Instance für Linux ähnlich der folgenden ist.

```
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% /
```

Die Ausgabe des Befehls ipconfig auf einer EC2-Instance für Windows Server sieht in etwa wie folgt aus:

```
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  . :
```

Beispiele: Registrieren von Aufgaben für ein Wartungsfenster

Sie können eine Aufgabe inRun Command, eine Fähigkeit von AWS Systems Manager, mit einem Wartungsfenster registrieren, indem Sie die AWS Command Line Interface (AWS CLI) verwenden, wie unter Aufgaben im Wartungsfenster registrieren gezeigt. Sie können auch Aufgaben für Workflows, AWS Lambda Funktionen und AWS Step Functions Aufgaben von Systems Manager Automation registrieren, wie weiter unten in diesem Thema gezeigt wird.

> **Note**
>
> Geben Sie ein oder mehrere Ziele für Wartungsfenster Run Command-Typ-Aufgaben an. Je nach Aufgabe sind Ziele für andere Aufgabentypen im Wartungsfenster (Automatisierung AWS Lambda, und AWS Step Functions) optional. Weitere Informationen zur Ausführung von Aufgaben, die keine Ziele angeben, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

In diesem Thema finden Sie Beispiele für die Verwendung des Befehls AWS Command Line Interface (AWS CLI)`register-task-with-maintenance-window`, um jeden der vier unterstützten Aufgabentypen in einem Wartungsfenster zu registrieren. Die Beispiele dienen nur zur Veranschaulichung. Sie können sie abwandeln, um funktionsfähige Befehle zur Aufgabenregistrierung zu erstellen.

Verwenden der cli-input-json Option --

Zur besseren Verwaltung Ihrer Aufgabenoptionen können Sie die Befehlsoption `--cli-input-json` mit in einer JSON-Datei referenzierten Optionswerten verwenden.

Um den Inhalt der JSON-Beispieldatei zu verwenden, den wir in den folgenden Beispielen bereitgestellt haben, führen Sie auf Ihrem lokalen Computer die die folgenden Schritte aus:

1. Erstellen Sie eine Datei mit einem Namen wie z. B. `MyRunCommandTask.json`, `MyAutomationTask.json` oder einem anderen von Ihnen bevorzugten Namen.
2. Kopieren Sie den Inhalt der JSON-Beispieldatei in die Datei.
3. Ändern Sie den Inhalt der Datei für Ihre Aufgabenregistrierung ab und speichern Sie dann die Datei.
4. Führen Sie in demselben Verzeichnis, in dem Sie die Datei gespeichert haben, den folgenden Befehl aus. Ersetzen Sie *`MyFile.json`* durch Ihren Dateinamen.

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
```

Informationen zu Pseudoparametern

In einigen Beispielen verwenden wir Pseudoparameter als Methode zur Übergabe von ID-Informationen an Ihre Aufgaben. Zum Beispiel werden `{{TARGET_ID}}` und `{{RESOURCE_ID}}` verwendet, um IDs von AWS -Ressourcen an Automation-, Lambda- und Step Functions-Aufgaben zu übergeben. Weitere Informationen zu Pseudoparametern im `--task-invocation-parameters`-Inhalt finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben.

Weitere Informationen

- Über register-task-with-maintenance -windows-Optionen.
- register-task-with-maintenance-window in der AWS CLI Befehlsreferenz
- RegisterTaskWithMaintenanceWindow in der AWS Systems Manager -API-Referenz

Beispiele der Aufgabenregistrierung

Die folgenden Abschnitte enthalten einen AWS CLI Beispielbefehl für die Registrierung eines unterstützten Aufgabentyps und ein JSON-Beispiel, das mit der `--cli-input-json` Option verwendet werden kann.

Registrieren einer Systems Manager Run Command-Aufgabe

Die folgenden Beispiele veranschaulichen, wie Sie Aufgaben von Systems Manager Run Command-Aufgaben mithilfe der AWS CLI bei einem Wartungsfenster registrieren.

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\"]}}}"
```

JSON-Inhalt für die Verwendung mit der Dateioption **`--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": "DOC-EXAMPLE-BUCKET",
            "OutputS3KeyPrefix": "S3-PREFIX",
            "TimeoutSeconds": 3600
        }
    }
}
```

Registrieren einer Systems Manager Automation-Aufgabe

Die folgenden Beispiele veranschaulichen, wie Systems Manager Automation-Aufgaben mithilfe der bei einem Wartungsfenster registriert werden AWS CLI:

AWS CLI Befehl:

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"
```

JSON-Inhalt für die Verwendung mit der Dateioption **`--cli-input-json`**:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
        "TaskArn": "AWS-PatchInstanceWithRollback",
    "TaskType": "AUTOMATION","TaskInvocationParameters": {
        "Automation": {
            "DocumentVersion": "1",
            "Parameters": {
                "instanceId": [
                    "{{RESOURCE_ID}}"
                ]
            }
        }
    }
}
```

Registrieren einer AWS Lambda -Aufgabe

Die folgenden Beispiele veranschaulichen, wie Lambda-Funktionsaufgaben mithilfe der AWS CLI bei einem Wartungsfenster registriert werden.

Bei diesen Beispielen hat der Benutzer, der die Lambda-Funktion erstellt hat, ihr den Namen `SSMrestart-my-instances` gegeben und zwei Parameter mit dem Namen `instanceId` und `targetType` erstellt.

> **Important**
>
> Die IAM-Richtlinie für Maintenance Windows erfordert, dass Sie den Namen von Lambda-Funktionen (oder Aliasen) das Präfix `SSM` hinzufügen. Bevor Sie mit der Registrierung dieser Art von Aufgabe fortfahren, aktualisieren Sie ihren Namen so, dass er AWS Lambda einschließt`SSM`. Beispiel: Wenn Ihr Lambda-Funktionsname `MyLambdaFunction` lautet, ändern Sie ihn in `SSMMyLambdaFunction`.

AWS CLI Befehl:

Linux & macOS

> **Important**
>
> Wenn Sie Version 2 von verwenden AWS CLI, müssen Sie die Option `--cli-binary-format raw-in-base64-out` in den folgenden Befehl aufnehmen, wenn Ihre Lambda-Payload nicht base64-codiert ist. Die Option `cli_binary_format` ist nur in Version 2 verfügbar. Informationen zu diesen und anderen AWS CLI *`config`* Dateieinstellungen finden Sie im Benutzerhandbuch unter Unterstützte *`config`* Dateieinstellungen.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**
>
> Wenn Sie Version 2 von verwenden AWS CLI, müssen Sie die Option `--cli-binary-format raw-in-base64-out` in den folgenden Befehl aufnehmen, wenn Ihre Lambda-Payload nicht base64-codiert ist. Die Option `cli_binary_format` ist nur in Version 2 verfügbar. Informationen zu diesen und anderen AWS CLI *`config`* Dateieinstellungen finden Sie im Benutzerhandbuch unter Unterstützte *`config`* Dateieinstellungen.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\"}}'
```

JSON-Inhalt für die Verwendung mit der Dateioption **`--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
}
```

Register a Step Functions task (Eine Step Functions-Aufgabe registrieren)

Die folgenden Beispiele veranschaulichen, wie Sie Aufgaben von Step Functions-Zustandsautomaten mithilfe der AWS CLI bei einem Wartungsfenster registrieren.

> **Note**
>
> Aufgaben im Wartungsfenster unterstützen nur Step Functions Standard-State-Machine-Workflows. Sie unterstützen keine Express-State-Machine-Workflows. Informationen zu Workflowtypen für Zustandsmaschinen finden Sie unter Standard- und Express-Workflows im AWS Step Functions Entwicklerhandbuch.

In diesen Beispielen erstellte der Benutzer, der den Step Functions-Zustandsautomaten erstellt hatte, einen Zustandsautomaten mit dem Namen „`SSMMyStateMachine`“ und dem Parameter „`instanceId`“.

> **Important**
>
> Die AWS Identity and Access Management (IAM-) Richtlinie für Maintenance Windows erfordert, dass Sie Step Functions Functions-Zustandsmaschinen das Präfix voranstellen. `SSM` Bevor Sie mit der Registrierung dieser Art von Aufgabe fortfahren, müssen Sie ihren Namen so aktualisieren, dass er AWS Step Functions einschließt`SSM`. Beispiel: Wenn der Name des Zustandsautomaten `MyStateMachine` lautet, ändern Sie ihn in `SSMMyStateMachine`.

AWS CLI Befehl:

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"
```

JSON-Inhalt für die Verwendung mit der Dateioption **`--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
}
```

Über register-task-with-maintenance -windows-Optionen

Der Befehl register-task-with-maintenance-window bietet mehrere Optionen für die Konfiguration einer Aufgabe entsprechend Ihren Anforderungen. Einige sind erforderlich, einige sind optional und einige gelten nur für einen einzigen Wartungsfenster-Aufgabentyp.

In diesem Thema erhalten Sie Informationen zu einigen dieser Optionen, um Sie bei der Arbeit mit Beispielen in diesem Abschnitt des Tutorials zu unterstützen. Informationen über alle Befehlsoptionen finden Sie unter register-task-with-maintenance-window in der AWS CLI Command Reference.

Informationen über die Option **`--task-arn`**

Die Option `--task-arn` wird verwendet, um die Ressource anzugeben, die von der Aufgabe ausgeführt wird. Der von Ihnen angegebene Wert hängt wie in der folgenden Tabelle beschrieben, davon ab, welche Art von Aufgabe Sie registrieren möchten.

TaskArn Formate für Wartungsfensteraufgaben

Wartungsfenster-Aufgabentyp	TaskArn Wert
`RUN_COMMAND` und **`AUTOMATION`**	`TaskArn` ist der SSM-Dokumentname oder der Amazon-Ressourcenname (ARN). Zum Beispiel: `AWS-RunBatchShellScript` –oder– `arn:aws:ssm:` *`region`*`:11112222 3333:document/My-Document` .
`LAMBDA`	`TaskArn` ist der Funktionsname oder -ARN. Zum Beispiel: `SSMMy-Lambda-Function` –oder– `arn:aws:lambda:` *`region`*`:11112222 3333:function:SSMMyLambdaFu nction` .

Wartungsfenster-Aufgabentyp	TaskArn Wert
	Important Die IAM-Richtlinie für Maintenance Windows erfordert, dass Sie den Namen von Lambda-Funktionen (oder Aliasen) das Präfix `SSM` hinzufügen. Bevor Sie mit der Registrierung dieser Art von Aufgabe fortfahren, aktualisieren Sie ihren Namen so, dass er AWS Lambda einschließt`SSM`. Beispiel: Wenn Ihr Lambda-Funktionsname `MyLambdaFunction` lautet, ändern Sie ihn in `SSMMyLambdaFunction` .
`STEP_FUNCTIONS`	`TaskArn` ist der ARN des Zustandsautomaten. Zum Beispiel: `arn:aws:states:us-east-2:111122223333:stateMachine:SSMMyStateMachine` . **Important** Die IAM-Richtlinie für Wartungsfenster erfordert, dass Sie Step Functions-Zustandsautomaten-Namen das Präfix `SSM` geben. Bevor Sie diesen Aufgabentyp registrieren, müssen Sie seinen Namen in AWS Step Functions Include aktualisieren`SSM`. Beispiel: Wenn der Name des Zustandsautomaten `MyStateMachine` lautet, ändern Sie ihn in `SSMMyStateMachine` .

Informationen über die Option **`--service-role-arn`**

Die Rolle AWS Systems Manager , die bei der Ausführung der Wartungsfensteraufgabe übernommen werden soll.

Weitere Informationen finden Sie unter Einrichten von Maintenance Windows

Informationen über die Option **`--task-invocation-parameters`**

Die Option `--task-invocation-parameters` wird dazu verwendet, jene Parameter anzugeben, die nur für die vier Aufgabentypen gelten. Die unterstützten Parameter für jede der vier Arten von Aufgaben werden in der folgenden Tabelle beschrieben.

> **Note**
>
> Weitere Informationen über die Verwendung von Pseudoparametern in `--task-invocation-parameters`-Inhalten, z. B. {{TARGET_ID}}, finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben.

Aufgabenaufruf-Parameteroptionen für Wartungsfenster-Aufgaben

Wartungsfenster-Aufgabentyp	Verfügbare Parameter	Beispiel
RUN_COMMAND	Kommentar DocumentHash DocumentHashType NotificationConfig Ausgänge: 3 BucketName OutPutS3 KeyPrefix Parameter ServiceRoleArn TimeoutSeconds	`"TaskInvocationParameters": { "RunCommand": { "Comment": "My Run Command task comment", "DocumentHash": "6554ed3d--truncated--5EXAMPLE", "DocumentHashType": "Sha256", "NotificationConfig": { "NotificationArn": "arn:aws:sns: region:12345678`

Wartungsfenster-Aufgabentyp	Verfügbare Parameter	Beispiel

```
9012:my-sns-topic-
name",

 "NotificationEvents":
 [

 "FAILURE"
                ],

 "NotificationType":
 "Invocation"
            },
            "OutputS3
BucketName": "DOC-EXAM
PLE-BUCKET",
            "OutputS3
KeyPrefix": "  S3-PREFIX
 ",
            "Paramete
rs": {

 "commands": [

 "Get-ChildItem$env:
 temp-Recurse|Remove-
Item-Recurse-force"
                ]
            },
            "ServiceR
oleArn": "arn:aws:
iam::123456789012:
role/MyMaintenance
WindowServiceRole",
            "TimeoutS
econds": 3600
        }
    }
```

Wartungsfenster-Aufgabentyp	Verfügbare Parameter	Beispiel
-Automatisierung	DocumentVersion Parameter	<pre>"TaskInvocationPar ameters": { "Automation": { "Document Version": "3", "Paramete rs": { "instanceid": ["{{TARGET_ID}}"] } } }</pre>
LAMBDA	ClientContext Nutzlast Qualifier	<pre>"TaskInvocationPar ameters": { "Lambda": { "ClientCo ntext": "ew0KICAi --truncated--0KIEX AMPLE", "Payload" : "{ \"targetId\": \"{{TARGET_ID}}\", \"targetType\": \"{{TARGET_TYPE}}\ " }", "Qualifie r": "$LATEST" } }</pre>

Wartungsfenster-Aufgabentyp	Verfügbare Parameter	Beispiel
STEP_FUNCTIONS	Eingabe Name	<pre>"TaskInvocationPar ameters": { "StepFunc tions": { "Input": "{ \"targetId\": \"{{TARGET_ID}}\" }", "Name": "{{INVOCATION_ID}}" } }</pre>

Tutorial: Anzeigen von Informationen zu Wartungsfenstern (AWS CLI)

In diesem Tutorial sind Befehle enthalten, mit denen Sie Ihre Wartungsfenster, Aufgaben, Ausführungen und Aufrufe aktualisieren oder Informationen darüber abrufen können. Die Beispiele sind nach Befehl geordnet, um zu zeigen, wie Befehlsoptionen verwendet werden, um nach der Art von Details zu filtern, die Sie anzeigen möchten.

Wenn Sie die Schritte in diesem Tutorial ausführen, ersetzen Sie die Werte in kursiv-*rotem* Text durch Ihren eigenen Optionen und IDs. Ersetzen Sie z. B. die Wartungsfenster-ID *MW-0C50858D01Beispiel* und die Instance-ID *i-02573CafcfBeispiel* mit IDs der Ressourcen, die Sie erstellen.

Weitere Informationen zum Einrichten und Konfigurieren der AWS Command Line Interface (AWS CLI), finden Sie unter Installation, Aktualisierung und Deinstallation von AWS CLI und Konfigurieren der AWS CLI.

Befehlsbeispiele

- Beispiele für 'describe-maintenance-windows' (wartungsfenster-beschreiben)
- Beispiele für 'describe-maintenance-windows' (wartungsfenster-ziele-beschreiben)
- Beispiele für 'describe-maintenance-windows' (wartungsfenster-aufgaben-beschreiben)
- Beispiele für 'describe-maintenance-windows' (wartungsfenster-für-ziele-beschreiben)
- Beispiele für 'describe-maintenance-windows' (wartungsfenster-ausführungen-beschreiben)
- Beispiele für 'describe-maintenance-windows' (wartungsfenster-zeitplan-beschreiben)

Beispiele für 'describe-maintenance-windows' (wartungsfenster-beschreiben)

Alle Wartungsfenster in Ihrem AWS-Konto aufführen

Führen Sie den folgenden Befehl aus.

```
aws ssm describe-maintenance-windows
```

Das System gibt unter anderem folgende Informationen zurück

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

Alle aktivierten Wartungsfenster aufführen

Führen Sie den folgenden Befehl aus.

```
aws ssm describe-maintenance-windows --filters "Key=Enabled,Values=true"
```

Das System gibt unter anderem folgende Informationen zurück

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

Alle deaktivierten Wartungsfenster aufführen

Führen Sie den folgenden Befehl aus.

```
aws ssm describe-maintenance-windows --filters "Key=Enabled,Values=false"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-6e5c9d4b7cEXAMPLE",
            "Name": "My-Disabled-Maintenance-Window",
            "Enabled": false,
            "Duration": 2,
            "Cutoff": 1
        }
    ]
}
```

Alle Wartungsfenster aufführen, deren Name mit einem bestimmten Präfix beginnt

Führen Sie den folgenden Befehl aus.

```
aws ssm describe-maintenance-windows --filters "Key=Name,Values=My"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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
        }
    ]
}
```

Beispiele für 'describe-maintenance-windows' (wartungsfenster-ziele-beschreiben)

Die Ziele für ein Wartungsfenster anzeigen, das einem bestimmten Eigentümer-Informationswert entspricht

Führen Sie den folgenden Befehl aus.

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**
>
> Die unterstützten Filterschlüssel sind `Type`, `WindowTargetId` und `OwnerInformation`.

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Targets": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "tag:Name",
                    "Values": [
                        "Production"
                    ]
                }
            ],
            "OwnerInformation": "CostCenter1",
            "Name": "Target1"
        }
    ]
}
```

Beispiele für 'describe-maintenance-windows' (wartungsfenster-aufgaben-beschreiben)

Alle registrierten Aufgaben anzeigen, die das SSM-Befehlsdokument **`AWS-RunPowerShellScript`** aufrufen

Führen Sie den folgenden Befehl aus.

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"
```

Das System gibt unter anderem folgende Informationen zurück

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

Alle registrierten Aufgaben mit Priorität 3 anzeigen

Führen Sie den folgenden Befehl aus.

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"
```

Das System gibt unter anderem folgende Informationen zurück

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

Alle registrierten Aufgaben anzeigen, die Priorität "1" haben und Run Command verwenden

Führen Sie den folgenden Befehl aus.

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"
```

Das System gibt unter anderem folgende Informationen zurück

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

Beispiele für 'describe-maintenance-windows' (wartungsfenster-für-ziele-beschreiben)

Informationen über die Wartungsfensterziele oder -Aufgaben im Zusammenhang mit einem bestimmten Knoten aufführen

Führen Sie den folgenden Befehl aus.

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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window"
```

```
        }
    ]
}
```

Beispiele für 'describe-maintenance-windows' (wartungsfenster-ausführungen-beschreiben)

Alle Aufgaben aufführen, die vor einem bestimmten Datum ausgeführt wurden

Führen Sie den folgenden Befehl aus.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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
        }
    ]
}
```

Alle Aufgaben aufführen, die nach einem bestimmten Datum ausgeführt wurden

Führen Sie den folgenden Befehl aus.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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
        }
    ]
}
```

Beispiele für 'describe-maintenance-windows' (wartungsfenster-zeitplan-beschreiben)

Die nächsten zehn Wartungsfenster-Ausführungen, die für einen bestimmten Knoten geplant sind, anzeigen

Führen Sie den folgenden Befehl aus.

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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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"
}
```

Den Wartungsfenster-Zeitplan für Knoten anzeigen, die mit einem bestimmten Schlüssel-Wert-Paar markiert sind

Führen Sie den folgenden Befehl aus.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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"
}
```

Startzeiten für die nächsten vier Ausführungen eines Wartungsfensters anzeigen

Führen Sie den folgenden Befehl aus.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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: Anzeigen von Informationen über Aufgaben und Aufgabenausführungen (AWS CLI)

In diesem Tutorial wird veranschaulicht, wie Sie mithilfe der AWS Command Line Interface (AWS CLI) Details über von Ihnen abgeschlossene Wartungsfenster-Aufgaben anzuzeigen.

Wenn Sie direkt von Tutorial: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI) fortfahren, überprüfen Sie, dass genügend Zeit verstrichen ist, damit das Wartungsfenster mindestens einmal ausgeführt werden konnte, um die Ausführungsergebnisse anzuzeigen.

Wenn Sie die Schritte in diesem Tutorial ausführen, ersetzen Sie die Werte in kursiv-*rotem* Text durch Ihren eigenen Optionen und IDs. Ersetzen Sie z. B. die Wartungsfenster-ID *MW-0C50858D01Beispiel* und die Instance-ID *i-02573CafcfBeispiel* mit IDs der Ressourcen, die Sie erstellen.

So zeigen Sie Informationen über Aufgaben und Aufgabenausführungen an (AWS CLI)

1. Führen Sie den folgenden Befe.l aus, um eine Liste der Aufgabenausführungen für ein bestimmtes Wartungsfenster anzuzeigen:

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-executions \
   ```

```
    --window-id "mw-0c50858d01EXAMPLE"
```

Windows

```
aws ssm describe-maintenance-window-executions ^
    --window-id "mw-0c50858d01EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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. Führen Sie den folgenden Befehl aus, um Informationen zu der Aufgabenausführung eines Wartungsfensters abzurufen.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
    "TaskIds": [
        "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
    ],
    "Status": "SUCCESS",
    "StartTime": 1557593493.096,
    "EndTime": 1557593498.611
}
```

3. Führen Sie den folgenden Befehl aus, um eine Liste der Aufgabenausführungen als Teil einer Wartungsfenster-Ausführung anzuzeigen.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
```

```
    "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. Führen Sie den folgenden Befehl aus, um Details zu einer Aufgabenausführung abzurufen.

   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"
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
       "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. Führen Sie den folgenden Befehl aus, um die spezifischen Aufgabenaufrufe abzurufen, die bei einer Aufgabenausführung durchgeführt werden.

   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"
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
       "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: Aktualisieren eines Wartungsfensters (AWS CLI)

Dieses Tutorial zeigt, wie Sie die AWS Command Line Interface (AWS CLI) verwenden, um ein Wartungsfenster zu aktualisieren. Es zeigt Ihnen auch, wie Sie verschiedene Aufgabentypen aktualisieren, einschließlich der Aufgabentypen für AWS Systems Manager Run Command und Automatisierung AWS Lambda, und AWS Step Functions.

In den Beispielen dieses Abschnitts werden die folgenden Systems Manager-Aktionen zum Aktualisieren eines Wartungsfensters verwendet:

- UpdateMaintenanceWindow
- UpdateMaintenanceWindowTarget
- UpdateMaintenanceWindowTask
- DeregisterTargetFromMaintenanceWindow

Weitere Informationen zum Aktualisieren eines Wartungsfensters über die Systems Manager-Konsole finden Sie unter Aktualisieren oder Löschen von Wartungsfenster-Ressourcen (Konsole).

Wenn Sie die Schritte in diesem Tutorial ausführen, ersetzen Sie die Werte in kursiv-*rotem* Text durch Ihren eigenen Optionen und IDs. Ersetzen Sie z. B. die Wartungsfenster-ID *MW-0C50858D01Beispiel* und die Instance-ID *i-02573CafcfBeispiel* mit IDs der Ressourcen, die Sie erstellen.

So aktualisieren Sie ein Wartungsfenster (AWS CLI)

1. Öffnen Sie das AWS CLI und führen Sie den folgenden Befehl aus, um ein Ziel so zu aktualisieren, dass es einen Namen und eine Beschreibung enthält.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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. Führen Sie den folgenden Befehl aus, um mit der `replace`-Option das Beschreibungsfeld zu entfernen und ein zusätzliches Ziel hinzuzufügen. Das Beschreibungsfeld wird gelöscht, da die Aktualisierung das Feld nicht enthält (NULL-Wert). Stellen Sie sicher, dass Sie einen zusätzlichen Knoten angeben, der für die Verwendung mit Systems Manager konfiguriert wurde.

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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
    "Targets": [
        {
            "Key": "InstanceIds",
            "Values": [
                "i-02573cafcfEXAMPLE",
                "i-0471e04240EXAMPLE"
            ]
        }
    ],
    "Name": "My-Maintenance-Window-Target"
}
```

3. Die Option `start-date` erlaubt Ihnen, die Aktivierung eines Wartungsfensters bis zu einem angegebenen künftigen Zeitpunkt zu verzögern. Die Option `end-date` erlaubt Ihnen, ein in der Zukunft liegendes Datum sowie eine Uhrzeit festzulegen, nach dem das Wartungsfenster nicht mehr ausgeführt wird. Geben Sie die Optionen im erweiterten ISO-8601-Format an.

Führen Sie den folgenden Befehl aus, um ein Datum oder eine Zeitspanne für die regelmäßig geplanten Wartungsfenster-Ausführungen anzugeben.

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. Führen Sie den folgenden Befehl aus, um eine Run Command-Aufgabe zu aktualisieren.

> **Tip**
>
> Wenn es sich bei Ihrem Ziel um eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance für Windows Server handelt, ändern Sie `df` auf `ipconfig` und `AWS-RunShellScript` auf `AWS-RunPowerShellScript` im folgenden Befehl.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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. Passen Sie den folgenden Befehl aus und führen Sie ihn aus, um eine Lambda-Aufgabe zu aktualisieren.

   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"
   ```

   Das System gibt unter anderem folgende Informationen zurück

```
{
    "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. Wenn Sie eine Step Functions Functions-Aufgabe aktualisieren, passen Sie sie an und führen Sie den folgenden Befehl aus, um sie zu aktualisieren 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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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. Führen Sie den folgenden Befehl aus, um ein Ziel von einem Wartungsfenster abzumelden. In diesem Beispiel wird der `safe`-Parameter verwendet, um zu bestimmen, ob beliebige Aufgaben auf das Ziel verweisen und es sicher abgemeldet werden kann.

   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
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   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. Führen Sie den folgenden Befehl aus, um ein Ziel auch dann von einem Wartungsfenster abzumelden, wenn eine Aufgabe auf das Ziel verweist. Sie können den Abmeldevorgang mit dem `no-safe`-Parameter erzwingen.

   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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
}
```

9. Führen Sie den folgenden Befehl aus, um eine Run Command-Aufgabe zu aktualisieren. Dieses Beispiel verwendet einen Parameter Store Systems Manager-Parameter mit dem Namen `UpdateLevel` und der folgenden Formatierung: `{{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}}'}}"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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. Führen Sie den folgenden Befehl aus, um eine Automatisierungsaufgabe so zu aktualisieren, dass `WINDOW_ID`-Parameter und `WINDOW_TASK_ID`-Parameter als `task-invocation-parameters`-Parameter angegeben werden:

    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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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: Löschen eines Wartungsfensters (AWS CLI)

Um ein in diesen Tutorials erstelltes Wartungsfenster zu löschen, führen Sie den folgenden Befehl aus.

```
aws ssm delete-maintenance-window --window-id "mw-0c50858d01EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

Anleitungen zu Wartungsfenstern

Die Anleitungen in diesem Abschnitt zeigen, wie Sie ein AWS Systems Manager-Wartungsfenster mithilfe der AWS Command Line Interface (AWS CLI)- oder der Systems Manager-Konsole erstellen. Das Wartungsfenster, das Sie erstellen, aktualisiert SSM Agent auf verwalteten Knoten.

Inhalt

- Anleitung: Erstellen eines Wartungsfensters zum Aktualisieren von SSM Agent (AWS CLI)
- Walkthrough: Erstellen eines Wartungsfensters zum automatischen Aktualisieren von SSM Agent (Konsole)
- Walkthrough: Erstellen eines Wartungsfensters für das Einspielen von Patches (Konsole)

Sie können auch Beispiele für Befehle in der Systems Manager AWS CLI Reference finden.

Anleitung: Erstellen eines Wartungsfensters zum Aktualisieren von SSM Agent (AWS CLI)

Die folgende Walkthrough zeigt Ihnen, wie Sie mit der AWS Command Line Interface (AWS CLI) ein AWS Systems Manager-Wartungsfenster erstellen. In der Anleitung wird auch beschrieben, wie Sie Ihre verwalteten Knoten als Ziele anmelden und eine Systems-Manager-Run Command-Aufgabe für die Aktualisierung des SSM Agent anmelden.

Bevor Sie beginnen

Bevor Sie die folgenden Schritte ausführen können, müssen Sie entweder über Administratorrechte auf den Knoten verfügen, die Sie konfigurieren möchten, oder Sie müssen über die entsprechenden Berechtigungen in AWS Identity and Access Management (IAM) verfügen. Überprüfen Sie darüber hinaus, dass mindestens ein verwalteter Knoten für Linux oder Windows Server ausgeführt wird, der für Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung konfiguriert ist. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

Themen

- Schritt 1: Erste Schritte
- Schritt 2: Erstellen des Wartungsfensters
- Schritt 3: Registrieren von Wartungsfensterzielen (AWS CLI)
- Schritt 4: Registrieren einer Run Command-Aufgabe für das Wartungsfenster zum Aktualisieren von SSM Agent

Schritt 1: Erste Schritte

So führen Sie Befehle mithilfe der AWS CLI aus

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Überprüfen Sie, ob einen Knoten als Ziel für ein Wartungsfenster registriert werden kann.

   Führen Sie den folgenden Befehl aus, um zu sehen, welche Knoten online sind.

```
aws ssm describe-instance-information --query "InstanceInformationList[*]"
```

Verwenden Sie den folgenden Befehl, um weitere Details zu einem bestimmten Knoten anzuzeigen.

```
aws ssm describe-instance-information --instance-information-filter-list
 key=InstanceIds,valueSet=instance-id
```

Schritt 2: Erstellen des Wartungsfensters

Erstellen Sie anhand der folgenden Schritte ein Wartungsfenster und geben die grundlegenden Optionen, wie z. B. Zeitplan und Dauer, an.

Erstellen eines Wartungsfensters (AWS CLI)

1. Öffnen Sie die AWS CLI und führen Sie die folgenden Befehle aus, um ein Wartungsfenster zu erstellen, das wöchentlich am Sonntag um 02:00 Uhr in der pazifische Zeitzone der USA mit einer Stunde Ausfall ausgeführt wird.

   Linux & macOS

   ```
   aws ssm create-maintenance-window \
       --name "My-First-Maintenance-Window" \
       --schedule "cron(0 2 ? * SUN *)" \
       --duration 2 \
       --schedule-timezone "America/Los_Angeles" \
       --cutoff 1 \
       --no-allow-unassociated-targets
   ```

   Windows

   ```
   aws ssm create-maintenance-window ^
       --name "My-First-Maintenance-Window" ^
       --schedule "cron(0 2 ? * SUN *)" ^
       --duration 2 ^
       --schedule-timezone "America/Los_Angeles" ^
       --cutoff 1 ^
       --no-allow-unassociated-targets
   ```

Weitere Informationen zum Erstellen von Cron-Ausdrücken für den `schedule`-Parameter finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Eine Beschreibung des Verhältnisses zwischen den verschiedenen zeitplanbezogenen Optionen für Wartungsfenster finden Sie unter Wartungsfenster-Optionen für Planung und aktive Zeiträume.

Weitere Informationen zum Arbeiten mit der `--schedule`-Option finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

2. Führen Sie den folgenden Befehl aus, um dieses und alle anderen in Ihrem AWS-Konto erstellten Wartungsfenster in Ihrer aktuellen AWS-Region anzuzeigen.

```
aws ssm describe-maintenance-windows
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "WindowIdentities": [
        {
            "Cutoff": 1,
            "Name": "My-First-Maintenance-Window",
            "NextExecutionTime": "2019-02-03T02:00-08:00",
            "Enabled": true,
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Duration": 2
        }
    ]
}
```

Schritt 3: Registrieren von Wartungsfensterzielen (AWS CLI)

Führen Sie die folgenden Schritte aus, um ein Ziel für das Wartungsfenster zu registrieren, das Sie in Schritt 2 erstellt haben. Durch die Registrierung eines Ziels geben Sie an, welche Knoten aktualisiert werden sollen.

So registrieren Sie Wartungsfensterziele (AWS CLI)

1. Führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm register-target-with-maintenance-window \
       --window-id "mw-0c50858d01EXAMPLE" \
       --target "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" \
       --resource-type "INSTANCE"
   ```

   Windows

   ```
   aws ssm register-target-with-maintenance-window ^
       --window-id "mw-0c50858d01EXAMPLE" ^
       --target "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" ^
       --resource-type "INSTANCE"
   ```

   Das System gibt unter anderem folgende Informationen zurück, einschließlich einer Wartungsfenster-Ziel-ID. Kopieren oder notieren Sie sich den `WindowTargetId`-Wert. Sie benötigen diese ID im nächsten Schritt, um eine Aufgabe für dieses Wartungsfenster zu registrieren.

   ```
   {
      "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
   }
   ```

   Alternative Befehle

   Verwenden Sie den folgenden Befehl, um mehrere verwaltete Knoten anzumelden.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" \
    --resource-type "INSTANCE"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" ^
    --resource-type "INSTANCE"
```

Verwenden Sie den folgenden Befehl, um Knoten mithilfe von Tags anzumelden.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --targets "Key=tag:Environment,Values=Prod" "Key=tag:Role,Values=Web" \
    --resource-type "INSTANCE"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --targets "Key=tag:Environment,Values=Prod" "Key=tag:Role,Values=Web" ^
    --resource-type "INSTANCE"
```

2. Führen Sie den folgenden Befehl aus, um die Ziele für ein Wartungsfenster anzuzeigen.

```
aws ssm describe-maintenance-window-targets --window-id "mw-0c50858d01EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Targets": [
        {
```

```
            "ResourceType": "INSTANCE",
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "i-02573cafcfEXAMPLE"
                    ],
                    "Key": "InstanceIds"
                }
            ],
            "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
        },
        {
            "ResourceType": "INSTANCE",
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "Prod"
                    ],
                    "Key": "tag:Environment"
                },
                {
                    "Values": [
                        "Web"
                    ],
                    "Key": "tag:Role"
                }
            ],
            "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
        }
    ]
}
```

Schritt 4: Registrieren einer Run Command-Aufgabe für das Wartungsfenster zum Aktualisieren von SSM Agent

Gehen Sie wie folgt vor, um eine Run Command-Aufgabe für das Wartungsfenster zu registrieren, das Sie in Schritt 2 erstellt haben. Die Run Command-Aufgabe aktualisiert den SSM Agent auf den registrierten Zielen.

So registrieren Sie eine Run Command-Aufgabe für ein Wartungsfenster zum Aktualisieren von SSM Agent (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um eine Run Command-Aufgabe mithilfe des `WindowTargetId`-Werts aus Schritt 3 für das Wartungsfenster zu registrieren. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen. Die Aufgabe aktualisiert den SSM Agent mithilfe des `AWS-UpdateSSMAgent`-Dokuments.

   Linux & macOS

   ```
   aws ssm register-task-with-maintenance-window \
       --window-id "mw-0c50858d01EXAMPLE" \
       --task-arn "AWS-UpdateSSMAgent" \
       --name "UpdateSSMAgent" \
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    \
       --service-role-arn "arn:aws:iam:account-id:role/MW-Role" \
       --task-type "RUN_COMMAND" \
       --max-concurrency 1 --max-errors 1 --priority 10
   ```

   Windows

   ```
   aws ssm register-task-with-maintenance-window ^
       --window-id "mw-0c50858d01EXAMPLE" ^
       --task-arn "AWS-UpdateSSMAgent" ^
       --name "UpdateSSMAgent" ^
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    ^
       --service-role-arn "arn:aws:iam:account-id:role/MW-Role" ^
       --task-type "RUN_COMMAND" ^
       --max-concurrency 1 --max-errors 1 --priority 10
   ```

   > **Note**
   >
   > Wenn die Ziele, die Sie im vorherigen Schritt registriert haben, unter Windows Server 2012 R2 oder früher ausgeführt werden, müssen Sie das `AWS-UpdateEC2Config`-Dokument verwenden.

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

2. Führen Sie den folgenden Befehl aus, um alle registrierten Aufgaben für ein Wartungsfenster aufzuführen.

```
aws ssm describe-maintenance-window-tasks --window-id "mw-0c50858d01EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Tasks": [
        {
            "ServiceRoleArn": "arn:aws:iam::111122223333:role/MW-Role",
            "MaxErrors": "1",
            "TaskArn": "AWS-UpdateSSMAgent",
            "MaxConcurrency": "1",
            "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
            "TaskParameters": {},
            "Priority": 10,
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Values": [
                        "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
                    ],
                    "Key": "WindowTargetIds"
                }
            ],
            "Name": "UpdateSSMAgent"
        }
    ]
}
```

Walkthrough: Erstellen eines Wartungsfensters zum automatischen Aktualisieren von SSM Agent (Konsole)

Die folgende exemplarische Vorgehensweise zeigt Ihnen, wie Sie mit der AWS Systems Manager Konsole ein Wartungsfenster erstellen. In der Anleitung wird auch beschrieben, wie Sie Ihre verwalteten Knoten als Ziele anmelden und eine Systems-Manager-Run Command-Aufgabe für die Aktualisierung des SSM Agent anmelden.

Bevor Sie beginnen

Bevor Sie das folgende Verfahren abschließen, müssen Sie entweder über Administratorrechte für die Knoten verfügen, die Sie konfigurieren möchten, oder Ihnen müssen die entsprechenden Berechtigungen in AWS Identity and Access Management (IAM) erteilt worden sein. Überprüfen Sie darüber hinaus, dass mindestens ein verwalteter Knoten für Linux oder Windows Server in einer Hybrid- und Multi-Cloud-Umgebung ausgeführt wird, die für Systems Manager konfiguriert ist. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

Themen

- Schritt 1: Erstellen des Wartungsfensters (Konsole)
- Schritt 2: Registrieren von Wartungsfensterzielen (Konsole)
- Schritt 3: Registrieren einer Run Command-Aufgabe für das Wartungsfenster zum Aktualisieren von SSM Agent (Konsole)

Schritt 1: Erstellen des Wartungsfensters (Konsole)

So erstellen Sie ein Wartungsfenster (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus.
4. Geben Sie im Feld Name einen aussagekräftigen Namen ein, an dem Sie dieses Wartungsfenster erkennen können.
5. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein.
6. Wählen Sie Allow unregistered targets (Nicht registrierte Ziele erlauben), wenn Sie erlauben möchten, dass eine Wartungsfensteraufgabe auf verwalteten Knoten ausgeführt wird, obwohl

diese Knoten nicht als Ziele registriert wurden. Falls Sie diese Option wählen, können Sie die nicht registrierten Knoten (nach Knoten-ID) auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.

Sollten Sie diese Option nicht wählen, müssen Sie die zuvor registrierten Ziele auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.

7. Geben Sie mithilfe einer der drei Planungsoptionen einen Zeitplan für das Wartungsfenster an.

   Weitere Informationen zum Erstellen von CRON-/Rate-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

8. Geben Sie unter Duration (Dauer) die Anzahl der Stunden ein, die das Wartungsfenster ausgeführt werden soll.

9. Geben Sie unter Stop initiating tasks (Initiieren von Aufgaben beenden) die Anzahl der Stunden für den Zeitpunkt vor dem Ende des Wartungsfensters an, ab dem vom System keine neuen auszuführenden Aufgaben mehr geplant werden sollen.

10. (Optional) Geben Sie unter Window start date - optional (Fenster-Startdatum (optional)) ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an. Dies ist für den Zeitpunkt erforderlich, an dem das Wartungsfenster aktiviert werden soll. Auf diese Weise können Sie die Aktivierung des Wartungsfensters bis zum angegebenen künftigen Zeitpunkt verzögern.

    > **Note**
    >
    > Sie können kein Startdatum und keine Startzeit angeben, die in der Vergangenheit liegen.

11. (Optional) Geben Sie unter Window end date - optional ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an. Dies ist für den Zeitpunkt erforderlich, an dem das Wartungsfenster deaktiviert werden soll. Auf diese Weise können Sie ein in der Zukunft liegendes Datum sowie eine Uhrzeit festlegen, nach dem das Wartungsfenster nicht mehr ausgeführt wird.

12. (Optional) Geben Sie unter Schedule time zone - optional (geplante Zeitzone) im Internet Assigned Numbers Authority(IANA)-Format die Zeitzone an, auf der die geplanten Wartungsfenster-Ausführungen basieren sollen. Zum Beispiel: "America/Los_Angeles", "etc/UTC", oder "Asia/Seoul".

    Weitere Informationen zu gültigen Formaten finden Sie unter Time Zone Database (Zeitzonendatenbank) auf der IANA-Website.

13. (Optional) Weisen Sie im Abschnitt Manage tags (Tags verwalten) dem Wartungsfenster ein oder mehrere Tag-Schlüsselname-Wert-Paare zu.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise ein Wartungsfenster mit Tags versehen, um die Aufgabentypen, die darüber ausgeführt werden, die Arten der Ziele sowie die Umgebung, in der es ausgeführt wird, zu identifizieren. In diesem Fall könnten Sie z.B. die folgenden Schlüsselname-Wert-Paare angeben:

    - `Key=TaskType,Value=AgentUpdate`
    - `Key=OS,Value=Windows`
    - `Key=Environment,Value=Production`

14. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus. Das System leitet Sie zur Seite „Maintenance Window“ (Wartungsfenster) zurück. Der Status des soeben erstellten Wartungsfensters lautet Enabled (Aktiviert).

Schritt 2: Registrieren von Wartungsfensterzielen (Konsole)

Führen Sie die folgenden Schritte aus, um ein Ziel für das Wartungsfenster zu registrieren, das Sie in Schritt 1 erstellt haben. Durch die Registrierung eines Ziels geben Sie an, welche Knoten aktualisiert werden sollen.

So weisen Sie einem Wartungsfenster Ziele zu (Konsole)

1. Wählen Sie in der Wartungsfensterliste das soeben erstellte Wartungsfenster aus.
2. Wählen Sie Actions (Aktionen) und anschließend Register targets (Ziele registrieren) aus.
3. (Optional) Geben Sie im Feld Target Name (Zielname) einen Namen für das Ziel ein.
4. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein.
5. (Optional) Geben Sie im Feld Owner information (Eigentümerinformationen), Ihren Namen oder ihr Arbeits-Alias ein. Besitzerinformationen sind in allen EventBridge Amazon-Ereignissen enthalten, die während der Ausführung von Aufgaben für diese Ziele in diesem Wartungsfenster ausgelöst werden.

   Informationen EventBridge zur Überwachung von Systems Manager Manager-Ereignissen finden Sie unterÜberwachung von Systems Manager-Ereignissen mit Amazon EventBridge.

6. Wählen Sie im Bereich Targets (Ziele) eine der in der folgenden Tabelle beschriebenen Optionen.

Option	Beschreibung
Specify instance tags (Instance-Tags angeben)	Geben Sie unter Specify instance tags (Instance-Tags angeben) einen oder mehrere Tag-Schlüssel und (optional) Werte an, die den verwalteten Knoten in Ihrem Konto hinzugefügt wurden oder werden. Wenn das Wartungsfenster ausgeführt wird, versucht das Programm, Aufgaben auf allen verwaltet en Knoten auszuführen, denen diese Tags hinzugefügt wurden. Wenn Sie mehr als einen Tag-Schlüssel angeben, muss ein Knoten mit allen Tag-Schlüsseln und -Werten markiert werden, die Sie für die Aufnahme in die Zielgruppe angeben.

Option	Beschreibung
Manuelles Auswählen von Knoten	Aktivieren Sie in der Liste das Kontrollk ästchen für jeden Knoten, den Sie für das Wartungsfenster-Ziel aufnehmen möchten. Die Liste enthält alle Knoten in Ihrem Konto, die für die Verwendung mit Systems Manager konfiguriert sind. Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembe handlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung. Informationen zu Edge-Geräten, On-Premis es-Servern und virtuellen Maschinen (VMs) finden Sie unter Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebu ngen

Option	Beschreibung
Eine Ressourcengruppe auswählen	Wählen Sie für Resource group (Ressourcengruppe) den Namen einer vorhandenen Ressourcengruppe in Ihrem Konto aus der Liste aus. Weitere Informationen zum Erstellen von und Arbeiten mit Ressourcengruppen finden Sie unter den folgenden Themen: • Was sind Ressourcengruppen? im AWS Resource Groups -Benutzerhandbuch • Ressourcengruppen und Tagging für AWS im AWS News Blog Wählen Sie für Resource types (Ressourcentypen) bis zu fünf verfügbare Ressourcentypen aus oder wählen Sie All resource types (Alle Ressourcentypen). Wenn die Aufgaben, die Sie dem Wartungsfenster zugeordnet haben, für einen der dem Ziel hinzugefügten Ressourcentypen nicht für geeignet sind, meldet das System möglicherweise einen Fehler. Auch wenn ein solcher Fehler gemeldet wird, werden Aufgaben, für die ein unterstützter Ressourcentyp gefunden wurde, dennoch ausgeführt. Nehmen Sie beispielsweise an, Sie fügen diesem Ziel die folgenden Ressourcentypen hinzu: • `AWS::S3::Bucket` • `AWS::DynamoDB::Table` • `AWS::EC2::Instance`

Option	Beschreibung
	Wenn Sie dem Wartungsfenster später Aufgaben hinzufügen, nehmen Sie nur Aufgaben auf, die Aktionen für Knoten durchführen, wie z. B. das Anwenden einer Patch-Baseline oder das Neustarte n eines Knotens. Möglicherweise wird im Protokoll Wartungsfensterprotokoll ein Fehler gemeldet, dass keine Amazon Simple Storage Service (Amazon S3)-Buckets oder Amazon DynamoDB-Tabellen gefunden wurden. Das Wartungsfenster führt jedoch weiterhin Aufgaben auf den Knoten in Ihrer Ressourcengruppe aus.

7. Wählen Sie Register target.

Schritt 3: Registrieren einer Run Command-Aufgabe für das Wartungsfenster zum Aktualisieren von SSM Agent (Konsole)

Gehen Sie wie folgt vor, um eine Run Command-Aufgabe für das Wartungsfenster zu registrieren, das Sie in Schritt 1 erstellt haben. Die Run Command-Aufgabe aktualisiert den SSM Agent auf den registrierten Zielen.

So weisen Sie einem Wartungsfenster Aufgaben zu (Konsole)

1. Wählen Sie in der Wartungsfensterliste das soeben erstellte Wartungsfenster aus.
2. Wählen Sie Actions (Aktionen) und anschließend Register Run command Aufgabe (Run command-Aufgabe registrieren) aus.
3. (Optional) Geben Sie unter Name, einen Namen für die Aufgabe ein, z. B. UpdateSSMAgent.
4. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein.
5. Wählen Sie im Bereich Command document (Befehlsdokument) das SSM-Befehlsdokument `AWS-UpdateSSMAgent` aus.

> **Note**
>
> Wenn die Ziele, die Sie im vorherigen Schritt registriert haben, unter Windows Server 2012 R2 oder früher ausgeführt werden, müssen Sie das `AWS-UpdateEC2Config`-Dokument verwenden.

6. Wählen Sie für Document version (Dokumentversion) die zu verwendende Dokumentversion aus.
7. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.
8. Identifizieren Sie im Abschnitt Targets (Ziele) die Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Selecting registered target groups (Registrierte Zielgruppen auswählen) oder Selecting unregistered targets (Nicht registrierte Ziele auswählen) wählen.
9. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

10. (Optional) Wählen Sie für die IAM-Servicerolle eine Rolle aus, die Systems Manager bei der Ausführung einer Aufgabe im Wartungsfenster annehmen soll.

Wenn Sie keinen ARN für eine Servicerolle angeben, verwendet Systems Manager eine dienstverknüpfte Rolle in Ihrem Konto. Wenn in Ihrem Konto keine geeignete serviceverknüpfte Rolle für Systems Manager vorhanden ist, wird sie erstellt, wenn die Aufgabe erfolgreich registriert wurde.

> **Note**
>
> Um die Sicherheit zu verbessern, empfehlen wir dringend, eine benutzerdefinierte Richtlinie und eine benutzerdefinierte Servicerolle für die Ausführung Ihrer Aufgaben im Wartungsfenster zu erstellen. Die Richtlinie kann so gestaltet werden, dass sie nur die Berechtigungen gewährt, die für Ihre speziellen Wartungsfensteraufgaben erforderlich sind. Weitere Informationen finden Sie unter Konfigurieren Sie mit der Konsole Berechtigungen für Wartungsfenster.

11. (Optional) Führen Sie für Output options (Optionen für die Ausgabe) nun einen der folgenden Schritte aus:

    - Aktivieren Sie das Kontrollkästchen Enable writing to S3 (Schreiben in S3 aktivieren), um die Befehlsausgabe in einer Datei zu speichern. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

      > **Note**
      >
      > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket gewähren, sind die des Instance-Profils, das dem Knoten zugewiesen ist, und nicht die des Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das dem Knoten zugeordnete Instance-Profil über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

    - Aktivieren Sie das Kontrollkästchen CloudWatch Ausgabe, um die vollständige Ausgabe in Amazon CloudWatch Logs zu schreiben. Geben Sie den Namen einer CloudWatch Logs-Protokollgruppe ein.

12. Im Bereich SNS-Benachrichtigungen (SNS notifications) können Sie Systems Manager erlauben, Benachrichtigungen über Befehlsstatus mit Amazon Simple Notification Service (Amazon SNS) zu senden. Wenn Sie diese Option aktivieren, müssen Sie Folgendes angeben:

a. Die IAM-Rolle zum Starten von Amazon SNS-Benachrichtigungen.

b. Das zu verwendende Amazon SNS-Thema.

c. Die spezifischen Ereignistypen, über die Sie benachrichtigt werden möchten.

d. Den Benachrichtigungstyp, den Sie erhalten möchten, wenn sich der Status eines Befehls ändert. Wenn Befehle an mehrere Knoten gesendet wurden, wählen Sie die Option Invocation (Aufruf) aus, um eine Benachrichtigung auf Basis von Aufrufen (pro Knoten) zu erhalten, wenn sich der Status eines jeden Aufrufs ändert.

13. Im Abschnitt Parameter haben Sie die Möglichkeit, eine bestimmte Version von SSM Agent zu installieren, oder Sie können festlegen, dass der SSM Agent-Service auf eine ältere Version zurückgesetzt wird. Bei dieser Anleitung haben wir jedoch keine Version angegeben. Daher wird SSM Agent auf die neueste Version aktualisiert.
14. Wählen Sie Register run command task.

Walkthrough: Erstellen eines Wartungsfensters für das Einspielen von Patches (Konsole)

> **Important**
>
> Sie können dieses ältere Thema weiterhin zum Erstellen eines Wartungsfensters zum Patchen verwenden. Wir empfehlen jedoch, stattdessen eine Patch-Richtlinie zu verwenden. Weitere Informationen finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien und Patch Manager Patching-Konfiguration der Organisation.

Um die Auswirkungen auf die Verfügbarkeit Ihres Servers zu minimieren, empfehlen wir, ein Wartungsfenster zu konfigurieren, um die Patches dann einzuspielen, wenn der Geschäftsbetrieb dadurch nicht unterbrochen wird. Weitere Informationen über Wartungsfenster finden Sie unter AWS Systems Manager Maintenance Windows.

Sie müssen Rollen und Berechtigungen fürMaintenance Windows, eine Fähigkeit von, konfigurieren AWS Systems Manager, bevor Sie mit diesem Verfahren beginnen. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

So erstellen Sie ein Wartungsfenster für das Einspielen von Patches

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus.
4. Geben Sie im Feld Name einen Namen ein, aus dem hervorgeht, dass das Wartungsfenster für das Einspielen von kritischen und wichtigen Updates verwendet wird.
5. Geben Sie im Feld Description (Beschreibung) eine Beschreibung ein.
6. Wählen Sie Allow unregistered targets (Nicht registrierte Ziele erlauben), wenn Sie erlauben möchten, dass eine Wartungsfensteraufgabe auf verwalteten Knoten ausgeführt wird, obwohl diese Knoten nicht als Ziele registriert wurden. Falls Sie diese Option wählen, können Sie die nicht registrierten Knoten (nach Knoten-ID) auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.

   Sollten Sie diese Option nicht wählen, müssen Sie die zuvor registrierten Ziele auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.
7. Geben Sie oben im Abschnitt Schedule (Zeitplan) einen Zeitplan für das Wartungsfenster an, indem Sie eine der drei Planungsoptionen verwenden.

   Weitere Informationen zum Erstellen von CRON-/Rate-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.
8. Geben Sie unter Duration (Dauer) die Anzahl der Stunden ein, die das Wartungsfenster ausgeführt wird. Der Wert, den Sie angeben, bestimmt die spezifische Endzeit für das Wartungsfenster basierend auf dem Zeitpunkt, an dem es beginnt. Nach der resultierenden Endzeit dürfen keine Wartungsfenster-Aufgaben gestartet werden, abzüglich der Anzahl der Stunden, die Sie für Stop initiating tasks (Initiieren von Aufgaben beenden) im nächsten Schritt angeben.

   Beispiel: Wenn das Wartungsfenster um 15:00 Uhr beginnt, die Dauer drei Stunden beträgt und der Wert Stop initiating tasks (Initiieren von Aufgaben beenden) eine Stunde beträgt, können nach 17:00 Uhr keine Wartungsfenster-Aufgaben gestartet werden.
9. Geben Sie unter Stop initiating tasks (Initiieren von Aufgaben beenden) die Anzahl der Stunden für den Zeitpunkt vor dem Ende des Wartungsfensters an, ab dem vom System keine neuen auszuführenden Aufgaben mehr geplant werden sollen.

10. (Optional) Geben Sie unter Start date (optional) (Startdatum (optional)) ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an. Dies ist für den Zeitpunkt erforderlich, an dem das Wartungsfenster aktiviert werden soll. Auf diese Weise können Sie die Aktivierung des Wartungsfensters bis zum angegebenen künftigen Zeitpunkt verzögern.
11. (Optional) Geben Sie unter End date (optional) (Enddatum (optional)) ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an. Dies ist für den Zeitpunkt erforderlich, an dem das Wartungsfenster deaktiviert werden soll. Auf diese Weise können Sie ein in der Zukunft liegendes Datum sowie eine Uhrzeit festlegen, nach dem das Wartungsfenster nicht mehr ausgeführt wird.
12. (Optional) Geben Sie unter Time zone (optional) (Zeitzone (optional)) im Internet Assigned Numbers Authority(IANA)-Format die Zeitzone an, auf der die geplanten Wartungsfenster-Ausführungen basieren sollen. Zum Beispiel: "America/Los_Angeles", "etc/UTC", oder "Asia/Seoul".

    Weitere Informationen zu gültigen Formaten finden Sie unter Time Zone Database (Zeitzonendatenbank) auf der IANA-Website.
13. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus.
14. Wählen Sie in der Liste mit den Wartungsfenstern das gerade erstellte Wartungsfenster aus und klicken Sie anschließend auf Actions (Aktionen), Register targets (Ziele registrieren).
15. (Optional) Geben Sie im Abschnitt Maintenance window target details einen Namen, eine Beschreibung und Eigentümerinformationen (Ihren Namen oder Alias) für dieses Ziel an.
16. Wählen Sie für Targets (Ziele) die Option Specifying instance tags (Instance-Tags festlegen) aus.
17. Geben Sie im Feld Instance-Tags einen Tag-Schlüssel und einen Tag-Wert ein, um die Knoten zu identifizieren, die beim Wartungsfenster angemeldet werden sollen, und wählen Sie dann Add (Hinzufügen).
18. Wählen Sie Register target. Das System erstellt ein Ziel für das Wartungsfenster.
19. Wählen Sie auf der Detailseite des von Ihnen erstellten Wartungsfensters Actions (Aktionen), Register Run command task (Ausführungsbefehlaufgabe registrieren) aus.
20. (Optional) Geben Sie im Abschnitt Maintenance window task details (Aufgabendetails für Wartungszeitraum) einen Namen und eine Beschreibung für diese Aufgabe an.
21. Wählen Sie unter Command document (Befehlsdokument) die Option `AWS-RunPatchBaseline` aus.
22. Wählen Sie für Task priority (Aufgabenpriorität) eine Priorität aus. Null (`0`) ist die höchste Priorität.

23. Wählen Sie für Targets (Ziele) unter Target by (Auswahl nach) das Wartungsfensterziel aus, das Sie zuvor erstellt haben.
24. Für Rate control (Ratenregelung):

    - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

      > **Note**
      >
      > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

    - Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.
25. (Optional) Wählen Sie für die IAM-Servicerolle eine Rolle aus, die Systems Manager bei der Ausführung einer Aufgabe im Wartungsfenster annehmen soll.

    Wenn Sie keinen ARN für eine Servicerolle angeben, verwendet Systems Manager eine dienstverknüpfte Rolle in Ihrem Konto. Wenn in Ihrem Konto keine geeignete serviceverknüpfte Rolle für Systems Manager vorhanden ist, wird sie erstellt, wenn die Aufgabe erfolgreich registriert wurde.

    > **Note**
    >
    > Um die Sicherheit zu verbessern, empfehlen wir dringend, eine benutzerdefinierte Richtlinie und eine benutzerdefinierte Servicerolle für die Ausführung Ihrer Aufgaben im Wartungsfenster zu erstellen. Die Richtlinie kann so gestaltet werden, dass sie nur die Berechtigungen gewährt, die für Ihre speziellen Wartungsfensteraufgaben erforderlich sind. Weitere Informationen finden Sie unter Konfigurieren Sie mit der Konsole Berechtigungen für Wartungsfenster.

26. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

    > **Note**
    >
    > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Dienstrolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

    Um die Ausgabe in eine Amazon CloudWatch Logs-Protokollgruppe zu streamen, wählen Sie das CloudWatch Ausgabefeld aus. Geben Sie den Namen der Protokollgruppe in das Feld ein.

27. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

28. Für Parameters (Parameter):

    - Wählen Sie in der Liste Operation (Vorgang) die Option Scan (Scannen), um nach fehlenden Patches zu suchen, oder wählen Sie Install (Installieren), um nach fehlenden Patches zu suchen und diese direkt zu installieren.
    - Sie brauchen keine Angaben für das Feld Snapshot Id (Snapshot-ID) zu machen. Das System generiert diesen Parameter automatisch und stellt ihn bereit.
    - Sie müssen nichts in das Feld Install Override List (Überschreibungsliste installieren) eingeben, es sei denn, Sie möchten, dass Patch Manager einen anderen Patch als für die Patch-Baseline angegeben verwenden soll. Weitere Informationen finden Sie unter Parametername: `InstallOverrideList`.

- Geben Sie für Reboot option (Neustart-Option) an, ob die Knoten neu gestartet werden sollen, wenn während der `Install`-Operation Patches installiert werden, oder ob Patch Manager andere Patches erkennen soll, die seit dem letzten Neustart des Knotens installiert wurden. Weitere Informationen finden Sie unter Parametername: `RebootOption`.
- (Optional) Geben Sie im Feld Comment (Kommentar) eine Verfolgungsnotiz oder Erinnerung zu diesem Befehl ein.
- Geben Sie im Feld Timeout (seconds) (Timeout (Sekunden)) die Anzahl der Sekunden ein, die das System warten soll, bis der Vorgang beendet ist, bevor er als nicht erfolgreich eingestuft wird.

29. Wählen Sie Register run command task.

Wenn die Wartungsfenster-Aufgabe abgeschlossen ist, können Sie die Details zur Patch-Compliance in der Systems Manager-Konsole auf der Seite Managed Instances (Verwaltete Instances) einsehen. Verwenden Sie dazu die Filter `AWS:PatchSummary` und `AWS:PatchCompliance` in der Filterleiste.

> **Note**
>
> Sie können die Abfrage speichern, indem Sie die URL vor der Auswahl der Filter als Bookmark speichern.

Sie können auch Informationen auf der Ebene einzelner Knoten anzeigen, indem Sie den Knoten auf der Seite Managed Instances (Verwaltete Instances) auswählen und dann die Registerkarte Patch auswählen. Sie können auch die APIs DescribePatchGroupStateund die DescribeInstancePatchStatesForPatchGruppen-APIs verwenden, um Compliance-Details einzusehen. Weitere Informationen zu Patch-Compliance-Daten finden Sie unter Info zu Patch Compliance.

Patching-Zeitpläne mithilfe von Wartungsfenstern

Nach der Konfiguration einer Patch-Baseline (und optional einer Patch-Gruppe), können Sie Patches für Ihren Knoten mithilfe eines Wartungsfensters einspielen. Ein Wartungsfenster kann die Auswirkungen bei der Serververfügbarkeit verringern, da Sie die Möglichkeit haben, eine Uhrzeit für das Einspielen der Patches festzulegen, sodass der Geschäftsbetrieb nicht unterbrochen werden muss. Wartungsfenster funktionieren wie folgt:

1. Sie erstellen ein Wartungsfenster mit einem Zeitplan für Ihre Patching-Operationen.

2. Sie wählen die Ziele für das Wartungsfenster aus, indem Sie das Tag `Patch Group` oder `PatchGroup` für den Tag-Namen angeben und einen beliebigen Wert angeben, für den Sie Amazon Elastic Compute Cloud (Amazon EC2)-Tags definiert haben, z. B. „Produktionsserver“ oder „US-EAST-PROD“. (Sie müssen `PatchGroup` ohne Leerzeichen verwenden, wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben.
3. Sie erstellen eine neue Aufgabe für das Wartungsfenster und geben für diese Aufgabe das Dokument `AWS-RunPatchBaseline` an.

Wenn Sie die Aufgabe konfigurieren, können Sie entweder Knoten scannen oder Patches scannen und auf den Knoten installieren. Wenn Sie die Knoten scannen, scannt Patch Manager, eine Funktion von AWS Systems Manager, jeden Knoten und generiert eine Liste der fehlenden Patches für Sie zum überprüfen.

Wenn Sie Patches scannen und installieren, scannt Patch Manager jeden Knoten und vergleicht die Liste der installierten Patches mit der Liste der genehmigten Patches in der Baseline. Patch Manager identifiziert fehlende Patches und lädt dann alle fehlenden und genehmigten Patches herunter und installiert sie.

Wenn Sie einen einmaligen Scan oder eine einmalige Installation ausführen möchten, um ein Problem zu beheben, können Sie Run Command für den direkten Aufruf des Dokuments `AWS-RunPatchBaseline` verwenden.

> **Important**
>
> Nach dem Installieren von Patches führt Systems Manager einen Neustart eines jeden Knotens durch. Der Neustart ist erforderlich, um sicherzustellen, dass die Patches ordnungsgemäß installiert sind, und um sicherzustellen, dass das System den Knoten nach dem Einspielen der Patches nicht in einem potenziell fehlerhaften Zustand zurücklässt. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben

Wenn Sie eine Aufgabe inMaintenance Windows, mit einer Fähigkeit von AWS Systems Manager, registrieren, geben Sie die Parameter an, die für jeden der vier Tasktypen einzigartig sind. (In CLI-Befehlen werden diese mit der `--task-invocation-parameters` Option bereitgestellt.)

Sie können auch mithilfe der Pseudoparameter-Syntax wie `{{RESOURCE_ID}}`, `{{TARGET_TYPE}}` und `{{WINDOW_TARGET_ID}}` auf bestimmte Werte verweisen. Während der Ausführung übergibt die Wartungsfenster-Aufgabe anstelle der Pseudoparameter-Platzhalter richtige Werte. Die vollständige Liste der Pseudo-Parameter, die Sie verwenden können, finden Sie weiter unten in diesem Thema unter. Unterstützte Pseudoparameter

Important

Je nach dem für die Aufgabe erforderlichen ID-Format können Sie für den Zieltyp `RESOURCE_GROUP` auswählen, ob Sie die `{{TARGET_ID}}` und `{{RESOURCE_ID}}` zum Verweisen verwenden möchten, wenn Ihre Aufgabe ausgeführt wird. `{{TARGET_ID}}` gibt den vollständigen ARN der Ressource zurück. `{{RESOURCE_ID}}` gibt wie in diesen Beispielen gezeigt nur einen kürzeren Namen oder eine kürzere ID der Ressource zurück.

- `{{TARGET_ID}}`-Format: `arn:aws:ec2:us-east-1:123456789012:instance/i-02573cafcfEXAMPLE`
- `{{RESOURCE_ID}}`-Format: `i-02573cafcfEXAMPLE`

Für Zieltyp `INSTANCE` ergeben die Parameter `{{TARGET_ID}}` und `{{RESOURCE_ID}}` nur die Instance-ID. Weitere Informationen finden Sie unter Unterstützte Pseudoparameter. `{{TARGET_ID}}`und `{{RESOURCE_ID}}` kann verwendet werden, um AWS Ressourcen-IDs nur an Automation-, Lambda- und Step Functions Functions-Aufgaben zu übergeben. Diese beiden Pseudo-Parameter können nicht mit Run Command-Aufgaben verwendet werden.

Beispiele für Pseudoparameter

Angenommen, Ihre Payload für eine AWS Lambda Aufgabe muss anhand ihrer ID auf eine Instanz verweisen.

Unabhängig davon, ob Sie ein Wartungsfensterziel `INSTANCE` oder `RESOURCE_GROUP` verwenden, kann dies mit dem `{{RESOURCE_ID}}`-Pseudoparameter erreicht werden. Zum Beispiel:

```
"TaskArn": "arn:aws:lambda:us-east-2:111122223333:function:SSMTestFunction",
    "TaskType": "LAMBDA",
    "TaskInvocationParameters": {
        "Lambda": {
            "ClientContext": "ew0KICAi--truncated--0KIEXAMPLE",
            "Payload": "{ \"instanceId\": \"{{RESOURCE_ID}}\" }",
            "Qualifier": "$LATEST"
        }
    }
```

Wenn Ihre Lambda-Aufgabe zusätzlich zu Amazon Elastic Compute Cloud (Amazon EC2)-Instances für einen anderen unterstützten Zieltyp, z. B. eine Amazon DynamoDB-Tabelle, ausgeführt werden soll, kann dieselbe Syntax verwendet werden und `{{RESOURCE_ID}}` ergibt nur den Namen der Tabelle. Wenn Sie jedoch den vollständigen ARN der Tabelle benötigen, verwenden Sie `{{TARGET_ID}}`, wie im folgenden Beispiel gezeigt.

```
"TaskArn": "arn:aws:lambda:us-east-2:111122223333:function:SSMTestFunction",
    "TaskType": "LAMBDA",
    "TaskInvocationParameters": {
        "Lambda": {
            "ClientContext": "ew0KICAi--truncated--0KIEXAMPLE",
            "Payload": "{ \"tableArn\": \"{{TARGET_ID}}\" }",
            "Qualifier": "$LATEST"
        }
    }
```

Dieselbe Syntax funktioniert, wenn Sie auf Instances oder andere Ressourcentypen abzielen. Wenn einer Ressourcengruppe mehrere Ressourcentypen hinzugefügt wurden, wird die Aufgabe für jede der entsprechenden Ressourcen ausgeführt.

> **Important**
>
> Nicht alle Ressourcentypen, die möglicherweise in eine Ressourcengruppe einbezogen werden, ergeben einen Wert für den `{{RESOURCE_ID}}`-Parameter. Eine Liste der unterstützten Ressourcentypen finden Sie unter Unterstützte Pseudoparameter.

Ein weiteres Beispiel: Um eine Automation-Aufgabe auszuführen, die Ihre EC2-Instances beendet, geben Sie das Systems Manager-Dokument (SSM-Dokument) `AWS-StopEC2Instance` als `TaskArn`-Wert an und verwenden Sie den Pseudoparameter `{{RESOURCE_ID}}`:

```
"TaskArn": "AWS-StopEC2Instance",
    "TaskType": "AUTOMATION"
    "TaskInvocationParameters": {
        "Automation": {
            "DocumentVersion": "1",
            "Parameters": {
                "instanceId": [
                    "{{RESOURCE_ID}}"
                ]
            }
        }
    }
```

Um eine Automatisierungsaufgabe auszuführen, die einen Snapshot eines Amazon Elastic Block Store (Amazon EBS)-Volumes kopiert, geben Sie das `AWS-CopySnapshot`-SSM-Dokument als `TaskArn`-Wert an und verwenden den Pseudoparameter „`{{RESOURCE_ID}}`“:

```
"TaskArn": "AWS-CopySnapshot",
    "TaskType": "AUTOMATION"
    "TaskInvocationParameters": {
        "Automation": {
            "DocumentVersion": "1",
            "Parameters": {
                "SourceRegion": "us-east-2",
                "targetType":"RESOURCE_GROUP",
                "SnapshotId": [
                    "{{RESOURCE_ID}}"
                ]
            }
        }
    }
```

Unterstützte Pseudoparameter

Die folgende Liste beschreibt die Pseudoparameter, die Sie mit der `{{`*`PSEUDO_PARAMETER`*`}}`-Syntax in der `--task-invocation-parameters`-Option angeben können.

- **`WINDOW_ID`**: Die ID des Ziel-Wartungsfensters.

- **`WINDOW_TASK_ID`**: Die ID der Fensteraufgabe, die ausgeführt wird.
- **`WINDOW_TARGET_ID`**: Die ID des Fensterziels, die das Ziel (die Ziel-ID) umfasst.
- **`WINDOW_EXECUTION_ID`**: Die ID der aktuellen Fensterausführung.
- **`TASK_EXECUTION_ID`**: Die ID der aktuellen Aufgabenausführung.
- **`INVOCATION_ID`**: Die ID des aktuellen Aufrufs.
- **`TARGET_TYPE`**: Der Zieltyp. Unterstützte Typen sind u. a.: `RESOURCE_GROUP` und `INSTANCE`.
- **`TARGET_ID`**:

  Wenn der angegebene Zieltyp „`INSTANCE`“ lautet, wird der Pseudoparameter „`TARGET_ID`“ durch die ID der Instance ersetzt. z. B. `i-078a280217EXAMPLE`.

  Wenn der angegebene Zieltyp „`RESOURCE_GROUP`“ lautet, ist der für die Aufgabenausführung referenzierte Wert der vollständige ARN der Ressource. Zum Beispiel: `arn:aws:ec2:us-east-1:123456789012:instance/i-078a280217EXAMPLE`. Die folgende Tabelle enthält `TARGET_ID`-Beispielwerte für bestimmte Ressourcentypen in einer Ressourcengruppe.

  > **Note**
  >
  > `TARGET_ID` wird nicht für Run Command-Aufgaben unterstützt.

Ressourcentyp	Beispiel-TARGET_ID
`AWS::CloudWatch::Alarm`	`arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyCloudWatchAlarm i-078a280217EXAMPLE`
`AWS::EC2::Instance`	`arn:aws:ec2:us-east-1:123456789012:instance/ i-078a280217EXAMPLE`
`AWS::EC2::Image`	`arn:aws:ec2:us-east-1:123456789012:i`

Ressourcentyp	Beispiel-TARGET_ID	
	mage/ami-02250b373 2EXAMPLE	
AWS::EC2::Security Group	arn:aws:ec2:us-eas t-1:123456789012:s ecurity-group/sg-c EXAMPLE	
AWS::EC2::Snapshot	arn:aws:ec2:us-eas t-1:123456789012:s napshot/snap-03866 bf003EXAMPLE	
AWS::EC2::Volume	arn:aws:ec2:us-eas t-1:123456789012:v olume/vol-0912e04d 78EXAMPLE	
AWS::DynamoDB::Table	arn:aws:dynamodb:u s-east-1:123456789 012:table/MyTable	
AWS::RDS::DBCluster	arn:aws:rds:us-eas t-2:123456789012:c luster:My-Cluster	
AWS::RDS::DBInstance	arn:aws:rds:us-eas t-1:123456789012:d b:My-SQL-Instance	
AWS::S3::Bucket	arn:aws:s3::: DOC-EXAMPLE-BUCKET	
AWS::SSM::ManagedI nstance	arn:aws:ssm:us-eas t-1:123456789012:m anaged-instance/mi -0feadcf2d9EXAMPLE	

- **`RESOURCE_ID`**: Die kurze ID eines Ressourcentyps, der in einer Ressourcengruppe enthalten ist. Die folgende Tabelle enthält `RESOURCE_ID`-Beispielwerte für bestimmte Ressourcentypen in einer Ressourcengruppe.

> **Note**
>
> `RESOURCE_ID` wird nicht für Run Command-Aufgaben unterstützt.

Ressourcentyp	Beispiel-RESOURCE_ID
`AWS::CloudWatch::Alarm`	`MyCloudWatchAlarm`
`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::DynamoDB::Table`	`MyTable`
`AWS::RDS::DBCluster`	`My-Cluster`
`AWS::RDS::DBInstance`	`My-SQL-Instance`
`AWS::S3::Bucket`	`DOC-EXAMPLE-BUCKET`
`AWS::SSM::ManagedInstance`	`mi-0feadcf2d9EXAMPLE`

> **Note**
>
> Wenn die von Ihnen angegebene AWS Ressourcengruppe Ressourcentypen enthält, die keinen `RESOURCE_ID` Wert ergeben und in der obigen Tabelle nicht aufgeführt sind, wird der `RESOURCE_ID` Parameter nicht aufgefüllt. Für diese Ressource wird weiterhin ein Ausführungsaufruf ausgeführt. Verwenden Sie in diesen Fällen stattdessen Pseudoparameter „`TARGET_ID`“, der durch den vollständigen ARN der Ressource ersetzt wird.

Wartungsfenster-Optionen für Planung und aktive Zeiträume

Wenn Sie ein Wartungsfenster erstellen, müssen Sie angeben, wie oft das Wartungsfenster ausgeführt werden soll. Verwenden Sie dazu einen Cron- oder Rate-Ausdruck. Optional können Sie einen Datumsbereich angeben, in dem das Wartungsfenster nach seinem regulären Zeitplan laufen kann, sowie eine Zeitzone, auf der dieser reguläre Zeitplan basieren soll.

Beachten Sie jedoch, dass die Zeitzonenoption und die Optionen für Start- und Enddatum voneinander unabhängig sind. Das von Ihnen angegebene Start- und Enddatum (mit oder ohne einen Versatz für Ihre Zeitzone) bestimmt ausschließlich den gültigen Zeitraum, während dem das Wartungsfenster entsprechend seinem Zeitplan ausgeführt werden kann. Die Zeitzonenoption bestimmt die internationale Zeitzone, auf dessen Basis der Wartungsfenster-Zeitplan während seines gültigen Zeitraums ausgeführt wird.

> **Note**
>
> Sie geben das Start- und Enddatum im ISO-8601-Zeitstempelformat an. Zum Beispiel: `2021-04-07T14:29:00-08:00`
> Sie geben Zeitzonen in Internet Assigned Numbers Authority (IANA)-Format an. Beispiel: `America/Chicago`, `Europe/Berlin` oder `Asia/Tokyo`.

Beispiele

- Beispiel 1: Angeben eines Startdatums für das Wartungsfenster
- Beispiel 2: Angeben eines Start- und Enddatums für das Wartungsfenster
- Beispiel 3: Erstellen eines Wartungsfensters, das nur einmal ausgeführt wird

- Beispiel 4: Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster

Beispiel 1: Angeben eines Startdatums für das Wartungsfenster

Angenommen, Sie verwenden die AWS Command Line Interface (AWS CLI) zum Erstellen eines Wartungsfensters mit den folgenden Optionen:

- `--start-date 2021-01-01T00:00:00-08:00`
- `--schedule-timezone "America/Los_Angeles"`
- `--schedule "cron(0 09 ? * WED *)"`

Zum Beispiel:

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 *)"
```

Das bedeutet, dass der erste Durchlauf des Wartungsfensters erst nach dem angegebenen Startdatum und -zeitpunkt, d. h. am Freitag, dem 1. Januar 2021, um 12:00 Uhr US-Pazifikzeit, stattfinden wird. (Diese Zeitzone liegt acht Stunden hinter der UTC-Zeit.) In diesem Fall entsprechen

das Startdatum und die Startzeit des Zeitfensters nicht dem Zeitpunkt, zu dem das Wartungsfenster zum ersten Mal läuft. Zusammen betrachtet bedeuten die Werte `--schedule-timezone` und `--schedule`, dass das Wartungsfenster jeden Mittwoch um 9:00 Uhr in der US Pacific-Zeitzone ausgeführt wird (angegeben durch "Amerika/Los Angeles" im IANA-Format). Die erste Ausführung im aktivierten Zeitraum erfolgt Mittwoch, 4. Januar 2021, um 9.00 Uhr US Pacific-Zeitzone.

Beispiel 2: Angeben eines Start- und Enddatums für das Wartungsfenster

In diesem Beispiel gehen wir davon aus, dass Sie als Nächstes ein Wartungsfenster mit diesen Optionen erstellen:

- `--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)"`

Zum Beispiel:

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)"
```

Der aktivierte Zeitraum für dieses Wartungsfenster beginnt am 1. Januar 2019 um 3:15 Uhr japanische Standardzeit. Der gültige Zeitraum für dieses Wartungsfenster endet am Sonntag, 30. Juni 2019 um 6:15 Uhr japanische Standardzeit. (Diese Zeitzone liegt neun Stunden vor der UTC-Zeit.) Zusammen betrachtet bedeuten die Werte `--schedule-timezone` und `--schedule`, dass das Wartungsfenster jeden Dienstag um 3:15 Uhr in der japanischen Standardzeitzone ausgeführt wird (angegeben durch "Asien/Tokio" im IANA-Format). Der Grund hierfür ist, dass das Wartungsfenster alle sieben Tage ausgeführt wird und am Dienstag, 1. Januar um 3:15 Uhr aktiv wird. Die letzte Ausführung erfolgt am Dienstag, 25. Juni 2019 um 3:15 Uhr japanische Standardzeit. Dies ist der letzte Dienstag bevor der aktivierte Zeitraum für das Wartungsfenster fünf Tage später endet.

Beispiel 3: Erstellen eines Wartungsfensters, das nur einmal ausgeführt wird

Jetzt erstellen Sie ein Wartungsfenster mit dieser Option:

- `--schedule "at(2020-07-07T15:55:00)"`

Zum Beispiel:

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
```

Dieses Wartungsfenster wird nur einmal ausgeführt und zwar am 7. Juli 2020 um 15:55 Uhr UTC-Zeit. Das Wartungsfenster wurde aktiviert, um bei Bedarf bis zu fünf Stunden ausgeführt zu werden, jedoch können zwei Stunden vor dem Ende des Wartungsfensters keine neuen Aufgaben mehr gestartet werden.

Beispiel 4: Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster

Jetzt erstellen Sie ein Wartungsfenster mit dieser Option:

```
--schedule-offset 2
```

Zum Beispiel:

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
```

Ein Zeitplanversatz ist die Anzahl der Tage, die nach dem über einen CRON-Ausdruck angegebenen Datum und der angegebenen Uhrzeit gewartet werden soll, bevor das Wartungsfenster ausgeführt wird.

Im vorhergegangenen Beispiel wird mit dem CRON-Ausdruck die Ausführung eines Wartungsfensters um 23.30 Uhr am dritten Dienstag jedes Monats geplant:

```
--schedule "cron(0 30 23 ? * TUE#3 *)
```

Die Einbeziehung von `--schedule-offset 2` bedeutet allerdings, dass das Wartungsfenster erst um 23.30 Uhr zwei Tage nach dem dritten Dienstag jedes Monats ausgeführt wird.

Zeitplanversätze werden nur für CRON Ausdrücke unterstützt.

Weitere Informationen

- Referenz: Cron- und Rate-Ausdrücke für System Manager
- Erstellen eines Wartungsfensters (Konsole)
- Tutorial: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI)
- CreateMaintenanceWindow in der AWS Systems Manager-API-Referenz
- create-maintenance-window im Abschnitt AWS Systems Manager der AWS CLI-Befehlsreferenz
- Zeitzonendatenbank auf der IANA-Website

Wartungsfenster-Tasks ohne Ziele registrieren

Für jedes von Ihnen erstellte Wartungsfenster können Sie eine oder mehrere Aufgaben angeben, die beim Ausführen des Wartungsfensters ausgeführt werden sollen. In den meisten Fällen müssen Sie die Ressourcen oder Ziele angeben, für die Aufgabe ausgeführt werden soll. In einigen Fällen müssen Sie Ziele jedoch nicht explizit in der Aufgabe angeben.

Ein oder mehrere Ziele für Wartungsfenster Systems Manager Run Command-Typ-Aufgaben müssen angegeben werden. Abhängig von der Eigenschaft der Aufgabe sind Ziele für andere Aufgaben-Typen im Wartungsfenster optional (Systems Manager Automation, AWS Lambda und AWS Step Functions) enthalten.

Bei den Aufgabentypen Lambda und Step Functions hängt es vom Inhalt der von Ihnen erstellten Funktion oder des Zustandsautomaten ab, ob ein Ziel erforderlich ist.

In vielen Fällen müssen Sie ein Ziel für eine Automatisierungsaufgabe nicht explizit angeben. Angenommen, Sie erstellen beispielsweise eine Automation-Aufgabe, um eine Amazon Machine Image (AMI) für Linux mit dem `AWS-UpdateLinuxAmi`-Runbook zu aktualisieren. Wenn die Aufgabe ausgeführt wird, wird AMI mit den neuesten verfügbaren Linux-Verteilungspaketen und Amazon-Software aktualisiert. Neue Instances, die aus der AMI erstellt wurden, haben diese Updates bereits

installiert. Da die ID des AMI in den Eingabeparametern für das Runbook angegeben ist, muss in der Wartungsfenster-Aufgabe kein Ziel erneut angegeben werden.

Nehmen wir an, Sie verwenden das AWS Command Line Interface (AWS CLI), um eine Wartungsfenster-Automatisierungsaufgabe zu registrieren, die das `AWS-RestartEC2Instance`-Runbook verwendet. Da der neu zu startende Knoten im `--task-invocation-parameters`-Argument angegeben wird, müssen Sie nicht auch eine `--targets`-Option angeben.

> **Note**
>
> Bei Wartungsfensteraufgaben ohne festgelegtes Ziel können Sie keine Werte für `--max-errors` und `--max-concurrency` bereitstellen. Stattdessen fügt das System den Platzhalterwert 1 ein, der in der Antwort auf Befehle wie describe-maintenance-window-tasks und get-maintenance-window-task gemeldet wird. Diese Werte wirken sich nicht auf die Ausführung Ihrer Aufgabe aus und können ignoriert werden.

Das folgende Beispiel zeigt auch, dass die `--targets`, `--max-errors` und `--max-concurrency`-Optionen für eine ziellose Wartungsfensteraufgabe weggelassen werden.

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**
>
> Für Wartungsfensteraufgaben, die vor dem 23. Dezember 2020 registriert wurden: Wenn Sie Ziele für die Aufgabe angegeben haben und eines nicht mehr erforderlich ist, können Sie diese Aufgabe aktualisieren, um die Ziele mithilfe der Systems Manager-Konsole oder des update-maintenance-window-task AWS CLI-Befehls zu aktualisieren.

Weitere Informationen

- Fehlermeldungen: „Aufgaben im Wartungsfenster ohne Ziele unterstützen keine MaxConcurrency Werte“ und „Aufgaben im Wartungsfenster ohne Ziele unterstützen MaxErrors keine Werte“

Fehlerbehebung bei Wartungsfenstern

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit Wartungsfenstern.

Themen

- Aufgabenfehler bearbeiten: Auf der Seite zur Bearbeitung einer Wartungsfensteraufgabe gibt die IAM-Rollenliste eine Fehlermeldung aus: „Wir konnten die für diese Aufgabe spezifizierte IAM-Wartungsfensterrolle nicht finden. Sie wurde möglicherweise gelöscht oder noch nicht erstellt.“
- Nicht alle Wartungsfensterziele werden aktualisiert
- Die Aufgabe schlägt mit dem Aufrufstatus der Aufgabe fehl: „Die bereitgestellte Rolle enthält nicht die richtigen SSM-Berechtigungen.“
- Aufgabe schlägt mit der Fehlermeldung „Step fails when it is validating and resolving the step inputs (Schritt schlägt fehl, wenn die Schritteingaben überprüft und gelöst werden)“ fehl.
- Fehlermeldungen: „Aufgaben im Wartungsfenster ohne Ziele unterstützen keine MaxConcurrency Werte“ und „Aufgaben im Wartungsfenster ohne Ziele unterstützen MaxErrors keine Werte“

Aufgabenfehler bearbeiten: Auf der Seite zur Bearbeitung einer Wartungsfensteraufgabe gibt die IAM-Rollenliste eine Fehlermeldung aus: „Wir konnten die für diese Aufgabe spezifizierte IAM-Wartungsfensterrolle nicht finden. Sie wurde möglicherweise gelöscht oder noch nicht erstellt.“

Problem 1: Die AWS Identity and Access Management (IAM)-Wartungsfensterrolle, die Sie ursprünglich angegeben haben, wurde gelöscht, nachdem Sie die Aufgabe erstellt haben.

Mögliche Lösung: 1) Wählen Sie eine andere IAM-Wartungsfenster-Rolle aus, falls eine solche in Ihrem Konto vorhanden ist, oder erstellen Sie eine neue und wählen Sie sie für die Aufgabe aus.

Problem 2: Wenn die Aufgabe mit der AWS Command Line Interface (AWS CLI), AWS Tools for Windows PowerShell oder einem AWS-SDK erstellt wurde, wurde möglicherweise ein nicht vorhandener IAM-Wartungsfenster-Rollenname angegeben. Beispielsweise könnte die IAM-Wartungsfensterrolle gelöscht worden sein, bevor Sie die Aufgabe erstellt haben, oder der Rollenname könnte falsch eingegeben worden sein, z. B. **`myrole`** anstelle von **`my-role`**.

Mögliche Lösung: Wählen Sie den richtigen Namen der IAM-Wartungsfensterrolle aus, die Sie verwenden möchten, oder erstellen Sie eine neue, die Sie für die Aufgabe angeben können.

Nicht alle Wartungsfensterziele werden aktualisiert

Problem: Sie stellen fest, dass die Wartungsfensteraufgaben nicht auf allen Ressourcen ausgeführt wurden, auf die Ihr Wartungsfenster abzielt. Beispiel: In den Ausführungsergebnissen des Wartungsfensters wird die Aufgabe für diese Ressource beispielsweise als fehlgeschlagen oder zeitlich abgelaufen markiert.

Lösung: Die häufigsten Gründe für das Nicht-Ausführen einer Wartungsfensteraufgabe auf einer Zielressource, sind Konnektivität und Verfügbarkeit. Zum Beispiel:

- Systems Manager hat die Verbindung zur Ressource vor oder während des Wartungsfenstervorgangs unterbrochen.
- Die Ressource war offline oder wurde während des Wartungsfenstervorgangs beendet.

Sie können warten, bis die Zeit der nächsten geplanten Wartungsfensteraufgabe für die Ressourcen ausgeführt wird. Sie können die Wartungsfensteraufgabe manuell für die Ressourcen ausführen, die nicht verfügbar waren oder offline waren.

Die Aufgabe schlägt mit dem Aufrufstatus der Aufgabe fehl: „Die bereitgestellte Rolle enthält nicht die richtigen SSM-Berechtigungen.“

Problem: Sie haben eine Wartungsfenster-Servicerolle für eine Aufgabe angegeben, aber die Aufgabe wird nicht erfolgreich ausgeführt, und der Aufgabenaufrufstatus meldet, dass „die bereitgestellte Rolle nicht die richtigen SSM-Berechtigungen enthält.“

- Solution (Lösung): In Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben stellen wir eine grundlegende Richtlinie zur Verfügung, die Sie an Ihre benutzerdefinierte Wartungsfenster-Servicerolle anhängen können. Die Richtlinie enthält die für viele Aufgabenszenarien erforderlichen Berechtigungen. Aufgrund der Vielzahl von Aufgaben, die Sie ausführen können, müssen Sie jedoch möglicherweise zusätzliche Berechtigungen in der Richtlinie für Ihre Wartungsfenster-Rolle angeben.

  Manche Automatisierungsaktionen basieren z. B. auf AWS CloudFormation-Stacks. Daher müssen Sie möglicherweise die zusätzlichen Berechtigungen `cloudformation:CreateStack`, `cloudformation:DescribeStacks` und `cloudformation:DeleteStack` in die Richtlinie für Ihre Wartungsfenster-Servicerolle aufnehmen.

  Als weiteres Beispiel benötigt das Automation-Runbook `AWS-CopySnapshot` Berechtigungen zum Erstellen eines Amazon Elastic Block Store (Amazon EBS)-Snapshots. Daher müssen Sie möglicherweise die `ec2:CreateSnapshot`-Berechtigung hinzufügen.

  Weitere Informationen zu den von AWS-Managed-Automation-Runbook benötigten Rollenberechtigungen finden Sie in den Runbook-Beschreibungen in der Referenz zu AWS Systems Manager-Automation-Runbook.

  Weitere Informationen zu den Rollenberechtigungen, die für ein AWS-verwaltetes SSM-Dokument erforderlich sind, finden Sie im Abschnitt Dokumente der Systems-Manager-Konsole.

  Informationen zu den Rollenberechtigungen, die für Step-Functions-Aufgaben, Lambda-Aufgaben, benutzerdefinierte Automation-Runbooks und SSM-Dokumente erforderlich sind, erhalten Sie beim Autor dieser Ressourcen über die Berechtigungsanforderungen.

Aufgabe schlägt mit der Fehlermeldung „Step fails when it is validating and resolving the step inputs (Schritt schlägt fehl, wenn die Schritteingaben überprüft und gelöst werden)“ fehl.

Problem: Ein Automation-Runbook oder Systems Manager-Befehlsdokument, das Sie in einer Aufgabe verwenden, erfordert, dass Sie Eingaben wie `InstanceId` oder `SnapshotId` angeben, aber ein Wert wird nicht angegeben oder nicht korrekt angegeben.

- Lösung 1: Wenn Ihr Vorgang auf eine einzelne Ressource ausgerichtet ist, z. B. ein einzelner Knoten oder ein einzelner Snapshot, geben Sie die ID in die Eingabeparameter für die Aufgabe ein.
- Lösung 2: Wenn Ihre Aufgabe auf mehrere Ressourcen ausgerichtet ist, z. B. das Erstellen von Images aus mehreren Knoten, wenn Sie das Runbook `AWS-CreateImage` verwenden, können Sie einen der Pseudoparameter verwenden, die für Wartungsfenster-Aufgaben in den Eingabeparametern unterstützt werden, um Knoten-IDs im Befehl darzustellen.

  Die folgenden Befehle registrieren eine Systems Manager Automation-Aufgabe mit einem Wartungsfenster unter Verwendung der Option AWS CLI. Der `--targets`-Wert gibt eine Ziel-ID für das Wartungsfenster an. Auch wenn der `--targets`-Parameter eine Ziel-ID des Fensters angibt, erfordern Parameter des Automatisierung-Runbooks, dass eine Knoten-ID angegeben wird. In diesem Fall verwendet der Befehl den Pseudo-Parameter `{{RESOURCE_ID}}` als `InstanceId` Wert.

  AWS CLI-Befehl:

  Linux & macOS

  Der folgende Beispielbefehl startet Amazon Elastic Compute Cloud (Amazon EC2)-Instances neu, die zur Zielgruppe des Wartungsfensters mit der ID e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE gehören.

```
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"
```

Weitere Informationen zur Arbeit mit Pseudoparametern für Wartungsfensteraufgaben finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben und Beispiele der Aufgabenregistrierung.

Fehlermeldungen: „Aufgaben im Wartungsfenster ohne Ziele unterstützen keine MaxConcurrency Werte“ und „Aufgaben im Wartungsfenster ohne Ziele unterstützen MaxErrors keine Werte“

Problem: Wenn Sie eine Run Command-Typ-Aufgabe registrieren, müssen Sie mindestens ein Ziel angeben, auf dem die Aufgabe ausgeführt werden soll. Bei anderen Aufgabentypen (Automatisierung, AWS Lambda und AWS Step Functions) sind die Ziele je nach Art der Aufgabe optional. Die Optionen `MaxConcurrency` (die Anzahl der Ressourcen, auf denen eine Aufgabe gleichzeitig ausgeführt werden soll) und `MaxErrors` (die Anzahl der Fehlschläge, nach denen die Aufgabe auf den Zielressourcen ausgeführt werden soll, bevor die Aufgabe fehlschlägt) sind für Wartungsfensteraufgaben, die keine Ziele angeben, nicht erforderlich oder werden nicht unterstützt. Das System generiert diese Fehlermeldungen, wenn für eine dieser Optionen Werte angegeben werden, wenn kein Aufgabenziel angegeben ist.

Lösung: Wenn einer dieser Fehler angezeigt wird, entfernen Sie die Werte für Parallelität und Fehlerschwellenwert, bevor Sie mit der Registrierung oder Aktualisierung der Wartungsfensteraufgabe fortfahren.

Weitere Informationen zur Ausführung von Aufgaben, die keine Ziele angeben, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren im AWS Systems Manager-Benutzerhandbuch.

AWS Systems Manager Knotenverwaltung

AWS Systems Manager bietet die folgenden Funktionen für den Zugriff, die Verwaltung und die Konfiguration Ihrer verwalteten Knoten. Ein verwalteter Knoten ist eine Maschine, die für die Verwendung mit Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung konfiguriert ist.

Themen

- AWS Systems Manager Fleet Manager
- AWS Systems Manager-Compliance
- AWS Systems Manager-Bestand
- AWS Systems Manager Hybride Aktivierungen
- AWS Systems Manager Session Manager
- AWS Systems Manager Run Command
- AWS Systems Manager State Manager
- AWS Systems Manager Patch Manager
- AWS Systems Manager Distributor

AWS Systems Manager Fleet Manager

Fleet Manager, eine Funktion von AWS Systems Manager, ist eine einheitliche Benutzeroberfläche (UI), mit der Sie Ihre Knoten, die vor Ort AWS oder vor Ort laufen, remote verwalten können. Mit Fleet Manager können Sie sich den Zustand und den Leistungsstatus Ihrer gesamten Serverflotte von einer Konsole aus ansehen. Sie können auch Daten aus einzelnen Knoten sammeln, um allgemeine Problembehandlungs- und Verwaltungsaufgaben über die Konsole auszuführen. Dies umfasst die Verbindung mit Windows-Instances über das Remote Desktop Protocol (RDP), das Anzeigen von Ordner- und Dateiinhalten, die Verwaltung der Windows-Registry, die Benutzerverwaltung des Betriebssystems und vieles mehr. Um mit Fleet Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Fleet Manager aus.

An wen richtet sich Fleet Manager?

Jeder AWS Kunde, der eine zentrale Methode zur Verwaltung seiner Knotenflotte wünscht, sollte diese Option verwendenFleet Manager.

Welche Vorteile bietet Fleet Manager meiner Organisation?

Fleet Manager bietet die folgenden Vorteile:

- Ausführen einer Vielzahl gängiger Systemverwaltungs-Aufgaben, ohne eine manuelle Verbindung zu Ihren verwalteten Knoten herstellen zu müssen.
- Verwalten von Knoten, die auf mehreren Plattformen ausgeführt werden, über eine einzige einheitliche Konsole.
- Verwalten von Knoten, auf denen verschiedene Betriebssysteme ausgeführt werden, über eine einzige einheitliche Konsole.
- Verbessern Sie die Effizienz Ihrer Systemadministration.

Über welche Features verfügt Fleet Manager?

Nachstehend sind einige der wichtigsten Features von Fleet Manager aufgelistet:

- Zugreifen auf das Red-Hat-Knowledgebase-Portal

  Greifen Sie über Ihre Red Hat Enterprise Linux (RHEL)-Instances auf Binärdateien, Wissensfreigaben und Diskussionsforen im Knowledgebase-Portal von Red Hat zu.

- Status des verwalteten Knotens

  Anzeigen, welche verwalteten Knoten `running` sind und welche `stopped` sind. Weitere Informationen zu gestoppten Instances finden Sie unter Stoppen und starten Sie Ihre Instance im Amazon EC2 EC2-Benutzerhandbuch. Bei AWS IoT Greengrass Core-Geräten können Sie sich ansehen`online`, welche Geräte den Status haben`offline`, oder den Status anzeigen. `Connection lost`

  > **Note**
  >
  > Wenn Sie Ihre verwaltete Instance vor dem 12. Juli 2021 angehalten haben, wird die `stopped`-Markierung nicht angezeigt. Um die Markierung anzuzeigen, starten und beenden Sie die Instance.

- Anzeigen von Instance-Informationen

Zeigen Sie Informationen zu den Ordner- und Dateidaten an, die auf den an Ihre verwalteten Instances angeschlossenen Volumes gespeichert sind, sowie Leistungsdaten zu Ihren Instances in Echtzeit und auf Ihren Instances gespeicherte Protokolldaten.

- Informationen über Edge-Gerät anzeigen

  Sehen Sie sich AWS IoT Greengrass den Namen des Dings für das Gerät, den SSM Agent Ping-Status und die Version und mehr an.

- Verwalten von Konten und Registrierung

  Verwalten von Betriebssystem-Benutzerkonten auf Ihren Instances und Registrieren auf Ihren Windows-Instances.

- Steuern des Zugriffs auf Features

  Steuern Sie den Zugriff auf Fleet Manager Funktionen mithilfe von AWS Identity and Access Management (IAM-) Richtlinien. Mit diesen Richtlinien können Sie steuern, welche Benutzer oder Gruppen in Ihrer Organisation verschiedene Fleet Manager-Features verwenden können und welche verwalteten Knoten sie verwalten können.

Themen

- Erste Schritte mit Fleet Manager
- Arbeiten mit Fleet Manager
- Problembehandlung bei der Verfügbarkeit verwalteter Knoten

Erste Schritte mit Fleet Manager

Bevor Sie Fleet Manager, eine Funktion von AWS Systems Manager, verwenden können, um Ihre verwalteten Knoten zu überwachen und zu verwalten, führen Sie die in den folgenden Themen beschriebenen Schritte aus.

Themen

- Schritt 1: Erstellen einer IAM-Richtlinie mit Fleet Manager-Berechtigungen
- Schritt 2: Verifizieren Sie, dass Ihre Instances und Edge-Geräte von Systems Manager verwaltet werden können

Schritt 1: Erstellen einer IAM-Richtlinie mit Fleet Manager-Berechtigungen

Um die Funktion verwenden zu können Fleet Manager AWS Systems Manager, muss Ihr AWS Identity and Access Management (IAM-) Benutzer oder Ihre Rolle über die erforderlichen Berechtigungen verfügen. Sie können eine IAM-Richtlinie erstellen, die Zugriff auf alle Fleet Manager-Features bieten, oder ändern Sie Ihre Richtlinie, um Zugriff auf die ausgewählten Features zu gewähren.

Die folgenden Beispielrichtlinien enthalten die erforderlichen Berechtigungen für alle Fleet Manager-Features und die Berechtigungen, die für Untergruppen von Funktionen erforderlich sind.

Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Themen

- Beispielrichtlinie für Fleet Manager-Administratorzugriff
- Beispielrichtlinie für schreibgeschützten Fleet Manager-Zugriff

Beispielrichtlinie für Fleet Manager-Administratorzugriff

Die folgende Richtlinie gewährt Berechtigungen für alle Fleet Manager-Features. Das bedeutet, dass ein Benutzer lokale Benutzer und Gruppen erstellen und löschen, die Gruppenmitgliedschaft für jede lokale Gruppe ändern und Windows Server Registrierungsschlüssel oder -werte ändern kann. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
{
   "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:region:account-id:servicesetting/ssm/managed-
instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::account-id: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:*:account-id:instance/*",
               "arn:aws:ssm:*:account-id:managed-instance/*",
               "arn:aws:ssm:*:account-id: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"
            ],
            "Condition":{
               "BoolIfExists":{
```

```
                "ssm:SessionDocumentAccessCheck":"true"
            }
         }
      },
      {
         "Sid":"TerminateSession",
         "Effect":"Allow",
         "Action":[
            "ssm:TerminateSession"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/aws:ssmmessages:session-id":[
                  "${aws:userid}"
               ]
            }
         }
      },
      {
         "Sid":"KMS",
         "Effect":"Allow",
         "Action":[
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "arn:aws:kms:region:account-id:key/key-name"
         ]
      }
   ]
}
```

Beispielrichtlinie für schreibgeschützten Fleet Manager-Zugriff

Die folgende Richtlinie gewährt Berechtigungen für alle schreibgeschützten Fleet Manager–Features. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
{
   "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:*:account-id:instance/*",
               "arn:aws:ssm:*:account-id:managed-instance/*",
               "arn:aws:ssm:*:account-id: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"
         ],
         "Condition":{
            "BoolIfExists":{
               "ssm:SessionDocumentAccessCheck":"true"
            }
         }
      },
      {
         "Sid":"TerminateSession",
         "Effect":"Allow",
         "Action":[
            "ssm:TerminateSession"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/aws:ssmmessages:session-id":[
                  "${aws:userid}"
               ]
            }
         }
      },
      {
         "Sid":"KMS",
         "Effect":"Allow",
         "Action":[
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "arn:aws:kms:region:account-id:key/key-name"
         ]
      }
   ]
}
```

Schritt 2: Verifizieren Sie, dass Ihre Instances und Edge-Geräte von Systems Manager verwaltet werden können

Für Amazon Elastic Compute Cloud (Amazon EC2)-Instances, AWS IoT Greengrass-Core-Geräte und On-Premises-Server, Edge-Geräte und virtuelle Maschinen (VMs), die mithilfe von Fleet Manager, einer Funktion von AWS Systems Manager, überwacht und verwaltet werden können,

müssen sie verwaltete Knoten von Systems Manager sein. Dies bedeutet, dass Ihre Knoten bestimmte Voraussetzungen erfüllen und mit dem AWS Systems Manager-Agent (SSM Agent) konfiguriert sein müssen. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

Sie können Quick Setup, eine Funktion von AWS Systems Manager, verwenden, um Ihre Amazon-EC2-Instances schnell als verwaltete Instances in einem einzelnen Konto konfigurieren zu können. Wenn Ihr Unternehmen oder Ihre Organisation AWS Organizations verwendet, können Sie auch Instances über mehrere Organisationseinheiten (OUs) und AWS-Regionen hinweg konfigurieren. Weitere Informationen zur Verwendung von Quick Setup zum Konfigurieren verwalteter Instance finden Sie unter Amazon-EC2-Host-Verwaltung.

> **Note**
>
> Für Nicht-EC2-Maschinen, die nicht auf AWS laufen, verwenden Sie eine Hybrid-Aktivierung, um die Maschine für die Verwendung mit Systems Manager in einer Hybrid- und Multi-Cloud zu konfigurieren. Weitere Informationen zu Hybrid-Aktivierungen finden Sie unter AWS Systems Manager Hybride Aktivierungen.

Arbeiten mit Fleet Manager

Sie können Fleet Manager, eine Funktion von AWS Systems Manager, verwenden, um verschiedene Aufgaben auf Ihren verwalteten Knoten von der AWS Systems Manager Konsole aus auszuführen. In den folgenden Themen werden die Funktionen beschrieben, die von Fleet Manager bereitgestellt werden.

> **Note**
>
> Die einzige unterstützte Funktion für macOS-Instances ist die Anzeige des Dateisystems.

Themen

- Mit verwalteten Knoten arbeiten
- Verwenden der Standardeinstellung für die Host-Management-Konfiguration
- Verbindung zu einer Windows Server verwalteten Instanz herstellen mit Remote Desktop
- Verwaltung von Amazon EBS-Volumes auf verwalteten Instances
- Arbeiten mit dem Dateisystem

- Überwachung der Leistung verwalteter Knoten
- Arbeiten mit Prozessen
- Protokolle auf verwalteten Knoten anzeigen
- Verwaltung von Betriebssystembenutzerkonten auf verwalteten Knoten
- Verwaltung der Windows-Registrierung auf verwalteten Knoten
- Zugriff auf das Knowledgebase-Portal von Red Hat

Mit verwalteten Knoten arbeiten

Ein verwalteter Knoten ist jede Maschine, für die konfiguriert ist AWS Systems Manager. Sie können die folgenden Maschinentypen als verwaltete Knoten konfigurieren:

- Instances von Amazon Elastic Compute Cloud (Amazon EC2)
- Server in Ihren eigenen Räumlichkeiten (On-Premises-Server)
- AWS IoT Greengrass Kerngeräte
- AWS IoT und Geräte, die nicht zu den AWS Edge-Geräten gehören
- Virtuelle Maschinen (VMs), einschließlich VMs in anderen Cloud-Umgebungen

> **Note**
>
> In der Systems-Manager-Konsole wurde jede Maschine mit dem Präfix „mi-“ mit einer Hybrid-Aktivierung als verwalteter Knoten konfiguriert. Edge-Geräte zeigen ihren AWS IoT -Objektnamen an.

AWS Systems Manager bietet eine Stufe „Standard-Instances“ und eine Stufe „Advanced Instances“. Beide unterstützen verwaltete Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung. Mit der Stufe „Standard-Instances“ können Sie maximal 1.000 Maschinen pro Person registrieren. AWS-Konto AWS-Region Wenn Sie mehr als 1 000 Maschinen in einem einzigen Konto und einer Region anmelden müssen, verwenden Sie das Advanced-Instances-Kontingent. Sie können im Advanced-Instances-Kontingent so viele verwaltete Knoten erstellen, wie Sie möchten. Alle verwalteten Knoten, die für Systems Manager konfiguriert sind, werden auf pay-per-use Basis von Preisen berechnet. Weitere Informationen über das Aktivieren des Advanced-Instances-Kontingent finden Sie unter Aktivieren des Kontingents für erweiterte Instances. Weitere Informationen über die Preise finden Sie unter AWS Systems Manager – Preise.

> **Note**
>
> - Mithilfe von Advanced Instances können Sie in einer Hybrid- und Multi-Cloud-Umgebung auch eine Verbindung zu Ihren Nicht-EC2-Knoten herstellen. AWS Systems Manager Session Manager Session Managerbietet interaktiven Shell-Zugriff auf Ihre Instances. Weitere Informationen finden Sie unter AWS Systems Manager Session Manager.
> - Das standardmäßige Instance-Kontingent gilt auch für EC2-Instances, die eine On-Premises-Aktivierung von Systems Manager verwenden (was kein übliches Szenario ist).
> - Aktivieren Sie das Advanced-Instances-Kontingent, um Anwendungen, die von Microsoft auf virtuellen Maschinen (VMs) On-Premises-Instances veröffentlicht werden, zu patchen. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Für Patch-Anwendungen, die von Microsoft auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances veröffentlicht wurden, fallen keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden.

Anzeigen von verwalteten Knoten

Wenn Ihre verwalteten Knoten nicht in der Konsole aufgeführt werden, gehen Sie wie folgt vor:

1. Stellen Sie sicher, dass die Konsole in dem Bereich geöffnet ist, in AWS-Region dem Sie Ihre verwalteten Knoten erstellt haben. Über die Liste in der oberen, rechten Ecke der Konsole können Sie zwischen dein einzelnen Regionen wechseln.
2. Überprüfen Sie, ob die Einrichtungsschritte für Ihre verwalteten Knoten den Voraussetzungen von Systems Manager entsprechen. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.
3. Stellen Sie bei Nicht-EC2-Maschinen sicher, dass Sie den Hybrid-Aktivierungsprozess abgeschlossen haben. Weitere Informationen finden Sie unter Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen.

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - Die Fleet Manager-Konsole zeigt keine Amazon-EC2-Knoten an, die beendet wurden.

- Systems Manager erfordert genaue Zeitreferenzen, um Operationen auf Ihren Maschinen auszuführen. Wenn auf Ihren verwalteten Knoten das Datum und die Uhrzeit nicht korrekt eingestellt wurden, stimmen sie möglicherweise nicht mit dem Signaturdatum Ihrer API-Anforderungen überein. Weitere Informationen finden Sie unter Anwendungsfälle und bewährte Methoden.
- Wenn Sie Tags erstellen oder bearbeiten, kann das System bis zu einer Stunde benötigen, bis Änderungen im Tabellenfilter angezeigt werden.
- Nachdem der Status eines verwalteten Knotens mindestens 30 Tage lang `Connection Lost` gewesen ist, wird der Knoten möglicherweise nicht mehr in der Fleet Manager-Konsole aufgeführt. Beheben Sie das Problem, das den Verbindungsverlust verursacht hat, um ihn wieder in die Liste aufzunehmen. Tipps zur Fehlerbehebung finden Sie unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten.

Überprüfen des Supports für Systems Manager auf einem verwalteten Knoten

AWS Config stellt AWS verwaltete Regeln bereit. Dabei handelt es sich um vordefinierte, anpassbare Regeln, AWS Config anhand derer bewertet wird, ob Ihre AWS Ressourcenkonfigurationen den gängigen bewährten Methoden entsprechen. AWS Config Zu den verwalteten Regeln gehört die Regel ec2-instance-managed-by-systems-manager. Diese Regel prüft, ob die Amazon-EC2-Instances in Ihrem Konto durch Systems Manager verwaltet werden. Weitere Informationen finden Sie unter AWS Config Managed Rules.

Erhöhen des Sicherheitsstatus auf verwalteten Knoten

Informationen zur Steigerung des Sicherheitsstatus in Bezug auf nicht autorisierte Befehle auf Root-Ebene auf Ihren verwalteten Knoten finden Sie unter Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent.

Abmelden verwalteter Knoten

Sie können verwaltete Knoten jederzeit abmelden. Wenn Sie beispielsweise mehrere Knoten mit derselben AWS Identity and Access Management (IAM-) Rolle verwalten und irgendein bösartiges Verhalten feststellen, können Sie jederzeit eine beliebige Anzahl von Computern abmelden. Informationen über das Abmelden verwalteter Knoten finden Sie unter Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung.

Themen

- Konfigurieren von Instance-Kontingenten
- Zurücksetzen von Passwörtern für verwaltete Knoten
- Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung

Konfigurieren von Instance-Kontingenten

In diesem Thema werden die Szenarien beschrieben, in denen Sie das Advanced-Instances-Kontingent aktivieren müssen.

AWS Systems Manager bietet eine Standard-Instance-Stufe und eine Advanced-Instance-Stufe für Nicht-EC2-Computer in einer Hybrid- und Multi-Cloud-Umgebung.

Sie können bis zu 1.000 standardmäßige, hybridaktivierte Knoten pro Konto und ohne zusätzliche Kosten registrieren. AWS-Region Um mehr als 1 000 Hybrid-Knoten zu registrieren, müssen Sie jedoch das Advanced-Instances-Kontingent aktivieren. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Weitere Informationen finden Sie unter AWS Systems Manager - Preisgestaltung.

Selbst mit weniger als 1 000 registrierten hybrid-aktivierten Knoten erfordern zwei weitere Szenarien des Advanced-Instances-Kontingents:

- Sie möchten Session Manager verwenden, um eine Verbindung zu Nicht-EC2-Knoten herzustellen.
- Sie möchten Anwendungen (nicht Betriebssysteme), die von Microsoft auf Nicht-EC2-Knoten veröffentlicht wurden, patchen.

> **Note**
>
> Für Patch-Anwendungen, die von Microsoft auf Amazon-EC2-Instances veröffentlicht wurden, fallen keine Gebühren an.

Detaillierte Szenarien für Advanced-Instances-Kontingent

Die folgenden Informationen enthalten Details zu den drei Szenarien, für die Sie das Advanced-Instances-Kontingent aktivieren müssen.

Szenario 1: Sie möchten mehr als 1 000 hybrid-aktivierte Knoten registrieren

Mithilfe des Standard-Instances-Kontingents können Sie maximal 1 000 Nicht-EC2-Knoten in einer Hybrid- und Multi-Cloud-Umgebung pro AWS-Region in einem bestimmten Konto ohne zusätzliche Kosten registrieren. Wenn Sie mehr als 1 000 Nicht-EC2-Knoten in einer Region anmelden müssen, müssen Sie das Advanced-Instances-Kontingent verwenden. Sie können dann so viele Maschinen für Ihre Hybrid- und Multi-Cloud-Umgebung aktivieren, wie Sie möchten. Die Gebühren für das Advanced-Instance-Kontingent basieren auf der Anzahl der Advanced-Knoten, die als von Systems Manager verwaltete Knoten aktiviert wurden, und den Stunden, in denen diese Knoten ausgeführt werden.

Für alle von Systems Manager verwalteten Knoten, die den unter Erstellen einer Hybridaktivierung beschriebenen Aktivierungsprozess verwenden, um Knoten bei Systems Manager zu registrieren, fallen Gebühren an, wenn Sie in einem bestimmten Konto mehr als 1.000 lokale Knoten in einer Region haben.

> **Note**
>
> Sie können auch vorhandene Amazon Elastic Compute Cloud (Amazon EC2)-Instances mithilfe von Hybrid-Aktivierungen von Systems Manager installieren und mit ihnen als Nicht-EC2-Instances arbeiten, z. B. zu Testzwecken. Diese zählen auch als Hybrid-Knoten. Dies ist kein übliches Szenario.

Szenario 2: Patchen von von Microsoft veröffentlichten Anwendungen auf hybrid-aktivierten Knoten

Das Advanced-Instances-Kontingent ist auch erforderlich, wenn Sie von Microsoft veröffentlichte Anwendungen auf Nicht-EC2-Knoten in einer Hybrid- und Multi-Cloud-Umgebung patchen möchten. Wenn Sie das Advanced-Instances-Kontingent aktivieren, um Microsoft-Anwendungen auf Nicht-EC2-Knoten zu patchen, fallen Gebühren für alle On-Premises-Knoten an, selbst wenn Sie weniger als 1 000 haben.

Für Patch-Anwendungen, die von Microsoft auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances veröffentlicht wurden, fallen keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden.

Szenario 3: Herstellen einer Verbindung zu hybrid-aktivierten Knoten mit Session Manager

Session Manager bietet interaktiven Shell-Zugriff auf Ihre Instances. Um über Session Manager eine Verbindung zu hybrid-aktivierten Knoten herzustellen, müssen Sie die Advanced-Instances-Kontingente aktivieren. Dann fallen Gebühren für alle hybrid-aktivierten Knoten an, auch wenn Sie weniger als 1 000 haben.

Zusammenfassung: Wann brauche ich das Advanced-Instances-Kontingent?

Anhand der folgenden Tabelle können Sie überprüfen, wann Sie das Advanced-Instances-Kontingent verwenden müssen und für welche Szenarien zusätzliche Gebühren anfallen.

Szenario	Advanced-Instances-Kontingent erforderlich?	Es fallen zusätzliche Gebühren an.
Die Anzahl der hybrid-aktivierten Knoten in meiner Region in einem bestimmten Konto beträgt mehr als 1 000.	Ja	Ja
Ich möchte Patch Manager benutzen, um von Microsoft veröffentlichte Anwendungen auf einer beliebigen Anzahl hybrid-aktivierter Knoten zu patchen, sogar weniger als 1 000.	Ja	Ja
Ich möchte Session Manager benutzen, um sich mit einer beliebigen Anzahl hybrid-aktivierter Knoten zu verbinden, sogar weniger als 1 000.	Ja	Ja
1. Die Anzahl der hybrid-aktivierten Knoten in meiner Region in einem bestimmte	Nein	Nein

Szenario	Advanced-Instances-Kontingent erforderlich?	Es fallen zusätzliche Gebühren an.
n Konto beträgt 1 000 oder weniger; und 2. Ich patche keine Microsoft-Anwendungen auf hybrid-aktivierten Knoten; und 3. Ich verbinde mich nicht mit hybrid-aktivierten Knoten mit Session Manager.		

Themen

- Aktivieren des Kontingents für erweiterte Instances
- Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances

Aktivieren des Kontingents für erweiterte Instances

AWS Systems Manager bietet eine Stufe „Standard-Instances“ und eine Stufe „Advanced-Instances“ für Nicht-EC2-Computer in einer Hybrid- und Multi-Cloud-Umgebung. Mit dem Standard-Instances-Kontingent können Sie maximal 1 000 hybridaktivierte Maschinen pro AWS-Konto pro AWS-Region registrieren. Die erweiterte Instances-Ebene ist auch erforderlich, um Patch Manager zum Patchen von von Microsoft freigegebenen Anwendungen auf Nicht-EC2-Knoten zu verwenden und mit Session Manager eine Verbindung zu Nicht-EC2-Knoten herzustellen. Weitere Informationen finden Sie unter Konfigurieren von Instance-Kontingenten.

In diesem Abschnitt wird beschrieben, wie Sie Ihre Hybrid- und Multi-Cloud-Umgebung für die Nutzung des Kontingents für erweiterte Instances konfigurieren.

Bevor Sie beginnen

Prüfen Sie die Preisdetails für erweiterte Instances. Erweiterte Instances sind auf einem verfügbar. per-use-basis Weitere Informationen finden Sie unter AWS Systems Manager -Preise.

Konfigurieren von Berechtigungen zum Aktivieren des Kontingents für erweiterte Instances

Vergewissern Sie sich, dass Sie in AWS Identity and Access Management (IAM) berechtigt sind, Ihre Umgebung von der Stufe Standard-Instances auf die Stufe Advanced-Instances umzustellen.

Sie müssen entweder die `AdministratorAccess`-IAM-Richtlinie an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen oder über die Berechtigung zum Ändern der Service-Einstellung für die Aktivierungsebene in Systems Manager verfügen. Diese nutzt die folgenden API-Operationen:

- GetServiceSetting
- UpdateServiceSetting
- ResetServiceSetting

Gehen Sie wie folgt vor, um einem Benutzerkonto eine IAM-Richtlinie hinzufügen. Mit dieser Richtlinie können Benutzer die aktuelle Einstellung für das Kontingent für verwaltete Instances anzeigen. Diese Richtlinie ermöglicht es dem Benutzer auch, die aktuelle Einstellung im angegebenen und zu ändern oder zurückzusetzen. AWS-Konto AWS-Region

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich auf Users (Benutzer).
3. Wählen Sie in der Liste den Namen des Benutzers aus, in den Sie die Richtlinie einbinden möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Klicken Sie rechts auf der Seite unter Permission policies (Berechtigungsrichtlinien) auf Add inline policy (Inline-Richtlinie hinzufügen).
6. Wählen Sie den Tab JSON.
7. Ersetzen Sie den Standardinhalt durch folgenden Inhalt:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"

            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/
managed-instance/activation-tier"
        }
    ]
}
```

8. Wählen Sie Richtlinie prüfen.
9. Geben Sie auf der Seite Review Policy (Richtlinie prüfen) im Feld Name (Name) einen Namen für die Inline-Richtlinie ein. Zum Beispiel: **`Managed-Instances-Tier`**.
10. Wählen Sie Richtlinie erstellen aus.

Administratoren können schreibgeschützte Berechtigungen angeben, indem sie dem Benutzer die folgende Inline-Richtlinie zuweisen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "*"
        }
    ]
}
```

Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Aktivieren des Kontingents für erweiterte Instances (Konsole)

Das folgende Verfahren zeigt Ihnen, wie Sie mit der Systems Manager Manager-Konsole alle Nicht-EC2-Knoten, die mithilfe der Managed-Instance-Aktivierung hinzugefügt wurden, in der angegebenen AWS-Konto und auf die AWS-Region Advanced-Instance-Stufe umstellen.

Bevor Sie beginnen

Stellen Sie sicher, dass die Konsole in dem Bereich geöffnet ist, in AWS-Region dem Sie Ihre verwalteten Instanzen erstellt haben. Über die Liste in der oberen, rechten Ecke der Konsole können Sie zwischen dein einzelnen Regionen wechseln.

Überprüfen Sie, ob Sie die Einrichtungsanforderungen für Ihre Instances der Amazon Elastic Compute Cloud (Amazon EC2) und Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung erfüllt haben. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

> **Important**
>
> Nachfolgend wird beschrieben, wie Sie eine Einstellung auf Kontoebene ändern. Durch diese Änderung fallen für Ihr Konto Gebühren an.

So aktivieren Sie das Kontingent für erweiterte Instances (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie Einstellungen, Instance-Stufeneinstellungen ändern.
4. Überprüfen Sie die Informationen im Dialogfeld zum Ändern der Kontoeinstellungen, und fahren Sie dann fort.
5. Wenn Sie zustimmen, wählen Sie die Option, die Sie akzeptieren möchten, und klicken Sie dann auf Einstellung ändern.

Es kann einige Minuten dauern, bis alle Instances vom Kontingent für Standard-Instances in das Kontingent für erweiterte Instances verschoben wurden.

> **Note**
>
> Informationen zum Wechsel zurück zum Kontingent für Standard-Instances finden Sie unter Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances.

Aktivieren des Kontingents für erweiterte Instances (AWS CLI)

Das folgende Verfahren zeigt Ihnen, wie Sie alle lokalen Server und VMs, die mithilfe der AWS Command Line Interface Managed-Instance-Aktivierung hinzugefügt wurden, in der angegebenen Version ändern AWS-Konto und AWS-Region, um die Stufe Advanced-Instances zu verwenden.

> **Important**
>
> Nachfolgend wird beschrieben, wie Sie eine Einstellung auf Kontoebene ändern. Durch diese Änderung fallen für Ihr Konto Gebühren an.

Um die Stufe Advanced-Instances zu aktivieren, verwenden Sie AWS CLI

1. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Diensteinstellungen für verwaltete Knoten im aktuellen AWS-Konto und anzuzeigen AWS-Region.

   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
   ```

   Der Befehl gibt Informationen wie die folgenden zurück.

   ```
   {
       "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"
       }
   }
   ```

Die Stufe „Advanced-Instances“ () wird aktiviert PowerShell

Das folgende Verfahren zeigt Ihnen, wie Sie alle lokalen Server und VMs AWS Tools for Windows PowerShell , die mithilfe der Managed-Instance-Aktivierung hinzugefügt wurden, in der angegebenen AWS-Konto Version ändern und, um die Advanced-Instance-Stufe zu verwenden. AWS-Region

> **Important**
>
> Nachfolgend wird beschrieben, wie Sie eine Einstellung auf Kontoebene ändern. Durch diese Änderung fallen für Ihr Konto Gebühren an.

Um die Stufe „Advanced-Instances“ zu aktivieren, verwenden Sie PowerShell

1. Öffnen Sie den folgenden AWS Tools for Windows PowerShell Befehl und führen Sie ihn aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   ```
   Update-SSMServiceSetting `
       -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier" `
       -SettingValue "advanced"
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Diensteinstellungen für verwaltete Knoten im aktuellen AWS-Konto und anzuzeigen AWS-Region.

   ```
   Get-SSMServiceSetting `
       -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier"
   ```

   Der Befehl gibt Informationen wie die folgenden zurück.

   ```
   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
   ```

Es kann einige Minuten dauern, bis alle Knoten vom Standard-Instances-Kontingent in das Advanced-Instances-Kontingent verschoben wurden.

> **Note**
>
> Informationen zum Wechsel zurück zum Kontingent für Standard-Instances finden Sie unter Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances.

Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances

In diesem Abschnitt wird beschrieben, wie hybrid-aktivierte Knoten, die im Advanced-Instances-Kontingent ausgeführt werden, wieder zum Standard-Instances-Kontingent umgestellt werden. Diese Konfiguration gilt für alle hybridaktivierten Knoten in einem AWS-Konto und einem. AWS-Region

Bevor Sie beginnen

Lesen Sie die folgenden wichtigen Details.

> **Note**
>
> - Sie können nicht wieder zum Kontingent für Standard-Instances zurückkehren, wenn Sie im Konto und in der Region mehr als 1 000 hybrid-aktivierte Knoten ausführen. Sie müssen also zunächst die Registrierung für Knoten aufheben, bis 1 000 oder weniger vorhanden sind. Dies gilt auch für Amazon Elastic Compute Cloud (Amazon EC2)-Instances, die eine Hybrid-Aktivierung von Systems Manager verwenden (kein gängiges Szenario). Weitere Informationen finden Sie unter Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung.
> - Nach dem Zurücksetzen können Sie eine Funktion von nicht mehr verwendenSession Manager, um interaktiv auf Ihre AWS Systems Manager hybridaktivierten Knoten zuzugreifen.
> - Nach dem Zurücksetzen können Sie eine Funktion von nicht mehr verwendenPatch Manager, um von AWS Systems Manager Microsoft veröffentlichte Anwendungen auf hybridaktivierten Knoten zu patchen.
> - Das Zurücksetzen aller hybrid-aktivierten Knoten auf das Kontingent für Standard-Instances kann 30 Minuten oder länger dauern.

In diesem Abschnitt wird beschrieben, wie Sie alle hybridaktivierten Knoten auf einer Stufe AWS-Konto und AWS-Region von der Stufe „Advanced-Instances“ auf die Stufe „Standard-Instances“ zurücksetzen.

Zurücksetzen auf das Kontingent für Standard-Instances (Konsole)

Das folgende Verfahren zeigt Ihnen, wie Sie die Systems Manager Manager-Konsole verwenden, um alle hybridaktivierten Knoten in Ihrer Hybrid- und Multicloud-Umgebung so zu ändern, dass sie die Stufe „Standardinstanzen“ im angegebenen und verwenden. AWS-Konto AWS-Region

So stellen Sie das Kontingent für Standard-Instances wieder her (Konsole)

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/systems-manager/. AWS Systems Manager
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie das Dropdown-Menü Account settings (Kontoeinstellungen) und wählen Sie Instance tier settings (Instance-Kontingenteneinstellungen).
4. Wählen Sie die Option Change account settings (Kontoeinstellungen ändern) aus.
5. Lesen Sie die Informationen zum Ändern der Kontoeinstellungen im angezeigten Popup-Fenster und wählen Sie ggf. die Option zum Akzeptieren und Fortfahren aus.

Zurücksetzen auf das Kontingent für Standard-Instances (AWS CLI)

Das folgende Verfahren zeigt Ihnen, wie Sie die AWS Command Line Interface verwenden, um alle hybridaktivierten Knoten in Ihrer Hybrid- und Multicloud-Umgebung so zu ändern, dass sie die Stufe „Standard-Instances“ im angegebenen und verwenden. AWS-Konto AWS-Region

Um zur Stufe „Standard-Instances“ zurückzukehren, verwenden Sie AWS CLI

1. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl 30 Minuten später aus, um die Einstellungen für verwaltete Instanzen in der aktuellen Version AWS-Konto und anzuzeigen AWS-Region.

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "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"
    }
}
```

Der Status ändert sich auf Default (Standard), nachdem die Anfrage genehmigt wurde.

Rückkehr zur Stufe „Standardinstanzen“ () PowerShell

Das folgende Verfahren zeigt Ihnen, wie Sie hybridaktivierte Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung so ändern, dass sie die Stufe „Standard-Instances“ in der angegebenen und verwenden. AWS Tools for Windows PowerShell AWS-Konto AWS-Region

Um zur Stufe „Standard-Instances“ zurückzukehren, verwenden Sie PowerShell

1. Öffnen Sie den folgenden AWS Tools for Windows PowerShell Befehl und führen Sie ihn aus.

```
Update-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier" `
    -SettingValue "standard"
```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl 30 Minuten später aus, um die Einstellungen für verwaltete Instanzen in der aktuellen Version AWS-Konto und anzuzeigen AWS-Region.

```
Get-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier"
```

   Der Befehl gibt Informationen wie die folgenden zurück.

```
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
```

   Der Status ändert sich auf Default (Standard), nachdem die Anfrage genehmigt wurde.

Zurücksetzen von Passwörtern für verwaltete Knoten

Sie können das Passwort für einen beliebigen Benutzer auf einem verwalteten Knoten zurücksetzen. Dazu gehören Amazon Elastic Compute Cloud (Amazon EC2) -Instances, AWS IoT Greengrass

Kerngeräte und lokale Server, Edge-Geräte und virtuelle Maschinen (VMs), die von verwaltet werden. AWS Systems Manager Die Funktion zum Zurücksetzen von Passwörtern basiert auf Session Manager einer Funktion von. AWS Systems Manager Sie können diese Funktionalität verwenden, um eine Verbindung zu verwalteten Knoten herzustellen, ohne eingehende Ports öffnen, Bastion-Hosts zu pflegen oder SSH-Schlüssel verwalten zu müssen.

Die Option zum Zurücksetzen des Passworts ist nützlich, wenn ein Benutzer ein Passwort vergessen hat, oder wenn Sie schnell ein Passwort aktualisieren möchten, ohne eine RDP- oder SSH-Verbindung mit einem verwalteten Knoten herzustellen.

Voraussetzungen

Um das Passwort auf einem verwalteten Knoten zurücksetzen zu können, müssen die folgenden Anforderungen erfüllt sein:

- Der verwaltete Knoten, auf dem Sie ein Passwort ändern möchten, muss ein von Systems Manager verwalteter Knoten sein. Auch muss SSM Agent Version 2.3.668.0 oder höher auf dem verwalteten Knoten installiert sein. Informationen über das Installieren oder Aktualisieren von SSM Agent finden Sie unter Arbeiten mit SSM Agent.
- Die Funktionalität zum Zurücksetzen des Passworts verwendet die Session Manager-Konfiguration, die für Ihr Konto eingerichtet ist, um eine Verbindung mit dem verwalteten Knoten herzustellen. Aus diesem Grund müssen die Voraussetzungen für die Verwendung von Session Manager für Ihr Konto in der aktuellen AWS-Region erfüllt sein. Weitere Informationen finden Sie unter Einrichten von Session Manager.

  > **Note**
  >
  > Session Manager-Support für On-Premises-Knoten wird nur für das Advanced-Instances-Kontingent bereitgestellt. Weitere Informationen finden Sie unter Aktivieren des Kontingents für erweiterte Instances.

- Der AWS Benutzer, der das Passwort ändert, muss über die `ssm:SendCommand` Berechtigung für den verwalteten Knoten verfügen. Weitere Informationen finden Sie unter Den Zugriff von Run Command anhand von Tags beschränken.

Beschränken des Zugriffs

Sie können die Fähigkeit eines Benutzers, Passwörter zurückzusetzen, auf bestimmte verwaltete Knoten beschränken. Dies geschieht durch Verwendung von identitätsbasierten Richtlinien für die

Session Manager-Operation `ssm:StartSession` über dem SSM-Dokument `AWS-PasswordReset`. Weitere Informationen finden Sie unter Kontrollieren des Sitzungszugriffs von Benutzern auf Instances.

Verschlüsseln von Daten

Aktivieren Sie AWS Key Management Service (AWS KMS) die vollständige Verschlüsselung für Session Manager Daten, um die Option zum Zurücksetzen des Kennworts für verwaltete Knoten zu verwenden. Weitere Informationen finden Sie unter So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Zurücksetzen eines Passworts auf einem verwalteten Knoten

Sie können ein Passwort auf einem von Systems Manager verwalteten Knoten mithilfe der Systems Manager Fleet ManagerManager-Konsole oder der Taste AWS Command Line Interface (AWS CLI) zurücksetzen.

So ändern Sie das Passwort auf einem verwalteten Knoten (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem Knoten, der ein neues Passwort benötigt.
4. Wählen Sie Instance-Aktionen, Passwort zurücksetzen.
5. Geben Sie im Feld User name (Benutzername) den Namen des Benutzers ein, für den Sie das Passwort ändern. Dabei kann es sich um jeden Benutzernamen handeln, der ein Konto auf dem Knoten hat.
6. Wählen Sie Absenden aus.
7. Befolgen Sie die Eingabeaufforderungen im Befehlsfenster Enter new password (Neues Passwort eingeben), um das neue Passwort anzugeben.

> **Note**
>
> Wenn die Version von SSM Agent auf dem verwalteten Knoten kein Zurücksetzen von Passwörtern unterstützt, werden Sie aufgefordert, mit Run Command, einer Funktion von AWS Systems Manager, eine unterstützte Version zu installieren.

So setzen Sie das Passwort auf einem verwalteten Knoten zurück (AWS CLI)

1. Um das Passwort für einen Benutzer auf einem verwalteten Knoten zurückzusetzen, führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   > **Note**
   >
   > Um das AWS CLI zum Zurücksetzen eines Passworts zu verwenden, muss das Session Manager Plugin auf Ihrem lokalen Computer installiert sein. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugins für die 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. Befolgen Sie die Eingabeaufforderungen im Befehlsfenster Enter new password (Neues Passwort eingeben), um das neue Passwort anzugeben.

Fehlerbehebung beim Zurücksetzen von Passwörtern auf verwalteten Knoten

Viele Probleme beim Zurücksetzen von Passwörtern können behoben werden, wenn Sie sicherstellen, dass Sie die Voraussetzungen für das Zurücksetzen von Passwörtern gelesen haben. Bei anderen Problemen nehmen Sie die folgenden Informationen zur Behebung von Problemen beim Zurücksetzen von Passwörtern zu Hilfe.

Themen

- Verwalteter Knoten nicht verfügbar

- SSM Agentnicht up-to-date (Konsole)
- Optionen zum Zurücksetzen des Passworts werden nicht bereitgestellt (AWS CLI)
- Keine Berechtigung zum Ausführen von ssm:SendCommand
- Session Manager-Fehlermeldung

Verwalteter Knoten nicht verfügbar

Problem: Sie möchten auf der Seite Managed Instances (Verwaltete Instances) der Konsole das Passwort für einen verwalteten Knoten zurücksetzen, der jedoch nicht in der Liste enthalten ist.

- Solution (Lösung): Der verwaltete Knoten, mit dem Sie eine Verbindung herstellen möchten, wurde möglicherweise nicht für Systems Manager konfiguriert. Um eine EC2-Instance mit Systems Manager zu verwenden, muss der Instance ein AWS Identity and Access Management (IAM-) Instanzprofil angehängt werden, das Systems Manager die Erlaubnis erteilt, Aktionen auf Ihren Instances auszuführen. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

  Um eine Nicht-EC2-Maschine mit Systems Manager zu verwenden, erstellen Sie eine IAM-Servicerolle, die Systems Manager die Berechtigung gibt, Aktionen auf Ihren verwalteten Knoten durchzuführen. Weitere Informationen finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Servicerolle in Hybrid- und Multicloud-Umgebungen. (Session ManagerUnterstützung für lokale Server und VMs wird nur für die Stufe „Advanced-Instances“ bereitgestellt. Weitere Informationen finden Sie unter.) Aktivieren des Kontingents für erweiterte Instances

SSM Agentnicht up-to-date (Konsole)

Problem: Es wird eine Meldung angezeigt, dass die Version von SSM Agent die Funktionalität zum Zurücksetzen von Passwörtern nicht unterstützt.

- Lösung: Es ist Version 2.3.668.0 oder höher von SSM Agent erforderlich, um Passwörter zurückzusetzen. In der Konsole können Sie den Agent auf dem verwalteten Knoten aktualisieren, indem Sie Update SSM Agent (aktualisieren) auswählen.

  Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass

der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisen aufGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.

Optionen zum Zurücksetzen des Passworts werden nicht bereitgestellt (AWS CLI)

Problem: Sie haben mit dem AWS CLI `start-session` Befehl erfolgreich eine Verbindung zu einem verwalteten Knoten hergestellt. Sie haben das SSM-Dokument `AWS-PasswordReset` und einen gültigen Benutzernamen angegeben, aber die Eingabeaufforderungen zur Änderung des Passworts werden nicht angezeigt.

- Lösung: Die Version von SSM Agent auf dem verwalteten Knoten ist es nicht up-to-date. Es ist Version 2.3.668.0 oder höher erforderlich, um das Passwort zurückzusetzen.

  Eine aktualisierte Version von SSM Agent wird veröffentlicht, wenn neue Funktionen zu Systems Manager hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisen aufGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.

Keine Berechtigung zum Ausführen von **`ssm:SendCommand`**

Problem: Sie versuchen, eine Verbindung zu einem verwalteten Knoten herzustellen, um das Passwort zu ändern, aber es wird eine Fehlermeldung angezeigt, dass Sie nicht autorisiert sind, `ssm:SendCommand` auf dem verwalteten Knoten auszuführen.

- Lösung: Ihre IAM-Richtlinie muss die Berechtigung zum Ausführen des `ssm:SendCommand`-Befehls enthalten. Weitere Informationen finden Sie unter Den Zugriff von Run Command anhand von Tags beschränken.

Session Manager-Fehlermeldung

Problem: Sie erhalten eine Fehlermeldung zu Session Manager.

- Lösung: Die Unterstützung für das Zurücksetzen von Passwörtern erfordert, dass Session Manager korrekt konfiguriert ist. Weitere Informationen finden Sie unter Einrichten von Session Manager und Fehlerbehebung für Session Manager.

Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung

Wenn Sie einen lokalen Server, ein Edge-Gerät oder eine virtuelle Maschine (VM) nicht mehr mithilfe von verwenden möchten AWS Systems Manager, können Sie die Registrierung aufheben. Wenn Sie einen hybridaktivierten Knoten deregistrieren, wird er aus der Liste der verwalteten Knoten in Systems Manager entfernt. AWS Systems Manager Agent (SSM Agent), der auf dem hybridaktivierten Knoten ausgeführt wird, kann sein Autorisierungstoken nicht aktualisieren, da es nicht mehr registriert ist. SSM Agentwechselt in den Ruhezustand und reduziert die Ping-Frequenz für Systems Manager in der Cloud auf einmal pro Stunde.

Sie können einen On-Premises-Server, ein Edge-Gerät oder eine VM jederzeit erneut registrieren. Systems Manager speichert den Befehlsverlauf für einen verwaltete Knoten, deren Registrierung aufgehoben wurde, 30 Tage lang.

Im folgenden Verfahren wird beschrieben, wie Sie die Registrierung für einen hybrid-aktivierten Knoten mithilfe der Systems-Manager-Konsole aufheben. Informationen dazu, wie Sie dies mithilfe der AWS Command Line Interface tun, finden Sie unter deregister-managed-instance.

Um die Registrierung eines hybrid-aktivierten Knotens aufzuheben (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Aktivieren Sie das Kontrollkästchen neben dem verwalteten Knoten, dessen Registrierung Sie aufheben möchten.
4. Wählen Sie Knotenaktionen, Tools, Diesen verwalteten Knoten deregistrieren aus.
5. Überprüfen Sie die Informationen im Dialogfeld Diesen verwalteten Knoten deregistrieren. Wenn Sie zustimmen, wählen Sie Deregister aus.

Verwenden der Standardeinstellung für die Host-Management-Konfiguration

Die Einstellung Standard-Host-Management-Konfiguration ermöglicht es AWS Systems Manager , Ihre Amazon EC2 EC2-Instances automatisch als verwaltete Instances zu verwalten. Eine verwaltete Instance ist eine EC2-Instance, die für die Verwendung mit Systems Manager konfiguriert ist.

Die Verwaltung Ihrer Instances mit Systems Manager bietet unter anderem folgende Vorteile:

- Stellen Sie mit Session Manager eine sichere Verbindung zu Ihren EC2-Instances her.
- Führen Sie automatisierte Patch-Scans mit Patch Manager durch.
- Zeigen Sie mit Systems Manager Inventory detaillierte Informationen zu Ihren Instances an.
- Verfolgen und verwalten Sie Instances mithilfe von Fleet Manager.
- Halten Sie SSM Agent automatisch auf dem neuesten Stand.

Fleet Manager, Bestand, Patch Manager und Session Manager sind Funktionen von Systems Manager.

Die Standard-Host-Management-Konfiguration ermöglicht die Verwaltung von EC2-Instances, ohne dass Sie manuell ein AWS Identity and Access Management (IAM-) Instance-Profil erstellen müssen. Stattdessen erstellt und wendet die Standard-Host-Management-Konfiguration eine Standard-IAM-Rolle an, um sicherzustellen, dass Systems Manager über Berechtigungen zur Verwaltung aller Instances in der AWS-Konto und an der AWS-Region Stelle verfügt, an der sie aktiviert ist.

Wenn die bereitgestellten Berechtigungen für Ihren Anwendungsfall nicht ausreichen, können Sie auch Richtlinien zur Standard-IAM-Rolle hinzufügen, die von der Standardkonfiguration für die Host-Verwaltung erstellt wird. Wenn Sie keine Berechtigungen für alle Funktionen benötigen, die von der Standard-IAM-Rolle bereitgestellt werden, können Sie alternativ Ihre eigene benutzerdefinierte Rolle und Richtlinien erstellen. Alle Änderungen an der IAM-Rolle, die Sie für die Standardkonfiguration für die Host-Verwaltung auswählen, gelten für alle verwalteten Amazon-EC2-Instances in der Region und im Konto.

Weitere Informationen zu der Richtlinie, die von der Standardkonfiguration für die Host-Verwaltung verwendet wird, finden Sie unter AWS verwaltete Richtlinie: InstanceDefault AmazonSSMManagedEC2-Richtlinie.

Implementieren des Zugriffs mit geringsten Berechtigungen

Die in diesem Thema beschriebenen Verfahren sollten nur von Administratoren durchgeführt werden. Daher empfehlen wir, Zugriff mit den geringsten Berechtigungen zu implementieren, um zu

verhindern, dass nichtadministrative Benutzer die Standardkonfiguration für die Host-Verwaltung konfigurieren oder ändern. Beispielrichtlinien, die den Zugriff auf die Standardkonfiguration für die Host-Verwaltung einschränken, finden Sie unter Beispiele für Richtlinien mit den geringsten Berechtigungen für die Standardkonfiguration für die Host-Verwaltung weiter unten in diesem Thema.

> **Important**
>
> Registrierungsinformationen für Instances, die mit der Standard-Host-Management-Konfiguration registriert wurden, werden lokal in den `C:\ProgramData\Amazon` Verzeichnissen `var/lib/amazon/ssm` oder gespeichert. Das Entfernen dieser Verzeichnisse oder der enthaltenen Dateien verhindert, dass die Instance die erforderlichen Anmeldeinformationen für die Verbindung mit Systems Manager über die Standardkonfiguration für die Host-Verwaltung erhält. In diesen Fällen müssen Sie ein IAM-Instanzprofil verwenden, um die erforderlichen Berechtigungen für Ihre Instance bereitzustellen, oder die Instance neu erstellen.

Themen

- Voraussetzungen
- Aktivierung der Standardeinstellung für die Host-Management-Konfiguration
- Deaktivierung der Standardeinstellung für die Host-Management-Konfiguration
- Beispiele für Richtlinien mit den geringsten Berechtigungen für die Standardkonfiguration für die Host-Verwaltung

Voraussetzungen

Um die Standard-Host-Management-Konfiguration in der AWS-Region und an der AWS-Konto Stelle zu verwenden, an der Sie die Einstellung aktivieren, müssen die folgenden Anforderungen erfüllt sein.

- Eine zu verwaltende Instance muss Instance Metadata Service Version 2 (IMDSv2) verwenden.

  Die Standardkonfiguration für die Host-Verwaltung unterstützt die Instance-Metadaten-Service-Version 1 nicht. Informationen zur Umstellung auf IMDSv2 finden Sie unter Übergang zur Verwendung von Instance Metadata Service Version 2 im Amazon EC2 EC2-Benutzerhandbuch
- SSM Agent-Version 3.2.582.0 oder höher muss auf der zu verwaltenden Instance installiert sein.

Informationen zur Überprüfung der auf Ihrer Instance installierten Version von SSM Agent finden Sie unter Überprüfen der SSM Agent-Versionsnummer.

Weitere Informationen zur Aktualisierung von SSM Agent finden Sie unter Automatische Aktualisierung von SSM Agent.

- Sie als Administrator, der die Aufgaben in diesem Thema ausführt, benötigen Berechtigungen für die API-Operationen „Einstellung“, „GetServiceEinstellung“ und ResetServiceUpdateService„Einstellung“. Darüber hinaus müssen Sie über Berechtigungen für die `iam:PassRole`-Berechtigung für die `AWSSystemsManagerDefaultEC2InstanceManagementRole`-IAM-Rolle verfügen. Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die diese Berechtigungen vorsieht. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting",
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:region:account-id:servicesetting/ssm/managed-
instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::account-id:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
```

```
    ]
}
```

- Wenn ein IAM-Instance-Profil bereits mit einer mit Systems Manager zu verwaltenden EC2 Instance verknüpft ist, müssen Sie alle Berechtigungen entfernen, die die `ssm:UpdateInstanceInformation`-Operation zulassen. SSM Agent versucht, die Berechtigungen des Instance-Profils zu verwenden, bevor Sie die Berechtigungen der Standardkonfiguration für die Host-Verwaltung verwenden. Wenn Sie die `ssm:UpdateInstanceInformation`-Operation in Ihrem eigenen IAM-Instance-Profil zulassen, wird die Instance die Berechtigungen der Standardkonfiguration für die Host-Verwaltung nicht verwenden.

Aktivierung der Standardeinstellung für die Host-Management-Konfiguration

Sie können die Standard-Host-Management-Konfiguration von der Fleet Manager Konsole aus oder mithilfe von AWS Command Line Interface oder aktivieren AWS Tools for Windows PowerShell.

Sie müssen die Standard-Host-Management-Konfiguration nacheinander in jeder Region aktivieren, in der Sie Ihre Amazon EC2 EC2-Instances mit dieser Einstellung verwalten möchten.

Nach dem Aktivieren der Standard-Host-Management-Konfiguration kann es bis zu 30 Minuten dauern, bis Ihre Instances die Anmeldeinformationen der Rolle verwenden, die Sie in Schritt 5 des folgenden Verfahrens ausgewählt haben.

So aktivieren Sie die Standardkonfiguration für die Host-Verwaltung (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Kontoverwaltung, Standardkonfiguration für die Host-Verwaltung konfigurieren.
4. Aktivieren Sie Standardkonfiguration für die Host-Verwaltung aktivieren.
5. Wählen Sie die AWS Identity and Access Management (IAM-) Rolle, die verwendet wird, um die Systems Manager Manager-Funktionen für Ihre Instances zu aktivieren. Wir empfehlen die Verwendung der Standardrolle, die in der Standardkonfiguration für die Host-Verwaltung bereitgestellt wird. Sie enthält die Mindestberechtigungen für die Verwaltung Ihrer Amazon-EC2-Instances mit Systems Manager. Wenn Sie es vorziehen, eine benutzerdefinierte Rolle zu verwenden, muss die Vertrauensrichtlinie der Rolle Systems Manager als vertrauenswürdige Entität zulassen.

6. Wählen Sie Konfigurieren, um die Einrichtung abzuschließen.

So aktivieren Sie die Standardkonfiguration für die Host-Verwaltung (Befehlszeile)

1. Erstellen Sie auf Ihrem lokalen Computer eine JSON-Datei, die die folgende Vertrauensbeziehungsrichtlinie enthält.

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

2. Öffnen Sie AWS CLI oder Tools für Windows PowerShell und führen Sie je nach Betriebssystemtyp Ihres lokalen Computers einen der folgenden Befehle aus, um eine Servicerolle in Ihrem Konto zu erstellen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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. Führen Sie den folgenden Befehl aus, um Ihrer neu erstellten Rolle die von `AmazonSSMManagedEC2InstanceDefaultPolicy` verwaltete Richtlinie anzufügen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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. Öffnen Sie AWS CLI oder Tools für Windows PowerShell und führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   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"
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

5. Führen Sie den folgenden Befehl aus, um die aktuellen Diensteinstellungen für die Standard-Hostverwaltungskonfiguration im aktuellen AWS-Konto und anzuzeigen AWS-Region.

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"
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "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"
    }
}
```

Deaktivierung der Standardeinstellung für die Host-Management-Konfiguration

Sie können die Standard-Host-Management-Konfiguration von der Fleet Manager Konsole aus oder mithilfe von AWS Command Line Interface oder AWS Tools for Windows PowerShell deaktivieren.

Sie müssen die Einstellung für die Standard-Host-Management-Konfiguration nacheinander in jeder Region deaktivieren, in der Ihre Amazon EC2 EC2-Instances nicht mehr mit dieser Konfiguration verwaltet werden sollen. Wenn Sie sie in einer Region deaktivieren, wird sie nicht in allen Regionen deaktiviert.

Wenn Sie die Standardkonfiguration für die Host-Verwaltung ausschalten und Ihren Amazon-EC2-Instances kein Instance-Profil zugeordnet haben, das den Zugriff auf Systems Manager ermöglicht, werden diese nicht mehr von Systems Manager verwaltet.

So deaktivieren Sie die Standardkonfiguration für die Host-Verwaltung (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Kontoverwaltung, Standardkonfiguration für die Host-Verwaltung.
4. Deaktivieren Sie Einstellung der Standardkonfiguration für die Host-Verwaltung aktivieren.
5. Wählen Sie Konfigurieren, um die Standardkonfiguration für die Host-Verwaltung zu deaktivieren.

So deaktivieren Sie die Standardkonfiguration für die Host-Verwaltung (Befehlszeile)

- Öffnen Sie AWS CLI oder Tools für Windows PowerShell und führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

  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"
  ```

Beispiele für Richtlinien mit den geringsten Berechtigungen für die Standardkonfiguration für die Host-Verwaltung

Die folgenden Beispielrichtlinien zeigen, wie Sie verhindern können, dass Mitglieder Ihrer Organisation Änderungen an der Standardkonfiguration für die Host-Verwaltung in Ihrem Konto vornehmen.

Service-Kontrollrichtlinie für AWS Organizations

Die folgende Richtlinie zeigt, wie Sie verhindern können, dass Mitglieder, die keine Administratorrechte haben, Ihre AWS Organizations Einstellung für die Standard-Host-Management-Konfiguration aktualisieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

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

Richtlinie für IAM-Prinzipale

Die folgende Richtlinie zeigt, wie Sie verhindern können, dass IAM-Gruppen, -Rollen oder Benutzer in AWS Organizations Ihrem Unternehmen Ihre Einstellung für die Standard-Host-Management-Konfiguration aktualisieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ssm:UpdateServiceSetting",
                "ssm:ResetServiceSetting"
            ],
            "Resource": "arn:aws:ssm:region:account-id:servicesetting/ssm/managed-
instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Deny",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DeleteRole",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::account-id:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole"
        }
    ]
}
```

Verbindung zu einer Windows Server verwalteten Instanz herstellen mit Remote Desktop

Sie können eine Funktion von verwenden Fleet Manager AWS Systems Manager, um mithilfe von (RDP) eine Verbindung zu Ihren Windows Server Amazon Elastic Compute Cloud (Amazon EC2) - Instances herzustellen. Remote Desktop Protocol Fleet Manager Remote Desktop, das von NICE DCV unterstützt wird, bietet Ihnen eine sichere Verbindung zu Ihren Windows Server-Instances direkt von der Systems-Manager-Konsole aus. Sie können bis zu vier gleichzeitige Verbindungen in einem einzigen Browserfenster haben.

Sie können Remote Desktop nur mit Instances verwenden, auf denen Windows Server 2012 RTM oder höher ausgeführt wird. Remote Desktop unterstützt nur englischsprachige Eingaben.

> **Note**
>
> Fleet ManagerRemote Desktop ist ein reiner Konsolendienst und unterstützt keine Befehlszeilenverbindungen zu Ihren verwalteten Instanzen. Um über eine Shell eine Verbindung zu einer Windows Server verwalteten Instanz herzustellen, können Sie eine weitere Funktion von verwendenSession Manager. AWS Systems Manager Weitere Informationen finden Sie unter AWS Systems Manager Session Manager.

Informationen zur Konfiguration von AWS Identity and Access Management (IAM) - Berechtigungen, damit Ihre Instances mit Systems Manager interagieren können, finden Sie unter Instanzberechtigungen für Systems Manager konfigurieren.

Themen

- Einrichten Ihrer Umgebung
- Konfiguration von IAM-Berechtigungen für Remote Desktop
- Authentifizierung von Remote-Desktop-Verbindungen
- Dauer und Gleichzeitigkeit der Remoteverbindung
- Verbindung zu einem verwalteten Knoten über Remote Desktop

Einrichten Ihrer Umgebung

Vergewissern Sie sich vor der Verwendung von Remote Desktop, dass Ihre Umgebung die folgenden Anforderungen erfüllt:

- Konfiguration von verwalteten Knoten

  Stellen Sie sicher, dass Ihre Amazon-EC2-Instances als verwaltete Knoten in Systems Manager konfiguriert sind.

- SSM Agent-Mindestversion

  Stellen Sie sicher, dass auf den Knoten SSM Agent-Version 3.0.222.0 oder höher ausgeführt wird. Hinweise dazu, wie Sie überprüfen können, welche Agentenversion auf einem Knoten läuft, finden Sie unter Überprüfen der SSM Agent-Versionsnummer. Informationen über das Installieren oder Aktualisieren von SSM Agent finden Sie unter Arbeiten mit SSM Agent.

- Konfiguration des RDP-Ports

  Um Remote-Verbindungen zu akzeptieren, muss der Remote Desktop Services-Service auf Ihren Windows Server-Knoten den Standard-RDP-Port 3389 verwenden. Dies ist die Standardkonfiguration von AWS on Amazon Machine Images (AMIs). Sie müssen nicht explizit irgendwelche eingehenden Ports öffnen, um Remote Desktop zu verwenden.

- PSReadLine-Modulversion für Tastaturfunktionen

  Um sicherzustellen, dass Ihre Tastatur in PowerShell ordnungsgemäß funktioniert, stellen Sie sicher, dass auf den Knoten, auf denen Windows Server-2022 läuft, die PSReadLine-Modulversion 2.2.2 oder höher installiert ist. Wenn sie eine ältere Version verwenden, können Sie die erforderliche Version mit dem folgenden Befehl installieren.

  ```
  Install-Module `
      -Name PSReadLine `
      -Repository PSGallery `-MinimumVersion 2.2.2
  ```

- Session-Manager-Konfiguration

  Bevor Sie Remote Desktop verwenden können, müssen Sie die Voraussetzungen für die Einrichtung von Session Manager erfüllen. Wenn Sie über Remote Desktop eine Verbindung zu einer Instanz herstellen, AWS-Region werden alle für Sie AWS-Konto definierten Sitzungseinstellungen angewendet. Weitere Informationen finden Sie unter Einrichten von Session Manager.

  > **Note**
  >
  > Wenn Sie die Aktivitäten von Session Manager mit Amazon Simple Storage Service (Amazon S3) protokollieren, dann erzeugen Ihre Remotedesktop-Verbindungen den

folgenden Fehler in `bucket_name/Port/stderr`. Dieser Fehler ist ein erwartetes Verhalten und kann ignoriert werden.

```
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>
```

Konfiguration von IAM-Berechtigungen für Remote Desktop

Zusätzlich zu den erforderlichen IAM-Berechtigungen für Systems Manager und Session Manager, muss der Benutzer oder die Rolle, die Sie für den Zugriff auf die Konsole verwenden, die folgenden Aktionen erlauben:

- `ssm-guiconnect:CancelConnection`
- `ssm-guiconnect:GetConnection`
- `ssm-guiconnect:StartConnection`

Im Folgenden finden Sie Beispiele für IAM-Richtlinien, die Sie einem Benutzer oder einer Rolle zuordnen können, um verschiedene Arten der Interaktion mit Remote Desktop zu erlauben. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

Standardrichtlinie für die Verbindung mit EC2-Instances

```
{
    "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:*:account-id:instance/*",
                "arn:aws:ssm:*:account-id:managed-instance/*",
                "arn:aws:ssm:*::document/AWS-StartPortForwardingSession"
            ],
            "Condition": {
                "BoolIfExists": {
                    "ssm:SessionDocumentAccessCheck": "true"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm-guiconnect.amazonaws.com"
                }
            }
        },
```

```
        {
            "Sid": "GuiConnect",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:CancelConnection",
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:StartConnection"
            ],
            "Resource": "*"
        }
    ]
}
```

Richtlinie für die Verbindung mit EC2-Instances mit bestimmten Tags

> **Note**
>
> In der folgenden IAM-Richtlinie erfordert der `SSMStartSession` Abschnitt einen Amazon-Ressourcennamen (ARN) für die `ssm:StartSession` Aktion. Wie gezeigt, benötigt der von Ihnen angegebene ARN keine AWS-Konto ID. Wenn Sie eine Konto-ID angeben, wird eine Fleet Manager zurückgegeben`AccessDeniedException`.
> Für den `AccessTaggedInstances` Abschnitt, der sich weiter unten in der Beispielrichtlinie befindet, sind auch ARNs für `ssm:StartSession` erforderlich. Für diese ARNs geben AWS-Konto Sie IDs an.

```
{
    "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": {
                "BoolIfExists": {
                    "ssm:SessionDocumentAccessCheck": "true"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm-guiconnect.amazonaws.com"
                }
            }
        },
        {
            "Sid": "AccessTaggedInstances",
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:account-id:instance/*",
                "arn:aws:ssm:*:account-id:managed-instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/tag key": [
                        "tag value"
                    ]
                }
            }
        },
        {
```

```
            "Sid": "GuiConnect",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:CancelConnection",
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:StartConnection"
            ],
            "Resource": "*"
        }
    ]
}
```

Richtlinie für AWS IAM Identity Center Benutzer, um eine Verbindung zu EC2-Instances herzustellen

```
{
    "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: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": {
                "BoolIfExists": {
                    "ssm:SessionDocumentAccessCheck": "true"
                },
                "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"
            ],
            "Condition": {
                "BoolIfExists": {
                    "ssm:SessionDocumentAccessCheck": "true"
                }
            }
        },
        {
            "Sid": "GuiConnect",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:CancelConnection",
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:StartConnection"
            ],
            "Resource": "*"
        }
    ]
}
```

Authentifizierung von Remote-Desktop-Verbindungen

Wenn Sie eine Remote-Verbindung herstellen, können Sie sich mit Windows-Anmeldeinformationen oder dem Amazon-EC2-Schlüsselpaar (`.pem`-Datei) authentifizieren, das der Instance zugeordnet ist. Informationen zur Verwendung von Schlüsselpaaren finden Sie unter Amazon EC2 EC2-Schlüsselpaare und Windows -Instances im Amazon EC2 EC2-Benutzerhandbuch.

Wenn Sie für die AWS Management Console Nutzung authentifiziert sind, können Sie alternativ eine Verbindung zu Ihren Instances herstellen AWS IAM Identity Center, ohne zusätzliche Anmeldeinformationen angeben zu müssen. Ein Beispiel für eine Richtlinie, die die Authentifizierung von Fernverbindungen mit IAM Identity Center erlaubt, finden Sie unter Konfiguration von IAM-Berechtigungen für Remote Desktop.

Bevor Sie beginnen

Beachten Sie die folgenden Bedingungen für die Verwendung der IAM Identity Center-Authentifizierung, bevor Sie eine Verbindung über Remote Desktop herstellen.

- Remote Desktop unterstützt die IAM Identity Center-Authentifizierung für Knoten in derselben AWS-Region , in der Sie IAM Identity Center aktiviert haben.
- Remote Desktop unterstützt IAM Identity Center-Benutzernamen mit bis zu 16 Zeichen.
- Remote Desktop unterstützt IAM Identity Center-Benutzernamen, die aus alphanumerischen Zeichen und den folgenden Sonderzeichen bestehen: `. - _`

> **Important**
>
> Für IAM Identity Center-Benutzernamen, die die folgenden Zeichen enthalten, können keine Verbindungen hergestellt werden: `+ = , @`.
> IAM Identity Center unterstützt diese Zeichen in Benutzernamen, Fleet Manager-RDP-Verbindungen jedoch nicht.

- Wenn eine Verbindung mit IAM Identity Center authentifiziert wird, erstellt Remote Desktop einen lokalen Windows-Benutzer in der Gruppe Lokale Administratoren der Instance. Dieser Benutzer bleibt bestehen, nachdem die Remoteverbindung beendet wurde.
- Remote Desktop erlaubt keine IAM Identity Center-Authentifizierung für Knoten, die Microsoft Active Directory-Domain-Controller sind.
- Obwohl Remote Desktop die Verwendung der IAM Identity Center-Authentifizierung für Knoten, die einer Active Directory-Domain angeschlossen sind, ermöglicht, raten wir davon ab, dies zu tun. Diese Authentifizierungsmethode gewährt Benutzern administrative Berechtigungen, die restriktivere, von der Domain gewährte Berechtigungen außer Kraft setzen können.

Unterstützte Regionen für die IAM Identity Center-Authentifizierung

Remote Desktop-Verbindungen, die die IAM Identity Center-Authentifizierung verwenden, werden in den folgenden AWS-Regionen unterstützt:

- USA Ost (Ohio): (us-east-2)
- USA Ost (Nord-Virginia): (us-east-1)
- USA West (Nordkalifornien) (us-west-1)
- USA West (Oregon): (us-west-2)
- Afrika (Kapstadt) (af-south-1)
- Asien-Pazifik (Hongkong) (ap-east-1)
- Asien-Pazifik (Mumbai): (ap-south-1)

- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Osaka) (ap-northeast-3)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Asien-Pazifik (Jakarta) (ap-southeast-3)
- Kanada (Zentral): (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Stockholm) (eu-north-1)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Israel (Tel Aviv) (il-central-1)
- Südamerika (São Paulo) (sa-east-1)
- Europa (Mailand) (eu-south-1)
- Naher Osten (Bahrain) (me-south-1)
- AWS GovCloud (US-Ost) (us-gov-east-1)
- AWS GovCloud (US-West) (US-Regierung West-1)

Dauer und Gleichzeitigkeit der Remoteverbindung

Die folgenden Bedingungen gelten für aktive Remote-Desktop-Verbindungen:

- Verbindungsdauer

  Standardmäßig wird eine Remote-Desktop-Verbindung nach 60 Minuten getrennt. Um zu verhindern, dass eine Verbindung getrennt wird, können Sie die Option Sitzung erneuern wählen, bevor sie getrennt wird, um den Timer für die Verbindungsdauer zurückzusetzen.
- Verbindungstimeout

  Eine Remote-Desktop-Verbindung wird getrennt, nachdem sie länger als 10 Minuten inaktiv war.
- Gleichzeitige Verbindungen

Standardmäßig können Sie für dasselbe und maximal 5 aktive Remotedesktopverbindungen gleichzeitig haben. AWS-Konto AWS-Region Um eine Erhöhung des Servicekontingents auf bis zu 25 gleichzeitige Verbindungen zu beantragen, lesen Sie bitte den Abschnitt Beantragung einer Kontingenterhöhung im Benutzerhandbuch Service Quotas.

Verbindung zu einem verwalteten Knoten über Remote Desktop

Unterstützung für das Kopieren und Einfügen von Text durch den Browser

Mit den Browsern Google Chrome und Microsoft Edge können Sie Text von einem verwalteten Knoten auf Ihren lokalen Computer und von Ihrem lokalen Computer in einen verwalteten Knoten, mit dem Sie verbunden sind, kopieren und einfügen.

Mit dem Mozilla Firefox-Browser können Sie Text nur von einem verwalteten Knoten auf Ihren lokalen Computer kopieren und einfügen. Das Kopieren von Ihrem lokalen Computer auf den verwalteten Knoten wird nicht unterstützt.

So stellen Sie über Fleet Manager Remote Desktop eine Verbindung zu einem verwalteten Knoten her

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie den Knoten, zu dem Sie eine Verbindung herstellen möchten. Sie können entweder das Kontrollkästchen oder den Knotennamen auswählen.
4. Wählen Sie im Menü Knotenaktionen die Option Mit Remote Desktop verbinden.
5. Wählen Sie den gewünschten Authentication type (Authentifizierungs-Typ). Wenn Sie Benutzeranmeldeinformationen wählen, geben Sie den Benutzernamen und das Passwort für ein Windows-Benutzerkonto auf dem Knoten ein, zu dem Sie eine Verbindung herstellen möchten. Wenn Sie Schlüsselpaar wählen, können Sie die Authentifizierung mit einer der folgenden Methoden durchführen:

   a. Wählen Sie Lokale Maschine durchsuchen, wenn Sie den mit Ihrer Instance verbundenen PEM-Schlüssel aus Ihrem lokalen Dateisystem auswählen möchten.

      – oder –

b. Wählen Sie Schlüsselpaarinhalt einfügen, wenn Sie den Inhalt der PEM-Datei kopieren und in das vorgesehene Feld einfügen möchten.

6. Wählen Sie Connect (Verbinden) aus.
7. Um Ihre bevorzugte Bildschirmauflösung zu wählen, wählen Sie im Menü Aktionen die Option Auflösungen, und wählen Sie dann eine der folgenden Optionen:

   - Automatisch anpassen
   - 1920 x 1080
   - 1400 x 900
   - 1366 x 768
   - 800 x 600

   Die Option Automatisch anpassen legt die Auflösung auf der Grundlage der erkannten Bildschirmgröße fest.

Verwaltung von Amazon EBS-Volumes auf verwalteten Instances

Amazon Elastic Block Store (Amazon EBS) bietet Volumes für die Speicherung auf Blockebene, die in Verbindung mit Instances von Amazon Elastic Compute Cloud (EC2) verwendet werden. EBS-Volumes verhalten sich wie unformatierte Blockgeräte. Sie können diese Volumes als Geräte auf Ihren Instances mounten.

Sie könnenFleet Manager, eine Funktion von, verwenden AWS Systems Manager, um Amazon EBS-Volumes auf Ihren verwalteten Instances zu verwalten. Sie können beispielsweise ein EBS-Volume initialisieren, eine Partition formatieren und das Volume mounten, um es für die Nutzung verfügbar zu machen.

> **Note**
>
> Fleet Manager unterstützt derzeit die Amazon-EBS-Volume-Verwaltung nur für Windows Server-Instances.

Anzeigen von Details zu EBS-Volumes

So zeigen Sie Details für ein EBS-Volume mit Fleet Manager an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben der verwalteten Instance aus, deren EBS-Volumedetails Sie anzeigen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, EBS-Volumes.
6. Um Details zu einem EBS-Volume anzuzeigen, wählen Sie seine ID in der Spalte Volume-ID.

Initialisieren und Formatieren eines EBS-Volumes

So initialisieren und formatieren Sie ein EBS-Volume mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben der verwalteten Instance aus, die Sie initialisieren, formatieren und mounten möchten. Sie können ein EBS-Volume nur initialisieren, wenn sein Datenträger leer ist.
4. Wählen Sie die Option View details aus.
5. Wählen Sie im Menü Tools die Option EBS-Volumes.
6. Wählen Sie die Schaltfläche neben dem EBS-Volume, das Sie initialisieren und formatieren möchten.
7. Wählen Sie Initialisieren und formatieren.
8. Wählen Sie unter Partitionsstil den Partitionsstil aus, den Sie für das EBS-Volume verwenden möchten.
9. (Optional) Wählen Sie einen Laufwerksbuchstaben für die Partition.
10. (Optional) Geben Sie einen Partitionsnamen ein, um die Partition zu identifizieren.
11. Wählen Sie das Dateisystem aus, das zum Organisieren der in der Partition gespeicherten Dateien und Daten verwendet werden soll.

12. Wählen Sie Bestätigen, um das EBS-Volume zur Verwendung verfügbar zu machen. Sie können die Partitionskonfiguration von der AWS Management Console nach der Bestätigung nicht mehr ändern. Sie können sich jedoch mit SSH oder RDP bei der Instance anmelden, um die Partitionskonfiguration zu ändern.

Arbeiten mit dem Dateisystem

Sie könnenFleet Manager, eine Fähigkeit von AWS Systems Manager, verwenden, um mit dem Dateisystem auf Ihren verwalteten Knoten zu arbeiten. Mit Fleet Manager können Sie Informationen über das Verzeichnis und die Dateidaten anzeigen, die auf den Volumes gespeichert sind, die an Ihre verwalteten Knoten angefügt sind. Sie können beispielsweise den Namen, die Größe, die Erweiterung, den Besitzer und die Berechtigungen für Ihre Verzeichnisse und Dateien anzeigen. Bis zu 10.000 Zeilen von Dateidaten können als Text in der Fleet Manager-Konsole angezeigt werden. Sie können diese Funktion auch für `tail`-Dateien verwenden. Bei Verwendung von `tail`, um Dateidaten anzuzeigen, werden zunächst die letzten 10 Zeilen der Datei angezeigt. Wenn neue Datenzeilen in die Datei geschrieben werden, wird die Ansicht in Echtzeit aktualisiert. Daher können Sie Protokolldaten von der Konsole aus überprüfen, was die Effizienz Ihrer Fehlerbehebung und Systemverwaltung verbessern kann. Darüber hinaus können Sie Verzeichnisse erstellen und Dateien und Verzeichnisse kopieren, ausschneiden, einfügen, umbenennen oder löschen.

Wir empfehlen Ihnen, regelmäßige Backups zu erstellen oder Snapshots der Amazon Elastic Block Store (Amazon EBS)-Volumes zu erstellen, die an Ihre verwalteten Knoten angefügt sind. Beim Kopieren oder Ausschneiden und Einfügen von Dateien werden vorhandene Dateien und Verzeichnisse im Zielpfad mit dem gleichen Namen wie die neuen Dateien oder Verzeichnisse ersetzt. Schwerwiegende Probleme können auftreten, wenn Sie Systemdateien und -verzeichnisse ersetzen oder ändern. AWS garantiert nicht, dass diese Probleme gelöst werden können. Ändern Sie Systemdateien auf eigenes Risiko. Sie sind für alle Änderungen an Dateien und Verzeichnissen verantwortlich und stellen sicher, dass Sie Backups haben. Das Löschen oder Ersetzen von Dateien und Verzeichnissen kann nicht rückgängig gemacht werden.

> **Note**
>
> Fleet ManagerverwendetSession Manager, eine Fähigkeit von AWS Systems Manager, um Textvorschauen und `tail` Dateien anzuzeigen. Für Amazon Elastic Compute Cloud (Amazon EC2)-Instances muss das Instance-Profil, das Ihren verwalteten Instances angefügt ist, Berechtigungen für Session Manager bereitstellen, um diese Funktion zu verwenden. Weitere Informationen zum Hinzufügen von Session Manager-

Berechtigungen an ein Instance-Profil finden Sie unter Hinzufügen von Session Manager-Berechtigungen für eine vorhandene IAM-Rolle. Auch die AWS Key Management Service (AWS KMS)-Verschlüsselung muss in Ihren Sitzungseinstellungen aktiviert sein, um Fleet Manager-Features nutzen zu können. Weitere Hinweise zur Aktivierung der AWS KMS Verschlüsselung für finden Sie Session Manager unterSo aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Anzeigen des Dateisystems mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit dem Dateisystem, das Sie anzeigen möchten.
4. Wählen Sie Tools, Dateisysteme.

Anzeigen von Textvorschauen mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien, die Sie anzeigen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Wählen Sie den File name (Dateiname) des Verzeichnisses, das die Datei enthält, die Sie in der Vorschau anzeigen möchten.
6. Wählen Sie die Schaltfläche neben der Datei, deren Inhalt Sie in der Vorschau anzeigen möchten.
7. Wählen Sie Aktionen, Vorschau als Text.

Verfolgung von Dateien mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien, die Sie verfolgen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Wählen Sie den File name (Dateiname) des Verzeichnisses, das die Datei enthält, die verfolgen möchten.
6. Wählen Sie die Schaltfläche neben der Datei, deren Inhalt Sie verfolgen möchten.
7. Wählen Sie Aktionen, Enddatei.

So kopieren oder fügen Sie Dateien oder Verzeichnisse mit Fleet Manager ein

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien, die Sie kopieren oder ausschneiden und einfügen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Um eine Datei zu kopieren oder auszuschneiden, wählen Sie den File name (Dateiname) des Verzeichnisses, das die Datei enthält, die Sie kopieren oder ausschneiden möchten. Um ein Verzeichnis zu kopieren oder auszuschneiden, wählen Sie die Schaltfläche neben dem Verzeichnis, das Sie kopieren oder ausschneiden möchten, und fahren Sie dann mit Schritt 8 fort.
6. Wählen Sie die Schaltfläche neben der Datei, die Sie kopieren oder ausschneiden möchten.
7. Wählen Sie im Menü Actions (Aktionen) Copy (Kopieren) oder Cut (Ausschneiden).
8. Wählen Sie in der Ansicht File system (Dateisystem) die Schaltfläche neben dem Verzeichnis, in das Sie die Datei einfügen möchten.
9. Wählen Sie im Menü Actions (Aktionen) Paste (Einfügen).

So können Sie Dateien oder Verzeichnisse mit Fleet Manager umbenennen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.

3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien oder Verzeichnissen, die Sie umbenennen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Um eine Datei umzubenennen, wählen Sie den File name (Dateiname) des Verzeichnisses, das die Datei enthält, die Sie umbenennen möchten. Um ein Verzeichnis umzubenennen, wählen Sie die Schaltfläche neben dem Verzeichnis, das Sie umbenennen möchten, und fahren Sie dann mit Schritt 8 fort.
6. Wählen Sie die Schaltfläche neben der Datei, deren Inhalt Sie umbenennen möchten.
7. Wählen Sie Aktionen, Umbenennen.
8. Geben Sie im Feld Dateiname den neuen Namen für die Datei ein und wählen Sie Umbenennen.

So löschen Sie Dateien oder Verzeichnisse mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien oder Verzeichnissen, die Sie löschen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Um eine Datei zu löschen, wählen Sie File name (Dateiname) des Verzeichnisses, das die Datei enthält, die Sie löschen möchten. Um ein Verzeichnis zu löschen, wählen Sie die Schaltfläche neben dem Verzeichnis, das Sie löschen möchten, und fahren Sie dann mit Schritt 7 fort.
6. Wählen Sie die Schaltfläche neben der Datei, deren Inhalt Sie löschen möchten.
7. Wählen Sie Aktionen, Löschen.

So erstellen Sie ein Verzeichnisses mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens, in dem Sie ein Verzeichnis erstellen möchten.
4. Wählen Sie Tools, Dateisysteme.

5. Wählen Sie den File name (Dateiname) des Verzeichnisses, in dem Sie ein neues Verzeichnis erstellen möchten.
6. Wählen Sie Create directory (Verzeichnis erstellen).
7. Geben Sie im Feld Verzeichnisname den Namen für das neue Verzeichnis ein und wählen Sie Verzeichnis erstellen.

Überwachung der Leistung verwalteter Knoten

Sie können die FunktionFleet Manager, verwenden AWS Systems Manager, um Leistungsdaten zu Ihren verwalteten Knoten in Echtzeit einzusehen. Die Leistungsdaten werden von Leistungsindikatoren abgerufen.

Die folgenden Leistungsindikatoren sind in Fleet Manager verfügbar:

- CPU-Auslastung
- Festplatten-Input/Output-(I/O)-Auslastung
- Netzwerkdatenverkehr
- Speicherauslastung

> **Note**
>
> Fleet ManagerverwendetSession Manager, eine Fähigkeit von AWS Systems Manager, zum Abrufen von Leistungsdaten. Für Amazon Elastic Compute Cloud (Amazon EC2)-Instances muss das Instance-Profil, das Ihren verwalteten Instances angefügt ist, Berechtigungen für Session Manager bereitstellen, um diese Funktion zu verwenden. Weitere Informationen zum Hinzufügen von Session Manager-Berechtigungen an ein Instance-Profil finden Sie unter Hinzufügen von Session Manager-Berechtigungen für eine vorhandene IAM-Rolle. Auch die AWS Key Management Service (AWS KMS)-Verschlüsselung muss in Ihren Sitzungseinstellungen aktiviert sein, um Fleet Manager-Features nutzen zu können. Weitere Hinweise zum Aktivieren der AWS KMS Verschlüsselung für Session Manager finden Sie unterSo aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Anzeigen von Leistungsdaten mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, dessen Leistung Sie überwachen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Leistungsindikatoren.

Arbeiten mit Prozessen

Sie können eine Fähigkeit von verwenden Fleet Manager AWS Systems Manager, um mit Prozessen auf Ihren verwalteten Instanzen zu arbeiten. Mit Fleet Manager können Sie Informationen über Prozesse anzeigen. Beispielsweise können Sie zusätzlich zu ihren Handles und Threads die CPU-Auslastung und Speicherauslastung von Prozessen sehen. Mit Fleet Manager können Sie Prozesse von der Konsole aus starten und beenden.

> **Note**
>
> Fleet ManagerverwendetSession Manager, eine Fähigkeit von AWS Systems Manager, zum Abrufen von Prozessdaten. Für Amazon Elastic Compute Cloud (Amazon EC2)-Instances muss das Instance-Profil, das Ihren verwalteten Instances angefügt ist, Berechtigungen für Session Manager bereitstellen, um diese Funktion zu verwenden. Weitere Informationen zum Hinzufügen von Session Manager-Berechtigungen an ein Instance-Profil finden Sie unter Hinzufügen von Session Manager-Berechtigungen für eine vorhandene IAM-Rolle. Auch die AWS Key Management Service (AWS KMS)-Verschlüsselung muss in Ihren Sitzungseinstellungen aktiviert sein, um Fleet Manager-Features nutzen zu können. Weitere Hinweise zum Aktivieren der AWS KMS Verschlüsselung für Session Manager finden Sie unterSo aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

So zeigen Sie Details zu Prozessen mit Fleet Manager an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.

3. Wählen Sie die Verknüpfung der Instance aus, deren Prozesse Sie anzeigen möchten.
4. Wählen Sie Tools, Prozesse.

So starten Sie einen Prozess mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung der Instance aus, auf der Sie einen Prozess starten möchten.
4. Wählen Sie Tools, Prozesse.
5. Wählen Sie Start new process (Neuen Prozess starten).
6. Geben Sie im Feld Prozessname oder vollständiger Pfad den Namen des Prozesses oder den vollständigen Pfad zur ausführbaren Datei ein.
7. (Optional) Geben Sie im Feld Arbeitsverzeichnis den Verzeichnispfad ein, in dem der Prozess ausgeführt werden soll.

So beenden Sie einen Prozess mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung der Instance aus, auf der Sie einen Prozess starten möchten.
4. Wählen Sie Tools, Prozesse.
5. Wählen Sie die Schaltfläche neben dem Prozess, den Sie beenden möchten.
6. Wählen Sie Aktionen, Prozess beenden oder Aktionen, Prozessstruktur beenden.

> **Note**
>
> Durch das Beenden eines Prozessbaums werden auch alle Prozesse und Anwendungen beendet, die diesen Prozess verwenden.

Protokolle auf verwalteten Knoten anzeigen

Sie können die FunktionFleet Manager, verwenden, um Protokolldaten einzusehen AWS Systems Manager, die auf Ihren verwalteten Knoten gespeichert sind. Bei Windows-verwalteten Knoten können Sie aus der Konsole Windows-Ereignisprotokolle anzeigen und ihre Details kopieren. Um Ihnen bei der Suche nach Ereignissen zu helfen, filtern Sie Windows-Ereignisprotokolle nach Event level (Event-Ebene), Event ID (Ereignis-ID), Event source (Ereignisquelle) und Time created (Erstellungszeitpunkt). Sie können auch andere Protokolldaten mit dem Verfahren zum Anzeigen des Dateisystemen anzeigen. Weitere Informationen zum Anzeigen des Dateisystems mit Fleet Manager finden Sie unter Arbeiten mit dem Dateisystem.

Anzeigen von Windows-Ereignisprotokolle mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, dessen Ereignisprotokolle Sie anzeigen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Windows-Ereignisprotokolle.
6. Wählen Sie den Log name (Protokollnamen), welcher die Ereignisse enthält, die Sie anzeigen möchten.
7. Wählen Sie die Schaltfläche neben dem Log name (Protokollnamen), den Sie anzeigen möchten und wählen Sie dann View events (Anzeigen von Ereignissen).
8. Wählen Sie die Schaltfläche neben dem Ereignis, das Sie anzeigen möchten und wählen Sie dann View event details (Eventdetails anzeigen).
9. (Optional) Wählen Sie Copy as JSON (Copy as JSON), um die Ereignisdetails in die Zwischenablage zu kopieren.

Verwaltung von Betriebssystembenutzerkonten auf verwalteten Knoten

Sie können eine Funktion von verwenden Fleet Manager AWS Systems Manager, um Betriebssystem-Benutzerkonten (OS) auf Ihren verwalteten Knoten zu verwalten. Sie können beispielsweise Benutzer und Gruppen erstellen und löschen. Darüber hinaus können Sie Details wie Gruppenmitgliedschaft, Benutzerrollen und Status anzeigen.

> **Important**
>
> Fleet Managerverwendet Run Command undSession Manager, Funktionen von AWS Systems Manager, für verschiedene Benutzerverwaltungsvorgänge. Daher könnte ein Benutzer einem Betriebssystembenutzerkonto Berechtigungen erteilen, die andernfalls nicht möglich wären. Das liegt daran, dass AWS Systems Manager Agent (SSM Agent) auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances mit Root-Rechten (Linux) oder SYSTEM-Berechtigungen (Windows Server) ausgeführt wird. Weitere Informationen zum Einschränken des Zugriffs auf Befehle auf Root-Ebene über SSM Agent finden Sie unter Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent. Um den Zugriff auf diese Funktion einzuschränken, empfehlen wir, AWS Identity and Access Management (IAM-) Richtlinien für Ihre Benutzer zu erstellen, die nur Zugriff auf die von Ihnen definierten Aktionen gewähren. Weitere Informationen zum Erstellen von IAM-Richtlinien für Fleet Manager finden Sie unter Schritt 1: Erstellen einer IAM-Richtlinie mit Fleet Manager-Berechtigungen.

Erstellen einer Benutzergruppe

> **Note**
>
> Fleet Manager nutzt Session Manager, um Kennwörter für neue Benutzer festzulegen. Für Amazon-EC2-Instances muss das Instance-Profil, das Ihren verwalteten Instances angefügt ist, Berechtigungen für Session Manager bereitstellen, um dieses Feature zu verwenden. Weitere Informationen zum Hinzufügen von Session Manager-Berechtigungen an ein Instance-Profil finden Sie unter Hinzufügen von Session Manager-Berechtigungen für eine vorhandene IAM-Rolle. Außerdem muss die AWS Key Management Service (AWS KMS) Verschlüsselung in Ihren Sitzungseinstellungen aktiviert sein, um Fleet Manager Funktionen nutzen zu können. Weitere Informationen zum Aktivieren der AWS KMS -Verschlüsselung für Session Manager finden Sie unter So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Erstellen eines OS-Benutzerkontos mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen neuen Benutzer erstellen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Users und anschließend die Option Create user (Benutzer erstellen).
7. Geben Sie einen Wert für den Namen des neuen Benutzers an.
8. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Set password (Passwort festlegen). Am Ende des Verfahrens werden Sie aufgefordert, ein Passwort für den neuen Benutzer einzugeben.
9. Wählen Sie Create user (Benutzer erstellen). Wenn Sie das Kontrollkästchen zum Erstellen eines Kennworts für den neuen Benutzer aktiviert haben, werden Sie aufgefordert, einen Wert für das Kennwort einzugeben und Done (Fertig) zu wählen. Wenn das angegebene Passwort die Anforderungen der lokalen oder Domain-Richtlinien Ihres verwalteten Knotens nicht erfüllt, wird ein Fehler zurückgegeben.

Erstellen einer OS-Gruppe mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie eine Gruppe erstellen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Groups (Gruppen) und dann Create group (Gruppe erstellen) aus.
7. Geben Sie einen Wert für den Namen der neuen Gruppe an.
8. (Optional) Geben Sie einen Wert für die Description (Beschreibung) der neuen Gruppe ein.
9. (Optional) Wählen Sie die Benutzer aus, die zu den Group members (Gruppenmitgliedern) hinzugefügt werden sollen.
10. Wählen Sie Create group (Gruppe erstellen).

Benutzer- oder Gruppenmitgliedschaft aktualisieren

Hinzufügen eines Betriebssystem-Benutzerkontos zu einer neuen Gruppe mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem sich das Benutzerkonto befindet, das Sie aktualisieren möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Users.
7. Wählen Sie die Schaltfläche neben dem Benutzer, den Sie aktualisieren möchten.
8. Wählen Sie Aktionen, Benutzer zu Gruppe hinzufügen.
9. Wählen Sie unter Add to group (Zur Gruppe hinzufügen) die Gruppe aus, zu der Sie den Benutzer hinzufügen möchten.
10. Wählen Sie Add user to group (Benutzer zur Gruppe hinzufügen).

Bearbeiten der Mitgliedschaft einer Betriebssystemgruppe mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem sich die Gruppe befindet, die Sie aktualisieren möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Groups (Gruppen).
7. Wählen Sie die Schaltfläche neben der Gruppe, die Sie aktualisieren möchten.
8. Wählen Sie Aktionen, Gruppe ändern.
9. Wählen Sie unter Gruppenmitglieder die Benutzer aus, die Sie hinzufügen oder entfernen möchten.
10. Wählen Sie Modify group (Gruppe ändern).

Löschen eines Benutzers oder einer Gruppe

Löschen eines OS-Benutzerkontos mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem sich das Benutzerkonto befindet, das Sie löschen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Users.
7. Wählen Sie die Schaltfläche neben dem Benutzer, dem Sie löschen möchten.
8. Wählen Sie Aktionen, Lokalen Benutzer löschen.

Löschen einer OS-Gruppe mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem sich die Gruppe befindet, die Sie löschen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Group (Gruppe).
7. Wählen Sie die Schaltfläche neben der Gruppe, die Sie aktualisieren möchten.
8. Wählen Sie Aktionen, Lokale Gruppe löschen.

Verwaltung der Windows-Registrierung auf verwalteten Knoten

Sie können eine Funktion vonFleet Manager, verwenden AWS Systems Manager, um die Registrierung auf Ihren Windows Server verwalteten Knoten zu verwalten. Von der Fleet Manager-Konsole können Sie Registrierungseinträge und -werte erstellen, kopieren, aktualisieren und löschen.

> **Important**
>
> Wir empfehlen, ein Backup der Registry oder einen Snapshot des Amazon Elastic Block Store (Amazon EBS)-Root-Volume zu erstellen, das Ihrem verwalteten Knoten angefügt ist, bevor Sie die Registry ändern. Schwerwiegende Probleme können auftreten, wenn Sie eine falsche Änderung in der Registrierung vornehmen. Aufgrund dieser Probleme müssen Sie möglicherweise das Betriebssystem neu installieren oder das Root-Volume Ihres Knotens anhand eines Snapshots wiederherstellen. AWS garantiert nicht, dass diese Probleme gelöst werden können. Ändern Sie die Registrierung auf eigenes Risiko. Sie sind für alle Registrierungsänderungen verantwortlich und stellen sicher, dass Sie Backups haben.

Erstellen eines Windows-Registrierungsschlüssels oder -Eintrags

Erstellen eines Windows-Registrierungsschlüssel mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen Registry-Schlüssel erstellen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Windows Registry.
6. Wählen Sie den Hive aus, in dem Sie einen neuen Registrierungsschlüssel erstellen möchten, indem Sie den Registry name (Registry-Name) wählen.
7. Wählen Sie Registrierungsschlüssel erstellen.
8. Wählen Sie die Schaltfläche neben dem Registrierungseintrag, in dem Sie einen Schlüssel erstellen möchten.
9. Wählen Sie Create registry key (Registry-Schlüssel erstellen).
10. Geben Sie einen Wert für den Namen des neuen Registrierungsschlüssels ein und wählen Sie Submit (Senden)

Erstellen eines Windows-Registrierungseintrags mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben der Instance, in der Sie einen Registrierungseintrag erstellen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Windows Registry.
6. Wählen Sie den Hive und den darauffolgenden Registrierungsschlüssel aus, in dem Sie einen neuen Registrierungseintrag erstellen möchten, indem Sie den Registry name (Registry-Name) wählen.
7. Wählen Sie Erstellen, Registrierungseintrag erstellen.
8. Geben Sie einen Wert für den Namen des neuen Registrierungseintrags ein.
9. Wählen Sie den Typ des Wertes, den Sie für den Registrierungseintrag erstellen möchten. Weitere Informationen zu Registry-Werttypen finden Sie unter Registry value types (Registry-Wertetypen).
10. Geben Sie einen Wert für den Value (Wert) des neuen Registrierungseintrags ein.

Aktualisieren eines Windows-Registrierungseintrags

Aktualisieren eines Windows-Registrierungseintrags mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen Registry-Eintrag aktualisieren möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Windows Registry.
6. Wählen Sie den Hive und den nachfolgenden Registrierungsschlüssel aus, den Sie aktualisieren möchten, indem Sie das Kontrollkästchen Registry name (Registry-Name) anklicken.
7. Wählen Sie die Schaltfläche neben dem Registrierungseintrag aus, den Sie aktualisieren möchten.

8. Wählen Sie Aktionen, Registrierungseintrag aktualisieren.
9. Geben Sie den neuen Wert für den Value (Wert) des Registrierungseintrags ein.
10. Wählen Sie Aktualisieren.

Löschen eines Windows-Registrierungseintrags oder -schlüssels

Erstellen eines Windows-Registrierungsschlüssel mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen Registry-Schlüssel löschen möchten.
4. Wählen Sie Tools, Windows Registry.
5. Wählen Sie den Hive und den nachfolgenden Registrierungsschlüssel aus, den Sie löschen möchten, indem Sie das Kontrollkästchen Registry name (Registry-Name) anklicken.
6. Wählen Sie die Schaltfläche neben dem Registrierungsschlüssel, den Sie löschen möchten
7. Wählen Sie Aktionen, Registrierungsschlüssel löschen.

Löschen eines Windows-Registrierungseintrags mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen Registry-Eintrag löschen möchten.
4. Wählen Sie die Option View details aus.
5. Wählen Sie Tools, Windows Registry.
6. Wählen Sie den Hive und den nachfolgenden Registrierungsschlüssel, der den Eintrag enthält, aus, den Sie löschen möchten, indem Sie das Kontrollkästchen Registry name (Registry-Name) anklicken.
7. Wählen Sie die Schaltfläche neben dem Registrierungseintrag, den Sie löschen möchten
8. Wählen Sie Aktionen, Registrierungseintrag löschen.

Zugriff auf das Knowledgebase-Portal von Red Hat

Wenn Sie ein Fleet Manager RedHat-Kunde sind AWS Systems Manager, können Sie eine Funktion von verwenden, um auf das Knowledgebase-Portal zuzugreifen. Sie gelten als Red-Hat-Kunde, wenn Sie auf AWSRed Hat Enterprise Linux (RHEL)-Instances ausführen oder RHEL-Services verwenden. Das Knowledgebase-Portal umfasst Binärdateien sowie Wissensaustausch- und Diskussionsforen für Community-Support, die nur von Red Hat lizenzierten Kunden zur Verfügung stehen.

Zusätzlich zu den erforderlichen AWS Identity and Access Management (IAM-) Berechtigungen für Systems Manager und muss der Benutzer oder die RolleFleet Manager, die Sie für den Zugriff auf die Konsole verwenden, der `rhelkb:GetRhelURL` Aktion den Zugriff auf das Knowledgebase-Portal ermöglichen.

Zugreifen auf das Red-Hat-Knowledgebase-Portal

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/systems-manager/ AWS Systems Manager .
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die RHEL-Instance, die Sie verwenden möchten, um sich mit dem Knowledgebase-Portal von Red Hat zu verbinden.
4. Wählen Sie Kontomanagement, Auf Red Hat-Wissensdatenbank zugreifen, um die Red Hat-Wissensdatenbank zu öffnen.

Wenn Sie RHEL on verwenden, AWS um vollständig unterstützte RHEL Workloads auszuführen, können Sie mit Ihren Zugangsdaten auch über die Red Hat Website auf die Red Hat Knowledgebase zugreifen. AWS

Problembehandlung bei der Verfügbarkeit verwalteter Knoten

Für verschiedene AWS Systems Manager Funktionen wie Run Command DistributorSession Manager, und können Sie die verwalteten Knoten, auf denen Sie einen Vorgang ausführen möchten, manuell auswählen. In solchen Fällen zeigt das System, nachdem Sie angegeben haben, dass Sie Knoten manuell auswählen möchten, eine Liste der verwalteten Knoten an, auf denen Sie die Operation ausführen können.

Dieses Thema liefert Informationen zur Diagnose, warum ein verwalteter Knoten, für den Sie bestätigt haben, dass er ausgeführt wird, nicht in Ihren Listen verwalteter Knoten in Systems Manager aufgeführt wird.

Damit ein Knoten von Systems Manager verwaltet und in Listen verwalteter Knoten verfügbar ist, muss er drei primäre Anforderungen erfüllen:

- SSM Agent muss auf dem Knoten installiert sein und mit einem unterstützten Betriebssystem ausgeführt werden.

  > **Note**
  >
  > Einige AWS managed Amazon Machine Images (AMIs) sind so konfiguriert, dass sie Instances mit SSM Agentvorinstallierter Installation starten. (Sie können auch ein benutzerdefiniertes AMI zur Vorinstallation von SSM Agent konfigurieren). Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

- Für Amazon Elastic Compute Cloud (Amazon EC2) -Instances müssen Sie ein AWS Identity and Access Management (IAM-) Instance-Profil an die Instance anhängen. Das Instance-Profil ermöglicht es der Instance, mit dem Systems-Manager-Service zu kommunizieren. Wenn Sie der Instance kein Instance-Profil zuweisen, registrieren Sie sie mit einer Hybrid-Aktivierung, was kein übliches Szenario ist.
- SSM Agent muss in der Lage sein, eine Verbindung zu einem Systems Manager-Endpunkt herzustellen, um sich beim Service zu registrieren. Danach muss der verwaltete Knoten für den Service verfügbar sein, was vom Service bestätigt wird, der alle fünf Minuten ein Signal sendet, um den Zustand der Instance zu überprüfen.
- Nachdem der Status eines verwalteten Knotens mindestens 30 Tage lang `Connection Lost` gewesen ist, wird der Knoten möglicherweise nicht mehr in der Fleet Manager-Konsole aufgeführt. Beheben Sie das Problem, das den Verbindungsverlust verursacht hat, um ihn wieder in die Liste aufzunehmen.

Nachdem Sie überprüft haben, dass ein verwalteter Knoten ausgeführt wird, können Sie den folgenden Befehl verwenden, um zu überprüfen, ob SSM Agent erfolgreich beim Systems-Manager-Service registriert wurde. Dieser Befehl gibt keine Ergebnisse zurück, bis eine erfolgreiche Registrierung stattgefunden hat.

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
```

Wenn die Registrierung erfolgreich war und der verwaltete Knoten jetzt für Systems-Manager-Operationen verfügbar ist, gibt der Befehl ähnliche Ergebnisse wie die folgenden zurück.

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

Wenn die Registrierung noch nicht abgeschlossen wurde oder nicht erfolgreich war, gibt der Befehl ähnliche Ergebnisse wie die folgenden zurück:

```
{
    "InstanceAssociationStatusInfos": []
```

```
}
```

Wenn der Befehl nach etwa 5 Minuten keine Ergebnisse zurückgibt, verwenden Sie die folgenden Informationen, um Probleme mit Ihren verwalteten Knoten zu beheben.

Themen

- Lösung 1: Überprüfen Sie, ob SSM Agent installiert ist und auf dem verwalteten Knoten ausgeführt wird
- Lösung 2: Überprüfen Sie, ob ein IAM-Instance-Profil für die Instance angegeben wurde (nur EC2-Instances)
- Lösung 3: Überprüfen der Konnektivität des Service-Endpunkts
- Lösung 4: Überprüfen der Unterstützung des Zielbetriebssystems
- Lösung 5: Stellen Sie sicher, dass Sie in derselben AWS-Region Amazon EC2 EC2-Instance arbeiten
- Lösung 6: Überprüfen Sie die Proxy-Konfiguration, die Sie auf SSM Agent in Ihrem verwalteten Knoten angewendet haben
- Lösung 7: Installieren eines TLS-Zertifikats auf verwalteten Instances
- Problembehandlung bei der Verfügbarkeit von verwalteten Knoten mit ssm-cli

Lösung 1: Überprüfen Sie, ob SSM Agent installiert ist und auf dem verwalteten Knoten ausgeführt wird

Stellen Sie sicher, dass die neueste Version von SSM Agent auf dem verwalteten Knoten installiert ist und ausgeführt wird.

Informationen zum Feststellen, ob SSM Agent installiert ist und auf einem verwalteten Knoten ausgeführt wird, finden Sie unter Prüfen des SSM Agent-Status und Starten des Agenten.

Um SSM Agent auf einem verwalteten Knoten zu installieren bzw. deinstallieren, siehe folgende Themen:

- Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux
- Wie installiert man das SSM Agent auf Hybrid-Linux-Knoten
- Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Windows Server
- Wie installiert man den SSM Agent auf Hybrid-Windows-Knoten

Lösung 2: Überprüfen Sie, ob ein IAM-Instance-Profil für die Instance angegeben wurde (nur EC2-Instances)

Vergewissern Sie sich bei Amazon Elastic Compute Cloud (Amazon EC2)-Instances, ob die Instance mit einem AWS Identity and Access Management (IAM)-Instance-Profil konfiguriert ist, das es der Instance erlaubt, mit der Systems-Manager-API zu kommunizieren. Stellen Sie außerdem sicher, dass Ihr Benutzer über eine IAM-Vertrauensrichtlinie verfügt, die es Ihrem Benutzer ermöglicht, mit der Systems-Manager-API zu kommunizieren.

> **Note**
>
> On-Premises-Server, Edge-Geräte und virtuelle Maschinen (VMs) verwenden eine IAM-Servicerolle anstelle eines Instance-Profils. Weitere Informationen finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Servicerolle in Hybrid- und Multicloud-Umgebungen.

So stellen Sie fest, ob ein Instance-Profil mit den nötigen Berechtigungen einer EC2-Instance angefügt ist

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die Instance, die nach einem Instance-Profil überprüft werden soll.
4. Suchen Sie auf der Registerkarte Description (Beschreibung) im unteren Bereich die IAM-Rolle und wählen Sie den Namen der Rolle.
5. Vergewissern Sie sich auf der Seite Summary des Instance-Profils auf der Registerkarte Permissions (Berechtigungen), dass unter den Berechtigungsrichtlinien `AmazonSSMManagedInstanceCore` aufgeführt ist.

   Wenn stattdessen eine benutzerdefinierte Richtlinie verwendet wird, stellen Sie sicher, dass sie dieselben Berechtigungen wie `AmazonSSMManagedInstanceCore` bereitstellt.

   Öffnen Sie `AmazonSSMManagedInstanceCore` in der Konsole

   Informationen zu anderen Richtlinien, die an ein Instanzprofil für Systems Manager angehängt werden können, finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

Lösung 3: Überprüfen der Konnektivität des Service-Endpunkts

Stellen Sie sicher, dass die Instance eine Verbindung zu den Systems Manager Service-Endpunkten hat. Diese Konnektivität wird entweder durch das Erstellen und Konfigurieren von VPC-Endpunkten für Systems Manager oder durch die Genehmigung von ausgehenden HTTPS-Datenverkehr (Port 443) zu den Service-Endpunkten bereitgestellt.

Bei Amazon EC2 EC2-Instances wird der Systems Manager Manager-Serviceendpunkt für AWS-Region die Instance verwendet, um die Instance zu registrieren, wenn Ihre Virtual Private Cloud (VPC) -Konfiguration ausgehenden Datenverkehr zulässt. Wenn die VPC-Konfiguration, in der die Instance gestartet wurde, jedoch keinen ausgehenden Datenverkehr zulässt und Sie diese Konfiguration nicht ändern können, um Konnektivität zu den öffentlichen Service-Endpunkten zu erlauben, müssen Sie stattdessen Schnittstellenendpunkte für Ihre VPC konfigurieren.

Weitere Informationen finden Sie unter Verbessern der Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

Lösung 4: Überprüfen der Unterstützung des Zielbetriebssystems

Stellen Sie sicher, dass die ausgewählte Operation für den Typ von verwalteten Knoten ausgeführt werden kann, den Sie in der Liste erwarten. Einige Systems Manager-Vorgänge können nur auf Windows-Instanceen oder nur auf Linux-Instances abzielen. Zum Beispiel können die Systems Manager (SSM) Dokumente `AWS-InstallPowerShellModule` und `AWS-ConfigureCloudWatch` nur auf Windows-Instances ausgeführt werden. Wenn Sie auf der Seite Run a command (Ausführen eines Befehls) eines dieser Dokumente auswählen und die Option Choose instances manually (Instancen manuell auswählen) wählen, werden nur Ihre Windows-Instances aufgelistet und stehen zur Auswahl.

Lösung 5: Stellen Sie sicher, dass Sie in derselben AWS-Region Amazon EC2 EC2-Instance arbeiten

Amazon EC2 EC2-Instances werden in bestimmten Regionen erstellt und sind verfügbar AWS-Regionen, z. B. in der Region USA Ost (Ohio) (us-east-2) oder Europa (Irland) (eu-west-1). Stellen Sie sicher, dass Sie in derselben AWS-Region Amazon EC2 EC2-Instance arbeiten, mit der Sie arbeiten möchten. Weitere Informationen dazu erhalten Sie unter Choosing a Region (Region wählen) in Getting Started with the AWS Management Console.

Lösung 6: Überprüfen Sie die Proxy-Konfiguration, die Sie auf SSM Agent in Ihrem verwalteten Knoten angewendet haben

Überprüfen Sie, ob die Proxy-Konfiguration, die Sie auf SSM Agent in Ihrem verwalteten Knoten angewendet haben, korrekt ist. Wenn die Proxy-Konfiguration falsch ist, kann der Knoten keine Verbindung zu den erforderlichen Service-Endpunkten herstellen, oder Systems Manager identifiziert möglicherweise das Betriebssystem des verwalteten Knotens falsch. Weitere Informationen finden Sie unter Konfiguration SSM Agent für die Verwendung eines Proxys auf Linux-Knoten und Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances.

Lösung 7: Installieren eines TLS-Zertifikats auf verwalteten Instances

Auf jeder verwalteten Instance, die Sie verwenden, muss ein Transport Layer Security (TLS) - Zertifikat installiert sein. AWS Systems Manager AWS-Services Verwenden Sie diese Zertifikate, um Anrufe an andere AWS-Services zu verschlüsseln.

Auf jeder Amazon-EC2-Instance, die aus einem Amazon Machine Image (AMI) erstellt wurde, ist standardmäßig bereits ein TLS-Zertifikat installiert. Die meisten modernen Betriebssysteme enthalten das erforderliche TLS-Zertifikat von Amazon Trust Services-Zertifizierungsstellen in ihrem Trust Store.

Um zu überprüfen, ob das erforderliche Zertifikat auf Ihrer Instance installiert ist, führen Sie den folgenden Befehl basierend auf dem Betriebssystem Ihrer Instance aus. Achten Sie darauf, den *`regionalen`* Teil der URL durch den Teil zu ersetzen, AWS-Region in dem sich Ihre verwaltete Instanz befindet.

Linux & macOS

```
curl -L https://ssm.region.amazonaws.com
```

Windows

```
Invoke-WebRequest -Uri https://ssm.region.amazonaws.com
```

Der Befehl sollte einen `UnknownOperationException`-Fehler zurückgeben. Wenn Sie stattdessen eine SSL/TLS-Fehlermeldung erhalten, ist das erforderliche Zertifikat möglicherweise nicht installiert.

Wenn Sie feststellen, dass die erforderlichen CA-Zertifikate von Amazon Trust Services nicht auf Ihren Basisbetriebssystemen, auf Instances installiert sind, AMIs die nicht von Amazon bereitgestellt

wurden, oder auf Ihren eigenen lokalen Servern und VMs, müssen Sie ein Zertifikat von Amazon Trust Services installieren und zulassen oder AWS Certificate Manager (ACM) verwenden, um Zertifikate für einen unterstützten integrierten Service zu erstellen und zu verwalten.

Auf jeder Ihrer verwalteten Instances muss eines der folgenden Transport Layer Security (TLS)-Zertifikate installiert sein.

- Amazon Root CA 1
- Starfield Services Root Certificate Authority – G2
- Starfield Class 2 Certificate Authority

Informationen zur Verwendung von ACM finden Sie im AWS Certificate Manager -Benutzerhandbuch.

Wenn Zertifikate in Ihrer Datenverarbeitungsumgebung von einem Gruppenrichtlinienobjekt (GPO) verwaltet werden, dann müssen Sie möglicherweise die Gruppenrichtlinie so konfigurieren, dass eines dieser Zertifikate enthalten ist.

Weitere Informationen zu den Amazon Root- und Starfield-Zertifikaten finden Sie im Blogbeitrag How to Prepare for AWS's Move to Its Own Certificate Authority.

Problembehandlung bei der Verfügbarkeit von verwalteten Knoten mit **`ssm-cli`**

Die `ssm-cli` ist ein eigenständiges Befehlszeilentool, das in der SSM Agent-Installation enthalten ist. Wenn Sie SSM Agent 3.1.501.0 oder höher auf einem Computer installieren, können Sie `ssm-cli` Befehle auf diesem Computer ausführen. Die Ausgabe dieser Befehle hilft Ihnen festzustellen, ob die Maschine die Mindestanforderungen für eine Amazon EC2-Instance oder eine non-EC2-Maschine erfüllt AWS Systems Manager, die von verwaltet werden soll, und daher zu Listen verwalteter Knoten in Systems Manager hinzugefügt wurde. (SSM Agent Version 3.1.501.0 wurde im November 2021 veröffentlicht.)

Mindestanforderungen

Damit eine Amazon EC2-Instance oder ein non-EC2-Computer von verwaltet werden kann AWS Systems Manager und in Listen verwalteter Knoten verfügbar ist, muss sie drei primäre Anforderungen erfüllen:

- SSM Agent muss auf einer Maschine mit einem unterstützten Betriebssystem installiert sein und laufen.

Einige AWS von verwaltete Amazon Machine Images (AMIs) für EC2 sind so konfiguriert, dass Instances mit SSM Agent vorinstalliertem gestartet werden. (Sie können auch ein benutzerdefiniertes AMI zur Vorinstallation von SSM Agent konfigurieren). Weitere Informationen finden Sie unter Finden Sie AMIs mit dem SSM Agent vorinstallierten.

- Ein AWS Identity and Access Management (IAM)-Instance-Profil (für EC2-Instances) oder eine IAM-Servicerolle (für non-EC2-Maschinen), die die erforderlichen Berechtigungen für die Kommunikation mit dem Systems Manager-Service bereitstellt, muss an die Maschine angehängt werden.
- SSM Agent muss in der Lage sein, eine Verbindung zu einem Systems-Manager-Endpunkt herzustellen, um sich beim Service anzumelden. Danach muss der verwaltete Knoten für den Service verfügbar sein, was vom Service bestätigt wird, der alle fünf Minuten ein Signal sendet, um den Zustand des verwalteten Knoten zu überprüfen.

Vorkonfigurierte Befehle in **`ssm-cli`**

Es sind vorkonfigurierte Befehle enthalten, die die erforderlichen Informationen sammeln, um Ihnen bei der Diagnose zu helfen, warum eine Maschine, von der Sie bestätigt haben, dass sie läuft, nicht in Ihrer Liste der verwalteten Knoten in Systems Manager enthalten ist. Diese Befehle werden ausgeführt, wenn Sie die `get-diagnostics`-Option angeben.

Führen Sie auf der Maschine den folgenden Befehl aus, um `ssm-cli` für die Problembehebung in Bezug auf die Verfügbarkeit der verwalteten Knoten zu verwenden.

Linux & macOS

```
ssm-cli get-diagnostics --output table
```

Windows

Auf Windows Server-Maschinen müssen Sie zum `C:\Program Files\Amazon\SSM`-Verzeichnis navigieren, bevor Sie den Befehl ausführen.

```
ssm-cli.exe get-diagnostics --output table
```

PowerShell

Auf Windows Server-Maschinen müssen Sie zum `C:\Program Files\Amazon\SSM`-Verzeichnis navigieren, bevor Sie den Befehl ausführen.

```
.\ssm-cli.exe get-diagnostics --output table
```

Der Befehl liefert eine Ausgabe in Form einer Tabelle ähnlich der folgenden.

> **Note**
>
> Konnektivitätsprüfungen zu den `monitoring` Endpunkten `ssmmessages`, `s3`, `logs`, und beziehen sich auf zusätzliche optionale Funktionen wie `kms`, Session Manager die sich bei Amazon Simple Storage Service (Amazon S3) oder Amazon CloudWatch Logs anmelden und AWS Key Management Service (AWS KMS)-Verschlüsselung verwenden können.

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
                                      #
##################################################################################################
```

Die folgende Tabelle enthält zusätzliche Details für jede der von `ssm-cli` ausgeführten Überprüfungen.

`ssm-cli`-Diagnoseprüfungen

Check	Details
Amazon-EC2-Instance-Metadaten-Service	Gibt an, ob der verwaltete Knoten den Metadaten-Service erreichen kann. Ein fehlgeschlagener Test deutet auf ein Konnektivitätsproblem zu `http://169.254.169.254` hin, das durch das lokale Routing, den Proxy oder die Firewall- und Proxy-Konfigurationen des Betriebssystems verursacht werden kann.
Hybrid-Instance-Registrierung	Zeigt an, ob SSM Agent über eine Hybrid-Aktivierung registriert ist.
Konnektivität mit `ssm`-Endpunkt	Zeigt an, ob der Knoten in der Lage ist, die Service-Endpunkte für Systems Manager auf TCP-Port 443 zu erreichen. Ein fehlgeschlagener Test weist auf Verbindungsprobleme mit hin, je `https://ssm.` *`region`*`.amazonaws.com` nachdem AWS-Region , wo sich der Knoten befindet. Konnektivitätsprobleme können durch die VPC-Konfiguration verursacht werden, einschließlich Sicherheitsgruppen, Netzwerkzugriffs-Kontrolllisten, Routing-Tabellen oder OS-Firewalls und Proxys.
Konnektivität mit `ec2messages` -Endpunkt	Zeigt an, ob der Knoten in der Lage ist, die Service-Endpunkte für Systems Manager auf TCP-Port 443 zu erreichen. Ein fehlgeschlagener Test weist auf Verbindungsprobleme mit hin, je `https://ec2messages.` *`region`*`.amazonaws.com` nachdem

Check	Details
	AWS-Region , wo sich der Knoten befindet. Konnektivitätsprobleme können durch die VPC-Konfiguration verursacht werden, einschließlich Sicherheitsgruppen, Netzwerkzugriffs-Kontrollli sten, Routing-Tabellen oder OS-Firewalls und Proxys.
Konnektivität mit `ssmmessages` -Endpunkt	Zeigt an, ob der Knoten in der Lage ist, die Service-Endpunkte für Systems Manager auf TCP-Port 443 zu erreichen. Ein fehlgesch lagener Test weist auf Verbindungsproblem e mit hin, je `https://ssmmessage s.` *`region`*`.amazonaws.com` nachdem AWS-Region , wo sich der Knoten befindet. Konnektivitätsprobleme können durch die VPC-Konfiguration verursacht werden, einschließlich Sicherheitsgruppen, Netzwerkzugriffs-Kontrollli sten, Routing-Tabellen oder OS-Firewalls und Proxys.
Konnektivität mit `s3`-Endpunkt	Zeigt an, ob der Knoten in der Lage ist, den Service-Endpunkt für Amazon Simple Storage Service auf TCP-Port 443 zu erreichen. Ein fehlgeschlagener Test weist auf Verbindungsprobleme mit hin, je `https://s3.` *`region`*`.amazonaws.com` nachdem AWS-Region , wo sich der Knoten befindet. Eine Verbindung zu diesem Endpunkt ist nicht erforderlich, damit ein Knoten in Ihrer Liste der verwalteten Knoten erscheint.

Check	Details
Konnektivität mit `kms`-Endpunkt	Gibt an, ob der Knoten den Service-Endpunkt für AWS Key Management Service auf TCP-Port 443 erreichen kann. Ein fehlgeschlagener Test weist auf Verbindungsprobleme mit hin, je `https://kms.` *`region`*`.amazonaws.com` nachdem AWS-Region , wo sich der Knoten befindet. Eine Verbindung zu diesem Endpunkt ist nicht erforderlich, damit ein Knoten in Ihrer Liste der verwalteten Knoten erscheint.
Konnektivität mit `logs`-Endpunkt	Gibt an, ob der Knoten den Service-Endpunkt für Amazon CloudWatch Logs auf TCP-Port 443 erreichen kann. Ein fehlgeschlagener Test weist auf Verbindungsprobleme mit hin, je `https://logs.` *`region`*`.amazonaws.com` nachdem AWS-Region , wo sich der Knoten befindet. Eine Verbindung zu diesem Endpunkt ist nicht erforderlich, damit ein Knoten in Ihrer Liste der verwalteten Knoten erscheint.
Konnektivität mit `monitoring` -Endpunkt	Gibt an, ob der Knoten den Service-Endpunkt für Amazon CloudWatch auf TCP-Port 443 erreichen kann. Ein fehlgeschlagener Test weist auf Verbindungsprobleme mit hin, je `https://monitoring.` *`region`*`.amazonaws.com` nachdem AWS-Region , wo sich der Knoten befindet. Eine Verbindung zu diesem Endpunkt ist nicht erforderlich, damit ein Knoten in Ihrer Liste der verwalteten Knoten erscheint.

Check	Details
AWS Erweitern Sie im angezeigten Detailber eich die Option	Zeigt an, ob SSM Agent über die erforderlichen Anmeldeinformationen auf der Grundlage des IAM-Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (für Nicht-EC2-Maschinen) verfügt, die mit der Maschine verbunden sind. Ein fehlgeschlagener Test zeigt an, dass der Maschine kein IAM-Instance-Profil oder keine IAM-Servicerolle zugeordnet ist, oder dass sie nicht die erforderlichen Berechtigungen für Systems Manager enthält.
Agent-Service	Zeigt an, ob der SSM Agent-Service läuft und ob der Service als Root für Linux oder macOS bzw. als SYSTEM für Windows Server läuft. Ein fehlgeschlagener Test zeigt an, dass der SSM Agent-Service nicht ausgeführt oder nicht als Root oder SYSTEM ausgeführt wird.
Proxykonfiguration	Gibt an, ob SSM Agent konfiguriert ist, einen Proxy zu verwenden.
Sysprep-Image-Status (nur Windows)	Zeigt den Status von `Sysprep` auf dem Knoten an. SSM Agent wird nicht auf dem Knoten gestartet, wenn der `Sysprep`-Status einen anderen Wert als `IMAGE_STATE_COMPLETE` hat.
SSM Agent-Version	Zeigt an, ob die neueste verfügbare Version von SSM Agent installiert ist.

AWS Systems Manager-Compliance

Sie können Compliance, eine Funktion von AWS Systems Manager, verwenden, um Ihre Flotte verwalteter Knoten auf Patch-Compliance und Konfigurationsinkonsistenzen zu scannen. Sie können Daten aus mehreren AWS-Konten und Regionen sammeln und aggregieren und dann

bestimmte Ressourcen aufschlüsseln, die nicht konform sind. Standardmäßig zeigt Compliance aktuelle Compliance-Daten zum Patchen in Patch Manager und Assoziationen State Manager an. (Patch Manager und State Manager sind ebenfalls beide Funktionen von AWS Systems Manager.) Um mit Compliance zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im linken Navigationsbereich Compliance.

Patch-Compliance-Daten von Patch Manager können an gesendet werden AWS Security Hub. Mit dem Security Hub erhalten Sie einen umfassenden Überblick über Ihre Sicherheitswarnungen und den Compliance-Status mit hoher Priorität. Er überwacht auch den Patching-Status Ihrer Flotte. Weitere Informationen finden Sie unter Integrieren Patch Manager mit AWS Security Hub.

Compliance bietet die folgenden zusätzlichen Vorteile und Funktionen:

- Zeigen Sie den Compliance-Verlauf und die Änderungsnachverfolgung für Patch Manager-Patching-Daten und State Manager-Zuordnungen mithilfe von AWS Config an.
- Passen Sie Compliance an, um Ihre eigenen Compliance-Typen auf Grundlage Ihrer IT- oder Business-Anforderungen zu erstellen.
- Beheben Sie Probleme mithilfe von Run Command, einer anderen Funktion von AWS Systems ManagerState Manager, oder Amazon EventBridge.
- Portieren Sie Daten an Amazon Athena und Amazon QuickSight , um flottenweite Berichte zu generieren.

EventBridge -Unterstützung

Diese Systems Manager-Funktion wird als Ereignistyp in Amazon- EventBridge Regeln unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager.

Chef InSpec-Integration

Systems Manager ist in integriertChef InSpec. InSpec ist ein Open-Source-Laufzeit-Framework, mit dem Sie lesbare Profile auf GitHub oder Amazon Simple Storage Service (Amazon S3) erstellen können. Anschließend können Sie Systems Manager verwenden, um Compliance-Scans auszuführen und konforme bzw. nicht konforme Knoten anzuzeigen. Weitere Informationen finden Sie unter Verwenden von Chef InSpec Profilen mit Systems Manager Compliance.

Preisgestaltung

Compliance wird ohne Zusatzkosten angeboten. Sie zahlen nur für die AWS Ressourcen, die Sie tatsächlich nutzen.

Inhalt

- Erste Schritte mit Compliance
- Erstellen einer Ressource Data Sync für Compliance
- Arbeiten mit Compliance
- Löschen einer Ressource Data Sync für Compliance
- Beheben von Compliance-Problemen mithilfe von EventBridge
- Compliance-Walkthrough (AWS CLI)

Erste Schritte mit Compliance

Um mit Compliance, eine Funktion von AWS Systems Manager, zu beginnen, führen Sie die folgenden Aufgaben aus.

Aufgabe	Weitere Informationen
Compliance funktioniert mit Patch-Daten in Patch Manager und Assoziationen in State Manager. (Patch Manager und State Manager sind auch beide Funktionen von AWS Systems Manager.) Compliance funktioniert auch mit benutzerdefinierten Kompatibilitätstypen auf verwalteten Knoten, die mit Systems Manager verwaltet werden. Überprüfen Sie, ob Sie die Einrichtungsanforderungen für Ihre Instances der Amazon Elastic Compute Cloud (Amazon EC2) und Nicht-EC2-Geräte in einer Hybrid- und Multi-Cloud-Umgebung erfüllt haben.	Einrichten AWS Systems Manager
Aktualisieren Sie Systems Manager SSM Agent (SSM Agent) auf Ihren verwalteten Knoten auf die neueste Version.	Arbeiten mit SSM Agent

Aufgabe	Weitere Informationen
Wenn Sie die Patch-Compliance überwachen möchten, überprüfen Sie die Konfiguration des Patch Manager. Sie müssen Patch-Vorgänge mit dem Patch Manager durchführen, bevor die Patch-Compliance-Daten von Compliance angezeigt werden können.	AWS Systems Manager Patch Manager
Wenn Sie die Zuordnungs-Compliance überwachen möchten, überprüfen Sie, dass Sie State Manager-Zuordnungen erstellt haben. Sie müssen Zuordnungen erstellen, bevor die Daten zur Zuordnungs-Compliance von Compliance angezeigt werden können.	AWS Systems Manager State Manager
(Optional) Konfigurieren Sie das System, um den Compliance-Verlauf und die Änderungs nachverfolgung anzuzeigen.	Anzeigen von Compliance-Konfigurationsverlauf und Änderungsnachverfolgung
(Optional) Erstellen Sie benutzerdefinierte Compliance-Typen.	Compliance-Walkthrough (AWS CLI)
(Optional) Erstellen Sie eine Resource Data Sync zur Aggregation aller Compliance-Daten in einem Amazon Simple Storage Service (Amazon S3)-Bucket.	Erstellen einer Ressource Data Sync für Compliance

Erstellen einer Ressource Data Sync für Compliance

Sie können die Funktion zur Synchronisierung von Ressourcendaten verwenden AWS Systems Manager , um Compliance-Daten von all Ihren verwalteten Knoten an einen Amazon Simple Storage Service (Amazon S3) -Ziel-Bucket zu senden. Wenn Sie die Synchronisierung erstellen, können Sie verwaltete Knoten aus mehreren AWS-Konten AWS-Regionen, und Ihrer Hybrid- und Multi-Cloud-Umgebung angeben. Resource Data Sync aktualisiert die Daten dann automatisch, sobald neue Compliance-Daten erfasst werden. Da alle Compliance-Daten in einem Ziel-S3-Bucket gespeichert sind, können Sie Dienste wie Amazon Athena und Amazon verwenden, QuickSight

um die aggregierten Daten abzufragen und zu analysieren. Resource Data Sync muss einmalig für Compliance konfiguriert werden.

Führen Sie die folgenden Schritte aus, um mit der AWS Management Console eine Ressource Data Sync für Compliance zu erstellen.

So erstellen und konfigurieren Sie einen S3-Bucket für die Synchronisierung von Ressourcendaten (Konsole)

1. Öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie ein Bucket zum Speichern der zusammengefassten Compliance-Daten. Weitere Informationen finden Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service. Notieren Sie sich den Namen des Buckets und den Ort, AWS-Region an dem Sie ihn erstellt haben.
3. Öffnen Sie den Bucket, wählen Sie die Registerkarte Permissions (Berechtigungen) und anschließend die Option Bucket Policy (Bucket-Richtlinie).
4. Kopieren Sie die folgende Bucket-Richtlinie in den Richtlinien-Editor. Ersetzen Sie DOC-EXAMPLE-BUCKET und *Account-ID* durch den Namen des von Ihnen erstellten S3-Buckets und eine gültige ID. AWS-Konto Sie können auch das *Bucket-Prefix* durch den Namen eines Amazon S3-Präfixes (Unterverzeichnis) ersetzen. Wenn Sie kein Präfix erstellt haben, entfernen Sie das *Bucket-Prefix*/ aus dem ARN in der Richtlinie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SSMBucketPermissionsCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        },
        {
            "Sid": " SSMBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
```

```
            "Action": "s3:PutObject",
            "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/Bucket-Prefix/*/
accountid=Account_ID_number/*"],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Erstellen einer Resource Data Sync

1. Öffnen Sie AWS Systems Manager die Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie Account management (Kontoverwaltung), Resource Data Syncs und dann Create resource data sync (Resource Data Sync erstellen)
4. Geben Sie im Feld Sync name einen Namen für die Synchronisierungskonfiguration ein.
5. Geben Sie im Feld Bucket name (Bucket-Name) den Namen des zu Beginn dieses Vorgangs erstellten Amazon S3-Buckets an.
6. (Optional) Geben Sie im Feld Bucket-Präfix den Namen eines S3-Bucket-Präfixes (Unterverzeichnis) an.
7. Wählen Sie im Feld Bucket-Region die Option Diese Region aus, wenn sich der erstellte S3-Bucket in der aktuellen AWS-Region befindet. Wenn sich der Bucket in einer anderen Region befindet AWS-Region, wählen Sie „Andere Region“ und geben Sie den Namen der Region ein.

   > **Note**
   >
   > Wenn sich die Synchronisierung und der Ziel-S3-Bucket in verschiedenen Regionen befinden, müssen Sie möglicherweise Gebühren für die Datenübertragung zahlen. Weitere Informationen finden Sie unter Amazon S3 – Preise.

8. Wählen Sie Create (Erstellen) aus.

Arbeiten mit Compliance

Compliance, eine Fähigkeit von AWS Systems Manager, sammelt und meldet Daten über den Status von Patches in Patch Manager Patching und Assoziationen in. State Manager (Patch Managerund State Manager sind auch beide Funktionen von AWS Systems Manager.) Compliance berichtet auch zu benutzerdefinierten Compliance-Typen, die Sie für Ihre verwalteten Knoten angegeben haben. Dieser Abschnitt enthält Details über jeden dieser Compliance-Typen sowie Informationen zum Anzeigen von Systems Manager-Compliance-Daten. Dieser Abschnitt enthält auch Informationen zum Anzeigen des Compliance-Verlaufs und der Änderungsnachverfolgung.

> **Note**
>
> Systems Manager lässt sich in integrieren Chef InSpec. InSpec ist ein Open-Source-Runtime-Framework, mit dem Sie menschenlesbare Profile auf GitHub Amazon Simple Storage Service (Amazon S3) erstellen können. Anschließend können Sie Systems Manager verwenden, um Compliance-Scans auszuführen und konforme und nicht konforme Instances anzuzeigen. Weitere Informationen finden Sie unter Verwenden von Chef InSpec Profilen mit Systems Manager Compliance.

Info zu Patch Compliance

Nachdem Sie mit Patch Manager in Ihren Instances installiert haben, stehen Ihnen die Compliance-Statusinformationen sofort in der Konsole oder in der Antwort auf AWS Command Line Interface (AWS CLI)-Befehle oder entsprechenden Systems Manager-API-Vorgängen zur Verfügung.

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten.

Informationen zu State Manager-Zuordnungs-Compliance

Nachdem Sie eine oder mehrere State Manager Zuordnungen erstellt haben, stehen Ihnen Informationen zum Compliance-Status sofort in der Konsole oder als Reaktion auf AWS CLI Befehle oder entsprechende Systems Manager Manager-API-Operationen zur Verfügung. Für Zuordnungen zeigt Compliance den Status `Compliant` oder `Non-compliant` und den der Zuordnung zugewiesenen Schweregrad an, z. B. `Critical` oder `Medium`.

Informationen zu benutzerdefinierter Compliance

Einem verwalteten Knoten können Compliance-Metadaten zugewiesen werden. Diese Metadaten können anschließend mit anderen Compliance-Daten für Compliance-Berichte zusammengefasst werden. Beispiel: Ihr Unternehmen führt die Versionen 2.0, 3.0 und 4.0 von Software X auf Ihren verwalteten Knoten aus. Das Unternehmen möchte Version 4.0 zum Standard machen. Das bedeutet, dass Instances mit Versionen 2.0 und 3.0 nicht konform sind. Mithilfe des PutComplianceItemsAPI-Vorgangs können Sie explizit angeben, auf welchen verwalteten Knoten ältere Versionen von Software X ausgeführt werden. Sie können Compliance-Metadaten nur mithilfe der SDKs AWS CLI AWS Tools for Windows PowerShell, oder zuweisen. Mit dem folgenden CLI-Beispielbefehl werden einer verwalteten Instance Compliance-Metadaten zugewiesen und der Compliance-Typ im benötigten Format angegeben `Custom:`. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

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**
>
> Der `ResourceType`-Parameter unterstützt nur `ManagedInstance`. Wenn Sie einem AWS IoT Greengrass -Core-Gerät benutzerdefinierte Compliance hinzufügen, müssen Sie einen `ResourceType` von `ManagedInstance` angeben.

Compliance-Manager können daraufhin Zusammenfassungen anzeigen oder Berichte über nicht konforme verwaltete Knoten erstellen. Sie können einem verwalteten Knoten maximal 10 verschiedene benutzerdefinierte Compliance-Typen zuweisen.

Ein Beispiel für die Erstellung eines benutzerdefinierten Compliance-Typs und zum Anzeigen von Compliance-Daten finden Sie unter Compliance-Walkthrough (AWS CLI).

Anzeigen aktueller Compliance-Daten

In diesem Abschnitt wird beschrieben, wie Sie Compliance-Daten in der Systems Manager-Konsole und mithilfe der AWS CLI anzeigen. Weitere Informationen zum Anzeigen des Patch- und Zuordnungs-Compliance-Verlaufs und der Änderungsnachverfolgung finden Sie unter Anzeigen von Compliance-Konfigurationsverlauf und Änderungsnachverfolgung.

Themen

- Anzeigen aktueller Compliance-Daten (Konsole)
- Anzeigen aktueller Compliance-Daten (AWS CLI)

Anzeigen aktueller Compliance-Daten (Konsole)

Verwenden Sie die folgenden Verfahren, um Compliance-Daten in der Systems Manager-Konsole anzuzeigen.

So zeigen Sie aktuelle Compliance-Berichte in der Systems Manager-Konsole an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im linken Navigationsbereich Compliance.
3. Wählen Sie im Abschnitt Compliance dashboard filtering (Compliance-Dashboard-Filtrierung) eine Option zum Filtern von Compliance-Daten aus. Der Abschnitt Compliance resources

summary (Zusammenfassung der Compliance-Ressourcen) zeigt die Anzahl der Compliance-Daten basierend auf dem von Ihnen ausgewählten Filter an.

4. Um weitere detaillierte Informationen zu einer Ressource zu erhalten, scrollen Sie nach unten zum Bereich Details overview for resources (Detailübersicht für Ressourcen) und wählen Sie die ID eines verwalteten Knotens.
5. Wählen Sie auf der Detailseite Instance ID (Instance-ID) oder Name die Registerkarte Configuration compliance (Konfigurations-Compliance), um den detaillierten Bericht zur Konfigurations-Compliance anzuzeigen.

> **Note**
>
> Weitere Informationen zum Beheben von Compliance-Problemen finden Sie unter Beheben von Compliance-Problemen mithilfe von EventBridge.

Anzeigen aktueller Compliance-Daten (AWS CLI)

Mithilfe der folgenden AWS CLI Befehle können Sie Zusammenfassungen der Kompatibilitätsdaten für Patches, Verknüpfungen und benutzerdefinierte Kompatibilitätstypen im in der AWS CLI anzeigen.

list-compliance-summaries

Gibt eine Übersichtszahl der konformen und nicht konformen Zuordnungs-Statusarten entsprechend der angegebenen Filter zurück. (API:) ListComplianceSummaries

list-resource-compliance-summaries

Gibt eine Übersichtszahl auf Ressourcenebene zurück. Die Übersicht umfasst Informationen über konforme und nicht konforme Statusarten und die detaillierte Anzahl des Schweregrads von Compliance-Elementen entsprechend den festgelegten Filterkriterien. (API: ListResourceComplianceSummaries)

Sie können zusätzliche Compliance-Daten für das Einspielen von Patches mit den folgenden AWS CLI -Befehlen anzeigen.

describe-patch-group-state

Gibt allgemeine zusammengefasste Patch-Compliance-Statusarten für eine Patch-Gruppe zurück. (API: DescribePatchGroupState)

describe-instance-patch-states-for-patch-group

Gibt den allgemeinen Patch-Status für die Instances in der angegebenen Patch-Gruppe zurück. (API: DescribeInstancePatchStatesForPatchGroup)

> **Note**
>
> Eine Veranschaulichung der Konfiguration von Patches und der Anzeige von Informationen zur Patch-Konformität mithilfe von finden Sie unterAnleitung: Patchen einer Serverumgebung (AWS CLI). AWS CLI

Anzeigen von Compliance-Konfigurationsverlauf und Änderungsnachverfolgung

Standardmäßig werden von Systems-Manager-Compliance die aktuellen Patch-Vorgänge und Zuordnungs-Compliance-Daten für Ihre verwalteten Knoten angezeigt. Sie können den Verlauf der Einhaltung von Patches und Zuordnungen sowie die Änderungsnachverfolgung anzeigen, indem Sie AWS Config AWS Config bietet einen detaillierten Überblick über die Konfiguration der AWS Ressourcen in Ihrem AWS-Konto. Dazu gehört auch, wie die Ressourcen jeweils zueinander in Beziehung stehen und wie sie in der Vergangenheit konfiguriert wurden, damit Sie sehen können, wie sich die Konfigurationen und Beziehungen im Laufe der Zeit verändern. Zum Anzeigen von Patch-Einspielungen, des Zuordnungs-Compliance-Verlaufs und der Änderungsnachverfolgung müssen Sie die folgenden Ressourcen in AWS Config aktivieren:

- `SSM:PatchCompliance`
- `SSM:AssociationCompliance`

Weitere Informationen dazu, wie Sie diese spezifischen Ressourcen in AWS Config auswählen und konfigurieren, finden Sie unter Selecting Which Resources AWS Config Records im AWS Config - Entwicklerleitfaden.

> **Note**
>
> Informationen zur AWS Config Preisgestaltung finden Sie unter Preise.

Löschen einer Ressource Data Sync für Compliance

Wenn Sie Compliance nicht mehr zum Anzeigen von AWS Systems Manager Compliance-Daten verwenden möchten, empfehlen wir außerdem, die für die Erfassung von Compliance-Daten verwendeten Ressourcendatensynchronisationen zu löschen.

Löschen eines Compliance Resource Data Sync

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Klicken Sie auf Account management (Kontenverwaltung), Resource data syncs.
4. Wählen Sie eine Synchronisierung aus der Liste aus.

> **Important**
>
> Stellen Sie sicher, dass Sie die für Compliance verwendete Synchronisierung auswählen. Systems Manager unterstützt die Synchronisierung von Ressourcendaten für mehrere Funktionen. Wenn Sie die falsche Synchronisierung wählen, können Sie die Datenaggregation für Systems Manager Explorer oder Systems Manager Inventory unterbrechen.

5. Wählen Sie Löschen aus.
6. Löschen Sie den Amazon Simple Storage Service (Amazon S3)-Bucket, in dem die Daten gespeichert wurden. Weitere Informationen zum Löschen eines S3-Buckets finden Sie unter Löschen eines Buckets.

Beheben von Compliance-Problemen mithilfe von EventBridge

Mit Run Command, eine Funktion von AWS Systems Manager, lassen sich Probleme bei der Patch- und Zuordnungs-Compliance schnell zu beheben. Sie können Instance- oder AWS IoT Greengrass-Core-Geräte-IDs oder Tags anvisieren und das `AWS-RunPatchBaseline`-Dokument oder das `AWS-RefreshAssociation`-Dokument ausführen. Wenn das Compliance-Problem nicht durch die Aktualisierung der Zuordnung oder eine erneute Ausführung der Patch-Baseline behoben wird, müssen Sie Ihre Zuordnungen, Patch-Baselines oder Instance-Konfigurationen untersuchen, um herauszufinden, warum die Run Command-Operationen das Problem nicht behoben haben.

Weitere Informationen zu Patch-Vorgängen finden Sie unter AWS Systems Manager Patch Manager und Informationen über das `AWS-RunPatchBaseline` SSM-Dokument.

Weitere Informationen zu Zuordnungen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

Weitere Informationen zum Ausführen eines Befehls finden Sie unter AWS Systems Manager Run Command.

Compliance als Ziel eines EventBridge-Ereignisses angeben

Sie können Amazon EventBridge auch so konfigurieren, dass als Reaktion auf Systems Manager Compliance-Ereignisse eine Aktion ausgeführt wird. Zum Beispiel, wenn kritische Patch-Updates auf mindestens einem verwalteten Knoten nicht installiert werden oder eine Zuordnung zur Installation einer Antiviren-Software nicht ausgeführt wird, können Sie EventBridge so konfigurieren, dass das `AWS-RunPatchBaseline`-Dokument oder das `AWS-RefreshAssocation`-Dokument ausgeführt wird, wenn das Compliance-Ereignis eintritt.

Führen Sie die folgenden Schritte aus, um Compliance als Ziel eines EventBridge-Ereignisses festzulegen.

Konfigurieren von Compliance als Ziel eines EventBridge-Ereignisses (Konsole)

1. Öffnen Sie die Amazon EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.
3. Wählen Sie Create rule (Regel erstellen).
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben AWS-Region und auf demselben Event Bus haben.
5. Wählen Sie als Event bus (Event Bus) den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf übereinstimmende Ereignisse reagiert, die von Ihrem eigenen AWS-Konto stammen, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es stets an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus.
7. Wählen Sie Next (Weiter).

8. Wählen Sie für Event source (Ereignisquelle) AWS events or EventBridge partner events (-Ereignisse oder EventBridge-Partnerereignisse).
9. Wählen Sie im Abschnitt Ereignismuster die Option Ereignismusterformular aus.
10. Als Event source (Ereignisquelle) wählen Sie AWS-Services aus.
11. Wählen Sie für AWS service (-Service), die Option Systems Manager aus.
12. Wählen Sie für Event Type (Ereignistyp) Configuration Compliance.
13. Für Specific detail type(s) (Spezifische(r) Detail-Typ(en)), wählen Sie Configuration Compliance State Change (Konfiguration-Compliance-Statusänderung).
14. Wählen Sie Next (Weiter).
15. Bei Target types (Zieltypen) wählen Sie AWS-Service aus.
16. Für Select a target (Ziel auswählen), wählen Sie Systems Manager Run Command.
17. Wählen Sie in der Liste Document (Dokument) ein Systems Manager-Dokument (SSM-Dokument) aus, das Sie ausführen möchten, wenn das Ziel aufgerufen wird. Wählen Sie beispielsweise `AWS-RunPatchBaseline` als ein nicht konformes Patch-Ereignis oder `AWS-RefreshAssociation` als ein nicht konformes Zuweisungsereignis aus.
18. Geben Sie Informationen für die verbleibenden Felder und Parameter an.

    > **Note**
    >
    > Erforderliche Felder und Parameter sind mit einem Sternchen (*) neben den Namen gekennzeichnet. Um ein Ziel zu erstellen, müssen Sie bei jedem erforderlichen Parameter oder Feld einen Wert angeben. Andernfalls erstellt das System die Regel, führt diese aber nicht aus.

19. Wählen Sie Next (Weiter).
20. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Tagging Your Amazon EventBridge Resources (Taggen Ihrer Amazon EventBridge Resources) im Amazon EventBridge-Benutzerhandbuch.
21. Wählen Sie Next (Weiter).
22. Überprüfen Sie die Details der Regel und wählen Sie dann Create rule (Regel erstellen) aus.

Compliance-Walkthrough (AWS CLI)

Das folgende Verfahren führt Sie durch die Schritte zur Verwendung der AWS Command Line Interface (AWS CLI) zum Aufrufen des AWS Systems Manager PutComplianceItems-API-Vorgangs, mit der Sie einer Ressource benutzerdefinierte Compliance-Metadaten zuweisen können. Sie können mit dieser API-Operation zudem einem verwalteten Knoten manuell Patch- oder Zuordnungs-Compliance-Metadaten zuweisen, wie im folgenden Walkthrough dargestellt. Weitere Informationen zur Verwendung benutzerdefinierter Compliance finden Sie unter Informationen zu benutzerdefinierter Compliance.

So weisen Sie einer verwalteten Instance benutzerdefinierte Compliance-Metadaten zu (AWS CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um einem verwalteten Knoten benutzerdefinierte Compliance-Metadaten zuweisen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Der `ResourceType`-Parameter unterstützt nur einen Wert von `ManagedInstance`. Geben Sie diesen Wert an, auch wenn Sie einem verwalteten AWS IoT Greengrass-Core-Gerät benutzerdefinierte Compliance-Metadaten zuweisen.

   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. Wiederholen Sie den vorherigen Schritt, um einem oder mehreren Knoten weitere benutzerdefinierte Compliance-Metadaten zuzuweisen. Mit folgenden Befehlen können Sie verwalteten Knoten die Patch- oder Zuordnungs-Compliance-Metadaten auch manuell zuweisen:

   Zuordnungs-Compliance-Metadaten

   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
   ```

   Patch Compliance-Metadaten

   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. Führen Sie den folgenden Befehl aus, um eine Liste der Compliance-Elemente für einen bestimmten verwalteten Knoten anzuzeigen. Mithilfe von Filtern können Sie Details zu bestimmten Compliance-Daten anzeigen.

   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
   ```

Die folgenden Beispiele zeigen, wie Sie diesen Befehl mit Filtern verwenden.

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. Führen Sie den folgenden Befehl aus, um eine Übersicht der Compliance-Statusarten anzuzeigen. Mithilfe von Filtern können Sie Details zu bestimmten Compliance-Daten anzeigen.

```
aws ssm list-resource-compliance-summaries --filters One or more filters.
```

Die folgenden Beispiele zeigen, wie Sie diesen Befehl mit Filtern verwenden.

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. Mit dem folgenden Befehl zeigen Sie eine Übersichtszahl der konformen und nicht konformen Ressourcen für einen Compliance-Typ an. Mithilfe von Filtern können Sie Details zu bestimmten Compliance-Daten anzeigen.

```
aws ssm list-compliance-summaries --filters One or more filters.
```

Die folgenden Beispiele zeigen, wie Sie diesen Befehl mit Filtern verwenden.

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-Bestand

AWS Systems Manager Inventory sorgt für Transparenz in Ihrer AWS-Computing-Umgebung. Mit Inventory können Sie Metadaten aus Ihren verwalteten Knoten erfassen. Sie können diese Metadaten in einem zentralen Amazon Simple Storage Service (Amazon S3)-Bucket speichern und dann die integrierten Tools nutzen, um Daten abzufragen und schnell zu ermitteln, welche Knoten die Software ausführen, welche Konfigurationen im Rahmen Ihrer Software-Richtlinie erforderlich sind und welche Knoten aktualisiert werden müssen. Sie können Inventory mit nur einem Klick für all Ihre verwalteten

Knoten konfigurieren. Sie können auch Bestandsdaten von mehreren AWS-Regionen und AWS-Konten konfigurieren und anzeigen. Um mit Inventory zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Inventory.

Wenn die vorkonfigurierten, von Systems Manager Inventory Inventory erfassten Metadaten nicht Ihren Anforderungen entsprechen, können Sie einen benutzerdefinierten Bestand erstellen. Beim benutzerdefinierten Bestand handelt es sich lediglich um eine JSON-Datei mit Informationen, die Sie bereitstellen und zum verwalteten Knoten in einem bestimmten Verzeichnis hinzufügen. Bei der Datenerfassung erfasst Systems Manager Inventory diese Daten für den benutzerdefinierten Bestand. Beispiel: Wenn Sie ein großes Rechenzentrum betreiben, können Sie die Rack-Standorte der einzelnen Server als benutzerdefinierten Bestand angeben. Anschließend können Sie beim Anzeigen anderer Bestandsdaten die Daten im Rack-Bereich anzeigen.

> **Important**
>
> Systems Manager Inventory erfasst nur Metadaten von Ihren verwalteten Knoten. Inventory greift nicht auf proprietäre Informationen oder Daten zu.

In der folgenden Tabelle werden die Arten von Daten beschrieben, die Sie mit Systems Manager Inventory erfassen können. Außerdem werden darin verschiedene Angebote für die gezielte Erfassung von Knoten sowie die Erfassungsintervalle beschrieben, die angegeben werden können.

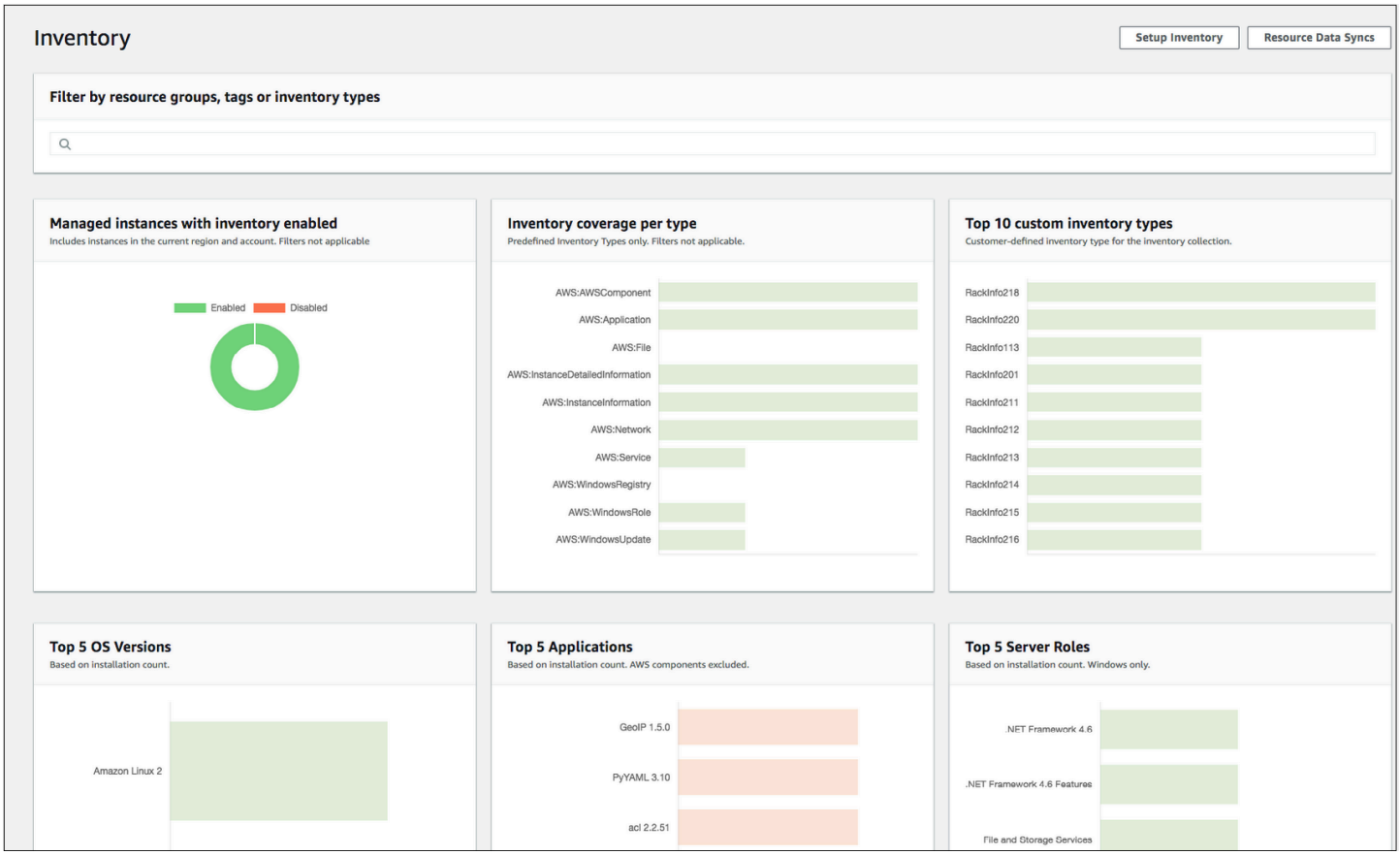
Konfiguration	Details
Metadatentypen	Sie können Inventory so konfigurieren, dass die folgenden Typen von Daten erfasst werden: • Anwendungen: Anwendungsnamen, Herausgeber, Versionen usw. • AWS-Komponenten: EC2-Treiber, Agenten, Versionen usw. • Dateien: Name, Größe, Version, Installat ionsdatum, Änderung und Zeitpunkt der letzten Zugriffe usw. • Netzwerkkonfiguration: IP-Adresse, MAC-Adresse, DNS-Gateway, Subnetzmaske usw.

Konfiguration	Details
	• Windows-Updates: Hotfix-ID, installiert durch, Installationsdatum usw. • Instance-Details: Systemname, Name des Betriebssystems (OS), Version des Betriebssystems, DNS, Domain, Arbeitsgruppe, Betriebssystemarchitektur usw. • Services: Name, Anzeigename, Status, abhängige Services, Servicetyp, Starttyp usw. • Tags: Tags, die Ihren Knoten zugewiesen werden. • Windows-Registry: Registry-Schlüsselpfad, Wertname, Werttyp und Wert. • Windows-Rollen: Name, Anzeigename, Pfad, Funktionstyp, Installationsstatus usw. • Custom inventory (Benutzerdefinierter Bestand): Metadaten, die einem verwalteten Knoten zugewiesen wurden, wie in Arbeiten mit benutzerdefiniertem Bestand beschrieben. **Note** Wie Sie eine Liste aller durch Inventory erfassten Metadaten anzeigen, lesen Sie nach unter Metadaten-Erfassung nach Bestand

Konfiguration	Details
Zu erfassende Knoten	Sie können wählen, ob Sie den Bestand für alle verwalteten Knoten in Ihrem AWS-Konto erfassen wollen oder einzelne Knoten oder Zielgruppen von Knoten auswählen möchten, indem Sie Tags verwenden. Weitere Informationen zur Erfassung von Bestandsdaten aller verwalteten Knoten finden Sie unter Inventarisieren Sie alle verwalteten Knoten in Ihrem AWS-Konto.
Wann Daten erfasst werden sollen	Sie können ein Intervall für die Erfassung in Minuten, Stunden und Tagen angeben. Das kürzeste mögliche Erfassungsintervall ist alle 30 Minuten.

> **Note**
>
> Abhängig von der Menge der erfassten Daten kann es einige Minuten in Anspruch nehmen, bis die Daten vom System in der Ausgabe bereitgestellt werden können, die Sie angegeben haben. Nachdem die Informationen erfasst wurden, werden die Daten über einen sicheren HTTPS-Kanal an einen AWS-Speicher für einfachen Text gesendet, auf den nur über Ihr AWS-Konto zugegriffen werden kann.

Sie können die Daten in der Systems Manager-Konsole auf der Seite Inventory anzeigen. Diese enthält mehrere vordefinierte Karten zur Datenabfrage.



> **Note**
>
> Inventory-Karten filtern automatisch verwaltete Amazon EC2-Instances mit dem Status Beendet und Angehalten heraus. Bei On-Premises- und von AWS IoT Greengrass-Core-Geräten verwalteten Knoten filtern Inventory-Karten automatisch Knoten im Zustand Terminated (Beendet) heraus.

Wenn Sie eine Resource Data Sync zum Synchronisieren und Speichern aller Daten in einem einzelnen Amazon S3-Bucket erstellen, können Sie die Daten auf der Seite Inventory Detailed View (Detailansicht zum Bestand) detailliert anzeigen. Weitere Informationen finden Sie unter Abfragen von Bestandsdaten aus mehreren Regionen und Konten.

EventBridge-Support

Diese Systems Manager Funktion wird als Event (Ereignis)-Typ in Amazon EventBridge Regeln unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen

mit Amazon EventBridge und Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager.

Inhalt

- Weitere Informationen über Systems Manager Inventory
- Einrichten von Systems Manager Inventory
- Konfigurieren der Bestandserfassung
- Arbeiten mit Systems Manager-Bestandsdaten
- Arbeiten mit benutzerdefiniertem Bestand
- Anzeigen von Bestandsverlauf und Änderungsnachverfolgung
- Anhalten der Datenerfassung und Löschen von Bestandsdaten
- Walkthroughs zu Systems Manager Inventory
- Fehlerbehebung bei Problemen mit Systems Manager Inventory

Weitere Informationen über Systems Manager Inventory

Wenn Sie AWS Systems Manager Inventory konfigurieren, geben Sie den Typ der zu erfassenden Metadaten, die verwalteten Knoten, aus denen die Metadaten erfasst werden sollen, und einen Zeitplan für die Erfassung der Metadaten an. Diese Konfigurationen werden als AWS Systems Manager State Manager-Zuordnung in Ihrem AWS-Konto gespeichert. Eine Zuordnung ist einfach eine Konfiguration.

> **Note**
>
> Inventory erfasst nur Metadaten. Es werden keine personenbezogenen oder vertraulichen Daten erfasst.

Themen

- Metadaten-Erfassung nach Bestand
- Arbeiten mit Datei- und Windows-Registrierungsbestand
- Verwandte AWS-Services

Metadaten-Erfassung nach Bestand

Das folgende Beispiel zeigt die vollständige Liste der Metadaten von jedem AWS Systems Manager Inventar-Plugin erfasst wurden.

```
{
    "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:Process",
    "version": "1.0",
    "attributes":[
      { "name": "StartTime",          "dataType": "STRING"},
      { "name": "CommandLine",        "dataType": "STRING"},
      { "name": "User",               "dataType": "STRING"},
      { "name": "FileName",           "dataType": "STRING"},
      { "name": "FileVersion",        "dataType": "STRING"},
      { "name": "FileDescription",    "dataType": "STRING"},
      { "name": "FileSize",           "dataType": "STRING"},
      { "name": "CompanyName",        "dataType": "STRING"},
      { "name": "ProductName",        "dataType": "STRING"},
      { "name": "ProductVersion",     "dataType": "STRING"},
      { "name": "InstalledDate",      "dataType": "STRING"},
      { "name": "InstalledDir",       "dataType": "STRING"},
      { "name": "UsageId",            "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

- Für`"typeName": "AWS:InstanceInformation"` kann der `InstanceStatus` einer der folgenden sein: Active, ConnectionLost, Stopped, Terminated (Aktiv, Verbindung abgebrochen, angehalten, beendet).
- Mit der Veröffentlichung von Version 2.5 ersetzt der RPM-Paketmanager ersetzt das Serial-Attribut durch Epoch. Das Epoch-Attribut ist eine monoton zunehmende Ganzzahl wie Serial. Wenn Sie eine Bestandsaufnahme unter Verwendung des `AWS:Application-`

Typs durchführen, bedeutet ein größerer Wert für Epoch eine neuere Version. Wenn Epoch-Werte gleich oder leer sind, verwenden Sie den Wert des Version- oder Release-Attributs, um die neuere Version zu bestimmen.

- Einige Metadaten sind in Linux-Instances nicht verfügbar. Insbesondere für „typeName“: „AWS:Network“ werden die folgenden Metadatentypen für Linux-Instances noch nicht unterstützt. Sie WERDEN für Windows unterstützt.
  - { "name": "SubnetMask", "dataType": "STRING"},
  - { "name": "DHCPServer", "dataType": "STRING"},
  - { "name": "DNSServer", "dataType": "STRING"},
  - { "name": "Gateway", "dataType": "STRING"},

Arbeiten mit Datei- und Windows-Registrierungsbestand

AWS Systems Manager Inventory gestattet Ihnen, Bestandsdateien auf Windows-, Linux- und macOS-Betriebssystemen zu durchsuchen und zu inventarisieren. Sie können auch die Windows-Registry durchsuchen und inventarisieren.

Dateien: Sie können Metadaten-Informationen zu Dateien erfassen, einschließlich Dateinamen, der Erstellungszeit der Dateien, der letzten Änderungs- und Zugriffszeit der Dateien oder Dateigrößen, um nur ein paar zu nennen. Um mit der Erfassung eines Dateibestands zu beginnen, geben Sie einen Dateipfad an, in dem Sie die Inventarisierung durchführen möchten, ein oder mehrere Muster, die definieren, welche Dateitypen inventarisiert werden soll, und ob der Pfad rekursiv durchsucht werden soll. Systems Manager inventarisiert alle Datei-Metadaten für Dateien im angegebenen Pfad, die dem Muster entsprechen. Die Dateiinventarisierung verwendet die folgenden Eingangsparameter.

```
{
"Path": string,
"Pattern": array[string],
"Recursive": true,
"DirScanLimit" : number // Optional
}
```

- Path: Der Verzeichnispfad, in dem Dateien inventarisiert werden sollen. Für Windows können Sie Umgebungsvariablen wie %PROGRAMFILES% verwenden, solange der Variable auf einen einzigen Verzeichnispfad abgebildet wird. Wenn Sie beispielsweise %PATH% verwenden, das auf mehreren Verzeichnispfade abgebildet wird, wirft Inventory einen Fehler auf.

- Pattern: Ein Array mit zu identifizierenden Mustern.
- Recursive: Ein Boolescher Wert, der angibt, ob Inventory die Verzeichnisse rekursiv durchlaufen soll.
- DirScanLimit: Ein optionaler Wert, der festlegt, wie viele Verzeichnisse gescannt werden sollen. Verwenden Sie diesen Parameter, um die Leistung Ihrer verwalteten Knoten möglichst wenig zu beeinträchtigen. Standardmäßig scannt Inventory maximal 5.000 Verzeichnisse.

> **Note**
>
> Inventory erfasst Metadaten für maximal 500 Dateien für alle angegebenen Pfade.

Hier finden Sie einige Beispiele, wie Sie die Parameter angeben, wenn Sie eine Inventarisierung von Dateien vornehmen wollen.

- Unter Linux und macOS erfassen Sie Metadaten von .sh-Dateien im Verzeichnis `/home/ec2-user` ohne die Unterverzeichnisse.

```
[{"Path":"/home/ec2-user","Pattern":["*.sh", "*.sh"],"Recursive":false}]
```

- Unter Windows erfassen Sie rekursiv Metadaten aller ".exe" -Dateien im Ordner Programme, einschließlich der Unterverzeichnisse.

```
[{"Path":"C:\Program Files","Pattern":["*.exe"],"Recursive":true}]
```

- Unter Windows erfassen Sie Metadaten bestimmter Log-Muster.

```
[{"Path":"C:\ProgramData\Amazon","Pattern":["*amazon*.log"],"Recursive":true}]
```

- Beschränken Sie die Anzahl der Verzeichnisse, wenn Sie eine rekursive Erfassung durchführen.

```
[{"Path":"C:\Users","Pattern":["*.ps1"],"Recursive":true, "DirScanLimit": 1000}]
```

Windows Registry: Sie können Schlüssel und Werte der Windows Registry erfassen. Sie können einen Schlüssel-Pfad auswählen und alle Schlüssel und Werte rekursiv erfassen. Sie können auch einen bestimmten Registrierungsschlüssel und seinen Wert für einen bestimmten Pfad erfassen. Inventory erfasst den Schlüsselpfad, den Namen, Typ und Wert.

```
{
"Path": string,
"Recursive": true,
"ValueNames": array[string] // optional
}
```

- Path: Der Pfad zum Registry-Schlüssel.
- Recursive: Ein Boolescher Wert, der angibt, ob Inventory die Registry-Pfade rekursiv durchlaufen soll.
- ValueNames: Ein Array von Wertnamen für die Inventarisierung von Registrierungsschlüsseln. Wenn Sie diesen Parameter verwenden, inventarisiert Systems Manager nur die angegebenen Wertenamen für den angegebenen Pfad.

> **Note**
>
> Inventory erfasst Metadaten für maximal 250 Registry-Schlüsselwerte für alle angegebenen Pfade.

Hier finden Sie einige Beispiele, wie Sie die Parameter angeben, wenn Sie eine Inventarisierung der Windows Registry vornehmen wollen.

- Erfassen Sie alle Schlüssel und Werte rekursiv für einen bestimmten Pfad.

```
[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon","Recursive": true}]
```

- Erfassen Sie alle Schlüssel und Werte für einen bestimmten Pfad (die rekursive Suche ist deaktiviert).

```
[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Intel\PSIS\PSIS_DECODER", "Recursive": false}]
```

- Erfasst einen bestimmten Schlüssel unter Verwendung der Option `ValueNames`.

```
{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon\MachineImage","ValueNames":["AMIName"]}
```

Verwandte AWS-Services

AWS Systems Manager Inventory stellt einen Snapshot Ihres aktuellen Bestands bereit; damit können Sie Software-Richtlinien verwalten und das Sicherheitsniveau Ihrer gesamten Flotte verbessern. Sie können Ihre Lagerverwaltung und Migrationsfunktionen mithilfe der folgenden AWS-Services erweitern:

- AWS Config bietet eine historische Aufzeichnung der Änderungen an Ihren Bestand sowie die Möglichkeit, Regeln zu erstellen, nach denen entsprechende Benachrichtigungen generiert werden, wenn ein Element der Konfiguration geändert wird. Weitere Informationen finden Sie unter Bestandserfassung von in Amazon EC2 verwalteten Instances im AWS Config-Entwicklerleitfaden.
- AWS Application Discovery Service wurde entwickelt, um den Bestand in Bezug auf den Typ des Betriebssystems, installierte Anwendungen, Prozesse, Verbindungen und Metriken zur Serverleistung in Ihren On-Premises-VMs zu erfassen und so die Migration zu AWS zu unterstützen. Weitere Informationen finden Sie im Application Discovery Service-Benutzerhandbuch.

Einrichten von Systems Manager Inventory

Bevor Sie AWS Systems Manager Inventory zum Sammeln von Metadaten über die Anwendungen, Services, AWS-Komponenten und mehr verwenden, die auf Ihren verwalteten Knoten ausgeführt werden, empfehlen wir, die Ressourcen-Datensynchronisierung zu konfigurieren, um die Speicherung Ihrer Bestandsdaten in einem einzigen Amazon Simple Storage Service (Amazon S3)-Bucket zu zentralisieren. Wir empfehlen außerdem, die Amazon EventBridge-Überwachung von Bestandsereignissen zu konfigurieren. Diese Prozesse erleichtern das Anzeigen und Verwalten von Bestandsdaten und -sammlungen.

Themen

- Konfigurieren von Resource Data Sync für Inventory
- Informationen zur EventBridge-Überwachung von Inventory-Ereignissen

Konfigurieren von Resource Data Sync für Inventory

In diesem Thema wird beschrieben, wie Sie die Ressourcendatensynchronisierung für AWS Systems Manager -Inventar einrichten und konfigurieren. Informationen zu Resource Data Sync für Systems Manager Explorer finden Sie unter Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen.

Über Resource Data Sync

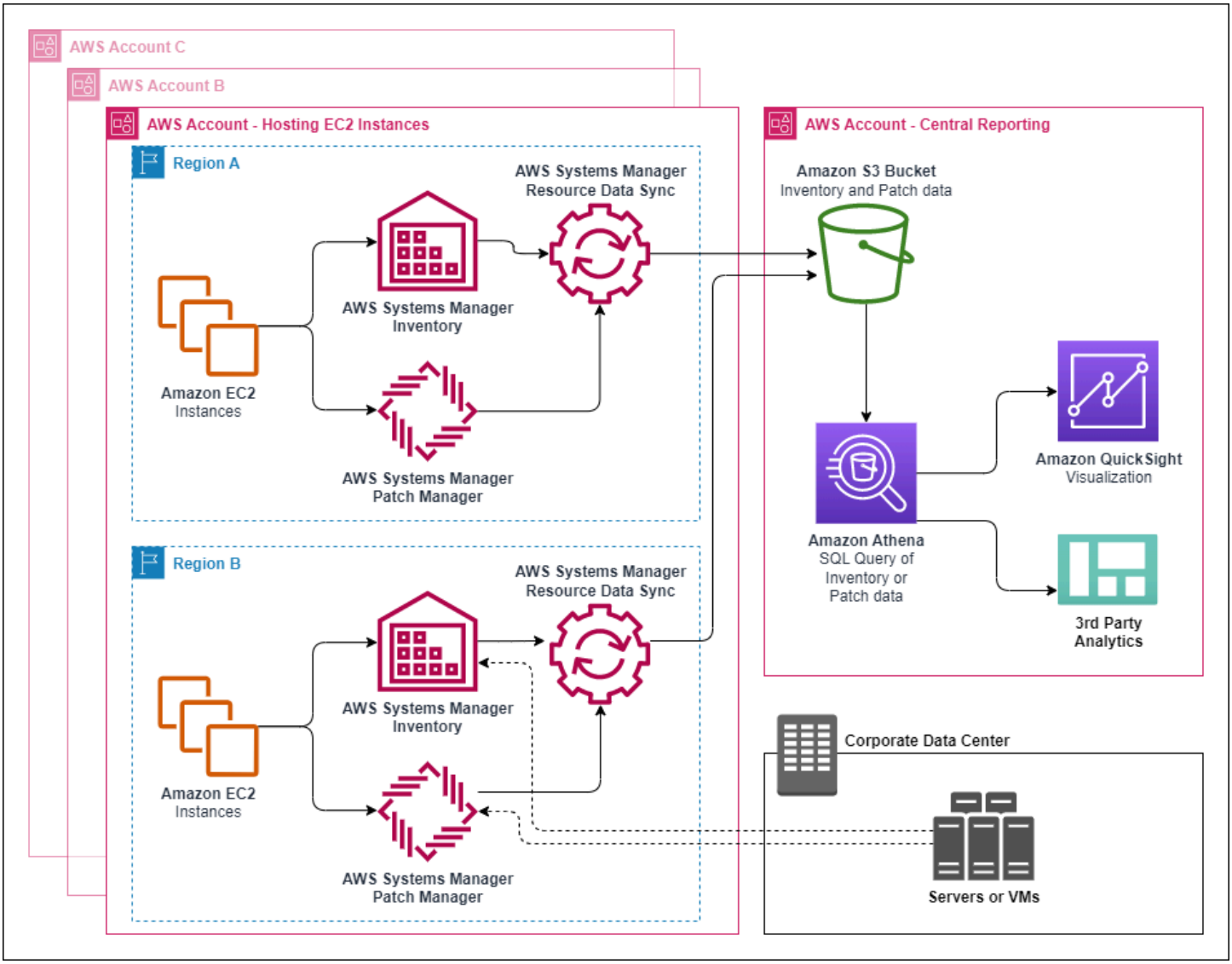
Sie können mit der Ressourcen-Datensynchronisierung von Systems Manager Bestandsdaten aus allen Ihren verwalteten Knoten an einen einzelnen Amazon Simple Storage Service (Amazon S3)-Bucket senden. Resource Data Sync aktualisiert die Daten dann automatisch, wenn neue Bestandsdaten erfasst werden. Da alle Inventardaten in einem Amazon S3 S3-Ziel-Bucket gespeichert sind, können Sie Dienste wie Amazon Athena und Amazon verwenden, QuickSight um die aggregierten Daten abzufragen und zu analysieren.

Sie können Inventory z. B. so konfigurieren, dass Daten über das Betriebssystem (OS) und die Anwendungen erfasst werden, die auf eine Flotte von 150 verwalteten Knoten ausgeführt werden. Einige dieser Knoten befinden sich in einem On-Premises-Rechenzentrum und andere werden in Amazon Elastic Compute Cloud (Amazon EC2) über mehrere AWS-Regionen hinweg ausgeführt. Wenn Sie die Ressourcen-Datensynchronisierung nicht konfiguriert haben, müssen Sie die erfassten Bestandsdaten für jeden verwalteten Knoten manuell einholen oder Sie müssen entsprechende Skripts erstellen, die diese Informationen sammeln. Anschließend müssten Sie die Daten in eine Anwendung portieren, um Abfragen ausführen und diese analysieren zu können.

Mit der Ressourcen-Datensynchronisierung führen Sie eine einmalige Operation aus, die alle Bestandsdaten von allen Ihren verwalteten Knoten synchronisiert. Wenn die Synchronisierung erfolgreich erstellt wurde, erstellt Systems Manager eine Ausgangsbasis mit allen Bestandsdaten und speichert diese in dem jeweiligen Amazon S3-Ziel-Bucket. Wenn neue Bestandsdaten erfasst werden, aktualisiert Systems Manager die Daten in dem Amazon S3-Bucket automatisch. Anschließend können Sie die Daten schnell und kostengünstig zu Amazon Athena und Amazon portieren. QuickSight

Diagramm 1 zeigt, wie Bestandsdaten von Amazon EC2 und anderen Maschinentypen in einer Hybrid- und Multi-Cloud-Umgebung in einem Ziel-Amazon-S3-Bucket zusammengeführt werden. Dieses Diagramm zeigt auch, wie die Ressourcendatensynchronisierung mit mehreren AWS-Konten und funktioniert. AWS-Regionen

Diagramm 1: Synchronisieren von Ressourcendaten mit mehreren AWS-Konten und AWS-Regionen



Wenn Sie einen verwalteten Knoten löschen, behält die Ressourcen-Datensynchronisierung die Bestandsdatei für den gelöschten Knoten bei. Für ausgeführte Knoten überschreibt die Ressourcen-Datensynchronisierung die alte Bestandsdatei jedoch automatisch, wenn neue Dateien erstellt und in den Amazon-S3-Bucket geschrieben werden. Wenn Sie Inventaränderungen im Laufe der Zeit verfolgen möchten, können Sie den AWS Config Dienst verwenden, um den `SSM:ManagedInstanceInventory` Ressourcentyp zu verfolgen. Weitere Informationen finden Sie unter Erste Schritte mit AWS Config.

Gehen Sie wie in diesem Abschnitt beschrieben vor, um mithilfe von Amazon S3 und AWS Systems Manager Konsolen eine Ressourcendatensynchronisierung für Inventory zu erstellen. Sie können sie auch verwenden AWS CloudFormation , um eine Ressourcendatensynchronisierung zu erstellen oder zu löschen. Um sie zu verwenden AWS CloudFormation, fügen Sie die

`AWS::SSM::ResourceDataSync`Ressource zu Ihrer AWS CloudFormation Vorlage hinzu. Informationen dazu finden Sie in einer der folgenden Dokumentationsressourcen:

- AWS CloudFormation Ressource für die Synchronisation von Ressourcendaten in AWS Systems Manager (Blog)
- Arbeiten mit AWS CloudFormation -Vorlagen im AWS CloudFormation -Benutzerhandbuch

> **Note**
>
> Sie können AWS Key Management Service (AWS KMS) verwenden, um Inventardaten im Amazon S3 S3-Bucket zu verschlüsseln. Ein Beispiel dafür, wie Sie mithilfe von AWS Command Line Interface (AWS CLI) eine verschlüsselte Synchronisation erstellen und wie Sie mit den zentralisierten Daten in Amazon Athena und Amazon arbeiten QuickSight, finden Sie unterWalkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten.

Bevor Sie beginnen

Bevor Sie eine Ressourcendatensynchronisierung erstellen, verwenden Sie das folgende Verfahren, um einen zentralen Amazon S3-Bucket zum Speichern aggregierter Bestandsdaten zu erstellen. Das Verfahren beschreibt, wie Sie eine Bucket-Richtlinie zuweisen, die es Systems Manager ermöglicht, Bestandsdaten aus mehreren Konten in den Bucket zu schreiben. Wenn Sie bereits über einen Amazon S3-Bucket verfügen, den Sie zum Aggregieren von Bestandsdaten für die Ressourcendatensynchronisierung verwenden möchten, müssen Sie den Bucket so konfigurieren, dass die Richtlinie im folgenden Verfahren verwendet wird.

> **Note**
>
> Systems Manager Inventory kann keine Daten zu einem angegebenen Amazon S3-Bucket hinzufügen, wenn dieser Bucket für die Verwendung von Object Lock konfiguriert ist. Stellen Sie sicher, dass der Amazon S3-Bucket, den Sie für die Resource Data Sync erstellen oder auswählen, nicht für die Verwendung von Amazon S3 Object Lock konfiguriert ist. Weitere Informationen finden Sie unter Funktionsweise von Amazon S3 Object Lock im Benutzerhandbuch zu Amazon Simple Storage Service.

So erstellen und konfigurieren Sie einen Amazon S3-Bucket für Resource Data Sync

1. Öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie einen Bucket zum Speichern der aggregierten Bestandsdaten. Weitere Informationen finden Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service. Notieren Sie sich den Namen des Buckets und den AWS-Region Ort, an dem Sie ihn erstellt haben.
3. Klicken Sie auf die Registerkarte Permissions (Berechtigungen) und anschließend auf Bucket Policy (Bucket-Richtlinie).
4. Kopieren Sie die folgende Bucket-Richtlinie in den Richtlinien-Editor. Ersetzen Sie DOC-EXAMPLE-BUCKET und *Account-ID* durch den Namen des von Ihnen erstellten S3-Buckets und eine gültige ID. AWS-Konto

   Damit mehrere Personen AWS-Konten Inventardaten an den zentralen Amazon S3 S3-Bucket senden können, geben Sie jedes Konto in der Richtlinie an, wie im folgenden `Resource` Beispiel gezeigt:

```
"Resource": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/accountid=123456789012/*",
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/accountid=444455556666/*",
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/accountid=777788889999/*"
],
"Condition": {
    "StringEquals": {
         "s3:x-amz-acl": "bucket-owner-full-control",
         "aws:SourceAccount": [
                 "123456789012",
                 "444455556666",
                 "777788889999"
         ]
     },
      "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:ssm:*:123456789012:resource-data-sync/*",
            "arn:aws:ssm:*:444455556666:resource-data-sync/*",
            "arn:aws:ssm:*:777788889999:resource-data-sync/*"
          ]
     }
}
```

> **Note**
>
> Informationen zum Anzeigen Ihrer AWS-Konto ID finden Sie unter Ihre Amazon Web Services Services-Konto-ID und deren Alias im IAM-Benutzerhandbuch.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Sid": "SSMBucketPermissionsCheck",
        "Effect": "Allow",
        "Principal": {
            "Service": "ssm.amazonaws.com"
        },
        "Action": "s3:GetBucketAcl",
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        },
        {
        "Sid": " SSMBucketDelivery",
        "Effect": "Allow",
        "Principal": {
            "Service": "ssm.amazonaws.com"
        },
        "Action": "s3:PutObject",
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/accountid=ID_number/*",
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/accountid=ID_number/*",
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/accountid=ID_number/*",
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/accountid=ID_number/*"
        ],
        "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": "ID_number"
            },
            "ArnLike": {
                "aws:SourceArn": "arn:aws:ssm:*:ID_number:resource-data-sync/*"
            }
        }
      }
```

```
    ]
}
```

Erstellen einer Resource Data Sync für Inventory

Führen Sie die folgenden Schritte aus, um mit der Systems Manager-Konsole eine Ressource Data Sync for Systems Manager Inventory zu erstellen. Informationen zum Erstellen einer Ressourcendatensynchronisierung mithilfe von finden Sie AWS CLI unterWalkthrough: Konfigurieren Ihrer verwalteten Knoten für Inventory mithilfe der CLI.

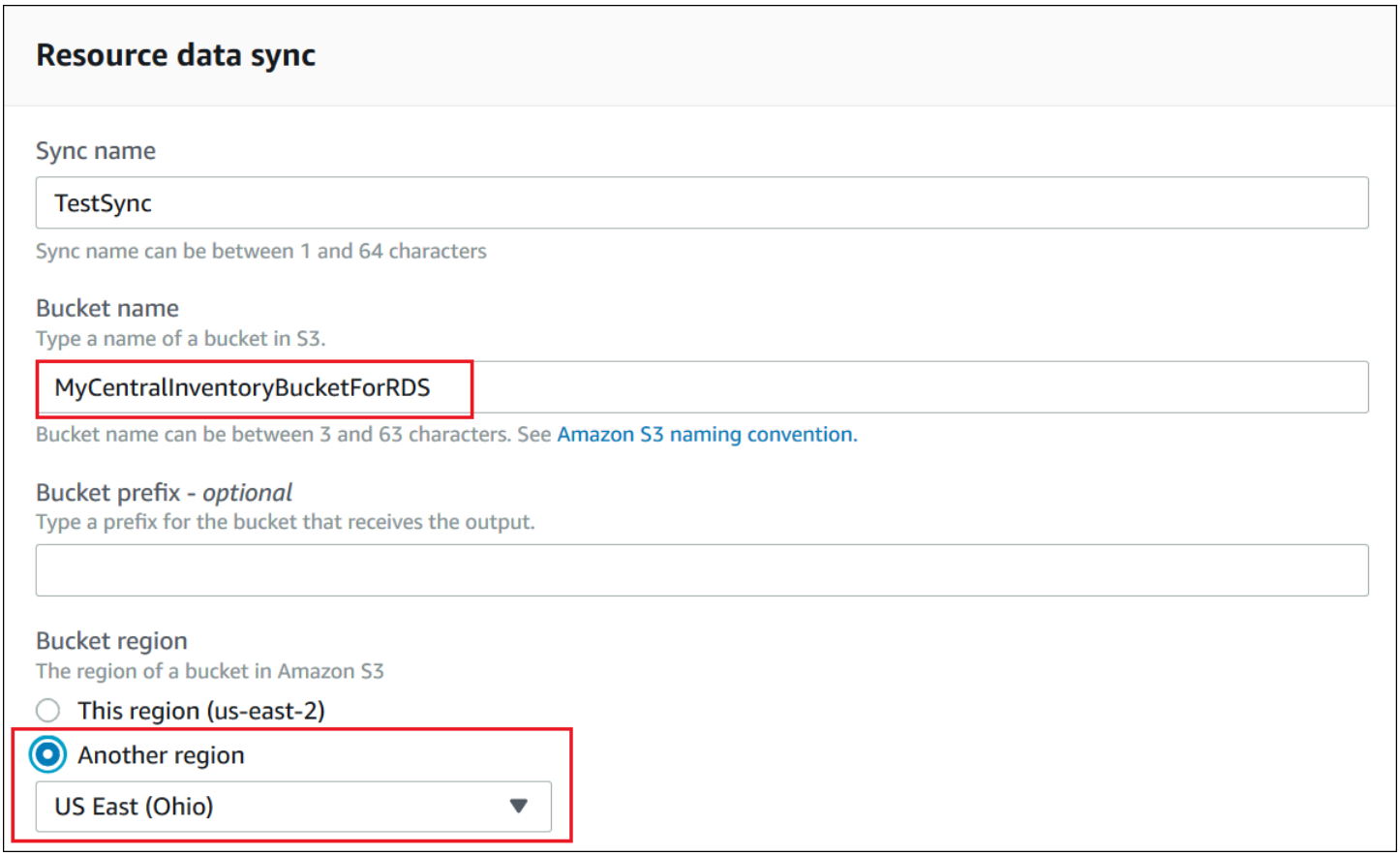
Erstellen einer Resource Data Sync

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Menü Account management (Kontoverwaltung) die Option Resource Data Sync (Ressourcendaten).
4. Wählen Sie Create resource data sync (Ressourcen-Datensynchronisierung erstellen).
5. Geben Sie im Feld Sync name einen Namen für die Synchronisierungskonfiguration ein.
6. Geben Sie in das Feld Bucket name (Bucket-name) den Namen des Amazon S3-Buckets ein, das Sie mit dem Verfahren To create and configure an Amazon S3 bucket for resource data sync (Erstellen und Konfigurieren eines Amazon S3-Buckets für Ressource Data Sync) erstellt haben.
7. (Optional) Geben Sie im Feld Bucket prefix (Bucket-Präfix) den Namen eines Amazon S3-Bucket-Präfixes (Unterverzeichnis) an.
8. Wählen Sie im Feld Bucket region (Bucket-Region) die Option This region (Diese Region) aus, wenn sich der erstellte Amazon S3-Bucket in der aktuellen AWS-Region befindet. Befindet sich der erstellte Bucket in einer anderen AWS-Region, wählen Sie Another region (Andere Region) aus und geben den Namen der Region an.

> **Note**
>
> Wenn sich der Synchronisierungs- und der Amazon S3-Ziel-Bucket in verschiedenen Regionen befinden, können Kosten für Datenübertragung anfallen. Weitere Informationen finden Sie unter Amazon S3 – Preise.

9. (Optional) Geben Sie im Feld KMS Key ARN (KMS-Schlüssel-ARN) einen KMS-Schlüssel-ARN zum Verschlüsseln von Bestandsdaten in Amazon S3 ein oder fügen Sie einen ein.
10. Wählen Sie Erstellen.

Um Inventardaten aus mehreren zu synchronisieren AWS-Regionen, müssen Sie in jeder Region eine Ressourcendatensynchronisierung einrichten. Wiederholen Sie diesen Vorgang an jedem AWS-Region Ort, an dem Sie Inventardaten sammeln und an den zentralen Amazon S3 S3-Bucket senden möchten. Wenn Sie die Synchronisierung in jeder Region erstellen, geben Sie den zentralen Amazon S3-Bucket im Bucket name (Bucket-Name) an. Verwenden Sie dann die Option Bucket region (Bucket-Region), um die Region auszuwählen, in der Sie den zentralen Amazon S3-Bucket erstellt haben, wie im folgenden Screenshot gezeigt. Wenn die Zuordnung zu Erfassen von Bestandsdaten das nächste Mal ausgeführt wird, speichert Systems Manager die Daten im zentralen Amazon S3-Bucket.



Erstellen einer Inventory Resource Data Sync für mehrere Konten, die in AWS Organizations definiert sind

Sie können Inventardaten von AWS-Konten Defined in AWS Organizations mit einem zentralen Amazon S3 S3-Bucket synchronisieren. Nachdem Sie das folgende Verfahren abgeschlossen haben, werden Bestandsdaten mit einzelnen Amazon S3-Schlüsselpräfixen im zentralen Bucket synchronisiert. Jedes key prefix steht für eine andere AWS-Konto ID.

Bevor Sie beginnen

Bevor Sie beginnen, stellen Sie sicher, dass Sie AWS-Konten in eingerichtet und konfiguriert haben AWS Organizations. Weitere Informationen finden Sie unter im AWS Organizations - Benutzerhandbuch.

Beachten Sie außerdem, dass Sie die organisationsbasierte Ressourcendatensynchronisierung für jede Ressource erstellen müssen AWS-Region und dass unter AWS-Konto definiert ist. AWS Organizations

Erstellen eines zentralen Amazon S3-Buckets

Verwenden Sie das folgende Verfahren, um einen zentralen Amazon S3-Bucket zum Speichern aggregierter Bestandsdaten zu erstellen. Das Verfahren beschreibt, wie Sie eine Bucket-Richtlinie zuweisen, die es Systems Manager ermöglicht, Bestandsdaten aus Ihrer AWS Organizations -Konto-ID in den Bucket zu schreiben. Wenn Sie bereits über einen Amazon S3-Bucket verfügen, den Sie zum Aggregieren von Bestandsdaten für die Ressourcendatensynchronisierung verwenden möchten, müssen Sie den Bucket so konfigurieren, dass die Richtlinie im folgenden Verfahren verwendet wird.

Um einen Amazon S3 S3-Bucket für die Ressourcendatensynchronisierung für mehrere Konten zu erstellen und zu konfigurieren, definiert in AWS Organizations

1. Öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie einen Bucket zum Speichern der aggregierten Bestandsdaten. Weitere Informationen finden Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service. Notieren Sie sich den Bucket-Namen und den AWS-Region Ort, an dem Sie ihn erstellt haben.
3. Klicken Sie auf die Registerkarte Permissions (Berechtigungen) und anschließend auf Bucket Policy (Bucket-Richtlinie).

4. Kopieren Sie die folgende Bucket-Richtlinie in den Richtlinien-Editor. Ersetzen Sie DOC-EXAMPLE-BUCKET und *organization-id* durch den Namen des Amazon S3 S3-Buckets, den Sie erstellt haben, und eine gültige Konto-ID. AWS Organizations

   Sie können auch das *bucket-prefix* durch den Namen eines Amazon S3-Präfixes (Unterverzeichnis) ersetzen. Wenn Sie kein Präfix erstellt haben, entfernen Sie *bucket-prefix/* aus dem ARN in der folgenden Richtlinie.

```
{
  "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:::DOC-EXAMPLE-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:::DOC-EXAMPLE-BUCKET/bucket-prefix/*/accountid=*/*"
      ]
    }
  ]
}
```

Erstellen einer Bestands-Resource Data Sync für Konten, die in AWS Organizations definiert sind

Das folgende Verfahren beschreibt, wie Sie mithilfe von eine Ressourcendatensynchronisierung AWS CLI für Konten erstellen, die in definiert sind. AWS Organizations Sie müssen den verwenden AWS CLI , um diese Aufgabe auszuführen. Sie müssen dieses Verfahren auch für jedes AWS-Region Verfahren ausführen, das AWS-Konto unter definiert ist AWS Organizations.

So erstellen Sie eine Ressourcendatensynchronisierung für ein Konto, das in AWS Organizations (AWS CLI) definiert ist

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob keine anderen Resource Data Syncs vorhanden sind. Sie können nur eine organisationsbasierte Resource Data Sync durchführen.

   ```
   aws ssm list-resource-data-sync
   ```

   Wenn der Befehl eine andere Resource Data Sync zurückgibt, müssen Sie sie löschen oder keine neue erstellen.
3. Führen Sie den folgenden Befehl aus, um eine Ressourcendatensynchronisierung für ein Konto zu erstellen, das in AWS Organizations definiert ist. Geben Sie für DOC-EXAMPLE-BUCKET den Namen des Amazon S3-Buckets an, den Sie zuvor in diesem Thema erstellt haben. Wenn Sie ein Präfix (Unterverzeichnis) für Ihren Bucket erstellt haben, geben Sie diese Informationen für *prefix-name* an.

```
aws ssm create-resource-data-sync --sync-name name --s3-
destination "BucketName=DOC-EXAMPLE-BUCKET,Prefix=prefix-
name,SyncFormat=JsonSerDe,Region=AWS-Region, for example us-
east-2,DestinationDataSharing={DestinationDataSharingType=Organization}"
```

4. Wiederholen Sie die Schritte 2 und 3 für alle AWS-Region Bereiche AWS-Konto , in denen Sie Daten mit dem zentralen Amazon S3 S3-Bucket synchronisieren möchten.

Ressourcendatensynchronisierungen verwalten

In jedem AWS-Konto Fall können 5 Ressourcendatensynchronisierungen durchgeführt werden. AWS-Region Sie können die AWS Systems Manager Fleet Manager Konsole verwenden, um Ihre Ressourcendatensynchronisierungen zu verwalten.

Um Ressourcendatensynchronisierungen anzuzeigen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Menü-Dropdown Kontoverwaltung die Option Ressourcendatensynchronisierung.
4. Wählen Sie eine Ressourcendatensynchronisierung aus der Tabelle aus, und klicken Sie dann auf Details anzeigen, um Informationen zu Ihrer Ressourcendatensynchronisierung anzuzeigen.

Löschen einer Resource Data Sync

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Menü-Dropdown Kontoverwaltung die Option Ressourcendatensynchronisierung.
4. Wählen Sie eine Ressourcendatensynchronisierung aus der Tabelle aus, und klicken Sie dann auf Löschen.

Informationen zur EventBridge-Überwachung von Inventory-Ereignissen

Sie können eine Regel in Amazon EventBridge konfigurieren, um ein Ereignis als Reaktion auf AWS Systems Manager-Inventory-Ressourcenstatusänderungen zu erstellen. EventBridge unterstützt

Ereignisse für die folgenden Inventory-Statusänderungen. Alle Ereignisse werden auf bestmögliche Weise ausgegeben.

Benutzerdefinierter Bestandstyp für eine bestimmte Instance gelöscht: Wenn eine Regel für die Überwachung dieses Ereignisses konfiguriert ist, erstellt EventBridge ein Ereignis, wenn ein benutzerdefinierter Bestandstyp für eine bestimmte verwaltete Instance gelöscht wird. EventBridge sendet ein Ereignis pro Knoten pro benutzerdefiniertem Bestandstyp. Hier ist ein Beispielereignismuster.

```
{
    "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"
    }
}
```

Benutzerdefiniertes Bestandstyp-Löschereignis für alle Instances: Wenn eine Regel für die Überwachung dieses Ereignisses konfiguriert ist, erstellt EventBridge ein Ereignis, wenn ein benutzerdefinierter Bestandstyp für alle verwalteten Knoten gelöscht wird. Hier ist ein Beispielereignismuster.

```
{
    "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"
    }
}
```

PutInventory ruft ein altes Schemaversionsereignis auf: Wenn eine Regel für die Überwachung dieses Ereignisses konfiguriert ist, erstellt EventBridge ein Ereignis, wenn ein PutInventory-Aufruf durchgeführt wird, der eine Schemaversion verwendet, die niedriger als das aktuelle Schema ist. Dieses Ereignis gilt für alle Inventararten. Hier ist ein Beispielereignismuster.

```
{
    "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"
    }
}
```

Informationen zum Konfigurieren von EventBridge für die Überwachung dieser Ereignisse finden Sie unter Konfigurieren von EventBridge für Systems Manager-Ereignisse.

Konfigurieren der Bestandserfassung

In diesem Abschnitt wird beschrieben, wie Sie die AWS Systems Manager Inventarerfassung auf einem oder mehreren verwalteten Knoten mithilfe der Systems Manager Manager-Konsole konfigurieren. Ein Beispiel für die Konfiguration der Inventarerfassung mithilfe von AWS Command Line Interface (AWS CLI) finden Sie unterWalkthroughs zu Systems Manager Inventory.

Wenn Sie die Inventarerfassung konfigurieren, erstellen Sie zunächst eine AWS Systems Manager State Manager Zuordnung. Systems Manager erfasst die Bestandsdaten, wenn der Zuordnungsstatus ausgeführt wird. Wenn Sie die Zuordnung nicht zuerst erstellen und versuchen, das `aws:softwareInventory` Plug-in beispielsweise mithilfe von aufzurufen, gibt das System den folgenden Fehler zurück: AWS Systems Manager Run Command `The aws:softwareInventory plugin can only be invoked via ssm-associate.`

> **Note**
>
> Beachten Sie das folgende Verhalten, wenn Sie mehrere Bestandszuordnungen für einen verwalteten Knoten erstellen:
>
> - Jedem Knoten kann eine Inventarzuordnung zugewiesen werden, die auf alle Knoten abzielt (--targets „Key=InstanceIds, Values=*“).
> - Jedem Knoten kann auch eine bestimmte Assoziation zugewiesen werden, die entweder Tag-Schlüssel/Wert-Paare oder eine Ressourcengruppe verwendet. AWS
> - Wenn einem Knoten mehrere Bestandszuordnungen zugewiesen sind, zeigt der Status Skipped (Übersprungen) für die Zuordnung an, die nicht ausgeführt wurde. Die zuletzt durchgeführte Zuordnung zeigt den aktuellen Status der Bestandszuordnung an.
> - Wenn einem Knoten mehrere Bestandszuordnungen zugewiesen sind und jede ein Tag-Schlüssel-Wert-Paar verwendet, können diese Bestandszuordnungen aufgrund des Tag-Konflikts nicht auf dem Knoten ausgeführt werden. Die Zuordnung wird weiterhin auf Knoten ausgeführt, bei denen der Tag-Schlüssel/Wert-Konflikt nicht besteht.

Bevor Sie beginnen

Führen Sie die folgenden Aufgaben aus, bevor Sie die Bestandserfassung konfigurieren.

- Aktualisieren Sie AWS Systems Manager SSM Agent die Knoten, die Sie inventarisieren möchten. Durch Ausführen der neuesten Version von SSM Agent stellen Sie sicher, dass Sie Metadaten für alle unterstützten Bestandstypen sammeln können. Informationen zur Aktualisierung von SSM Agent mithilfe von State Manager finden Sie unter Anleitung: Automatische Aktualisierung von SSM Agent (CLI).
- Überprüfen Sie, ob Sie die Einrichtungsanforderungen für Ihre Instances der Amazon Elastic Compute Cloud (Amazon EC2) und Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung erfüllt haben. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.
- Stellen Sie für Microsoft Windows-Knoten sicher, dass Ihr verwalteter Knoten mit Windows PowerShell 3.0 (oder höher) konfiguriert ist. SSM Agentverwendet das `ConvertTo-Json` Cmdlet in PowerShell , um die Windows Update-Inventardaten in das erforderliche Format zu konvertieren.
- (Optional) Erstellen Sie eine Resource Data Sync, um Bestandsdaten zentral in einem Amazon S3-Bucket zu speichern. Resource Data Sync aktualisiert die Daten dann automatisch, wenn neue Bestandsdaten erfasst werden. Weitere Informationen finden Sie unter Konfigurieren von Resource Data Sync für Inventory.
- (Optional) Erstellen Sie eine JSON-Datei für das Erfassen des benutzerdefinierten Bestands. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefiniertem Bestand.

Inventarisieren Sie alle verwalteten Knoten in Ihrem AWS-Konto

Sie können alle verwalteten Knoten in Ihrem inventarisieren, AWS-Konto indem Sie eine globale Inventarzuordnung erstellen. Eine globale Bestandszuordnung führt die folgenden Aktionen aus:

- Wendet die globale Inventarkonfiguration (Zuordnung) automatisch auf alle vorhandenen verwalteten Knoten in Ihrem an AWS-Konto. Verwaltete Knoten, die bereits über eine Bestandszuordnung verfügen, werden übersprungen, wenn die globale Bestandszuordnung angewendet wurde und ausgeführt wird. Wenn ein Knoten ausgelassen wird, zeigt die detaillierte Statusmeldung `Overridden By Explicit Inventory Association` an. Diese Knoten werden von der globalen Zuordnung übersprungen, sie melden aber immer noch den Bestand, wenn sie ihre zugewiesene Bestandszuordnung ausführen.
- Fügt neue Knoten, die in Ihrem erstellt wurden, automatisch AWS-Konto zur globalen Inventarzuordnung hinzu.

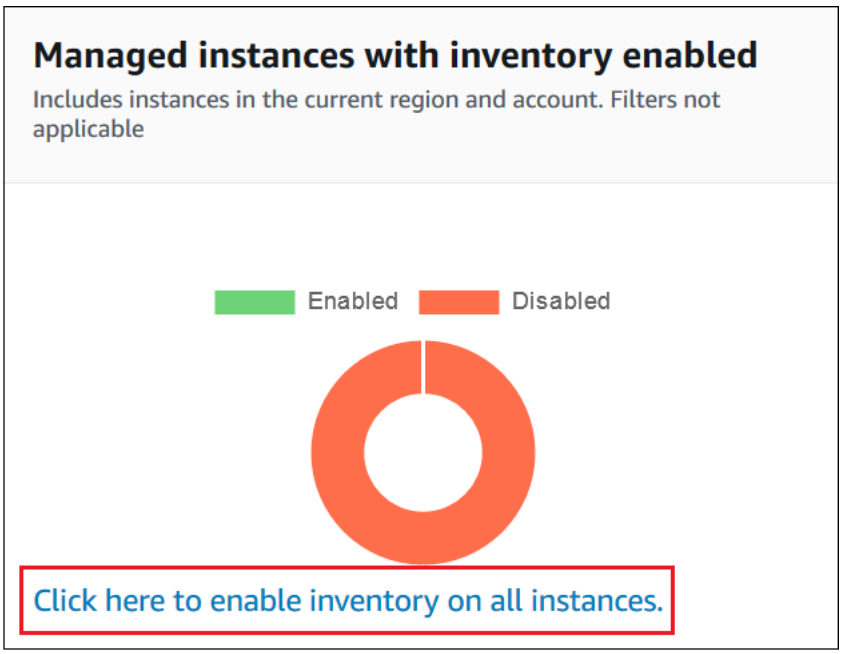
> **Note**
>
> - Wenn ein verwalteter Knoten für die globale Bestandszuordnung konfiguriert ist und Sie diesem Knoten eine bestimmte Zuordnung zuweisen, dann stellt Systems Manager Inventory die globale Zuordnung zurück und wendet die spezifische Zuordnung an.
> - Globale Bestandszuordnungen sind in SSM Agent-Version 2.0.790.0 oder höher verfügbar. Weitere Informationen zur Aktualisierung von SSM Agent auf Ihren Knoten finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

Konfigurieren der Bestandserfassung mit einem Klick (Konsole)

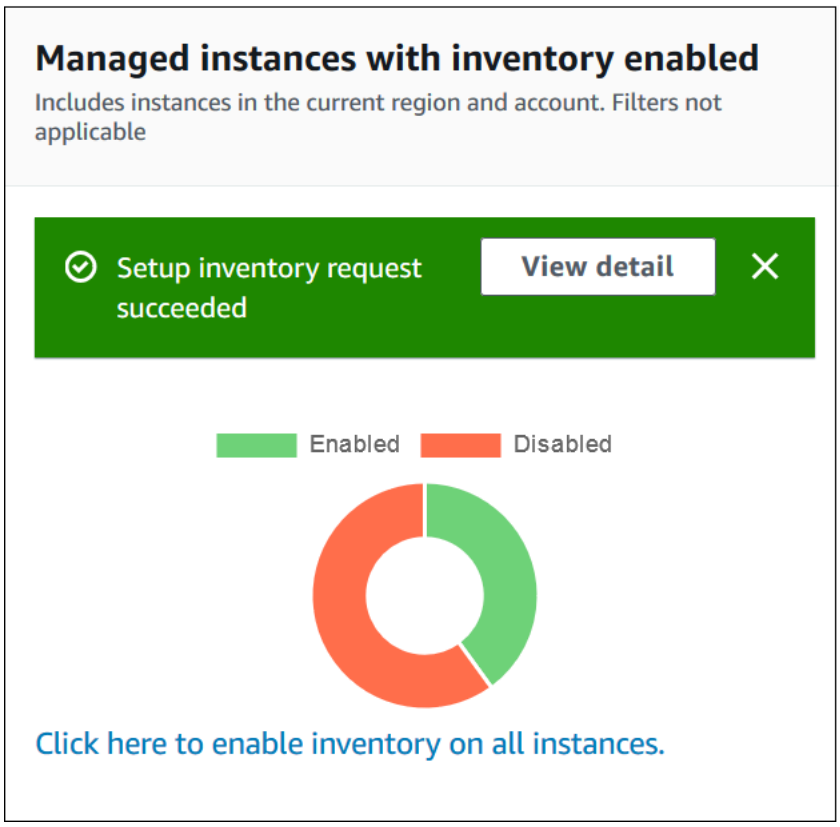
Gehen Sie wie folgt vor, um Systems Manager Inventory für alle verwalteten Knoten in Ihrem AWS-Konto und in einem einzigen zu konfigurieren AWS-Region.

So konfigurieren Sie all Ihre verwalteten Knoten in der aktuellen Region für Systems Manager Inventory

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie auf der Karte Managed instances with inventory enabled (Verwaltete Instances mit aktiviertem Bestand) die Option Click here to enable inventory on all instances (Klicken Sie hier, um den Bestand für alle Instances zu aktivieren) aus.



Wurde dieser Vorgang erfolgreich abgeschlossen, zeigt die Konsole die folgende Meldung an.



Je nach Anzahl der verwalteten Knoten in Ihrem Konto kann es einige Minuten dauern, bis die globale Bestandszuordnung angewendet wird. Warten Sie ein paar Minuten und aktualisieren

Sie dann die Seite. Überprüfen Sie, ob die Konfiguration des Bestands auf all Ihren verwalteten Knoten in der Grafik entsprechend angezeigt wird.

Konfigurieren der Erfassung über die Konsole

Dieser Abschnitt enthält Informationen zum Konfigurieren von Systems Manager Inventory für das Sammeln von Metadaten aus Ihren verwalteten Knoten mithilfe der Systems-Manager-Konsole. Sie können schnell Metadaten von allen Knoten in einem bestimmten AWS-Konto (und allen future Knoten, die möglicherweise in diesem Konto erstellt werden) sammeln oder Sie können Inventardaten selektiv mithilfe von Tags oder Knoten-IDs sammeln.

> **Note**
>
> Bevor Sie dieses Verfahren ausführen, prüfen Sie, ob bereits eine globale Bestandszuordnung existiert. Wenn bereits eine globale Bestandszuordnung vorhanden ist, wird diese bei jedem Start einer neuen Instance auf diese angewendet und die neue Instance wird inventarisiert.

So konfigurieren Sie die Bestandserfassung

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie die Option Setup Inventory.
4. Identifizieren Sie im Abschnitt Targets (Ziele) die Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie eine der folgenden Optionen auswählen.

   - Auswählen aller verwalteten Instances in diesem Konto – Diese Option wählt alle verwalteten Knoten aus, für die es keine Bestandszuordnung gibt. Wenn Sie diese Option auswählen, werden Knoten, für die bereits Bestandszuordnungen durchgeführt wurden, während der Bestandserfassung übersprungen und mit dem Status Skipped (Übersprungen) in den Bestandsergebnissen angezeigt. Weitere Informationen finden Sie unter Inventarisieren Sie alle verwalteten Knoten in Ihrem AWS-Konto.
   - Specifying a tag (Angeben eines Tags) – Mit dieser Option können Sie ein einzelnes Tag zum Identifizieren von Knoten in Ihrem Konto angeben, aus denen Sie den Bestand erfassen möchten. Wenn Sie ein Tag verwenden, wird jeder in der Zukunft mit demselben Tag erstellte

Knoten den Bestand ebenfalls melden. Wenn bereits eine Bestandszuordnung für alle Knoten besteht, überschreibt die Verwendung eines Tags zum Auswählen bestimmter Knoten als Ziel für einen anderen Bestand die Knoten-Mitgliedschaft in der Zielgruppe Alle verwalteten Instances. Verwaltete Knoten mit dem angegebenen Tag werden bei der künftigen Bestandserfassung von Allen verwalteten Instances übersprungen.

- Manually selecting instances (Manuelles Auswählen von Instances) – Mit dieser Option können Sie bestimmte verwaltete Knoten in Ihrem Konto auswählen. Die explizite Auswahl bestimmter Knoten überschreibt bei Verwendung dieser Option die Bestandszuordnungen im Ziel Alle verwalteten Instances. Der Knoten wird bei der künftigen Bestandserfassung von Alle verwalteten Instances übersprungen.

> **Note**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

5. Wählen Sie im Bereich Schedule (Planung) aus, wie oft das System die Bestand-Metadaten in Ihren Knoten erfassen soll.
6. Verwenden Sie die Listen im Bereich Parameters, um die verschiedenen Typen der Bestandserfassung zu aktivieren oder zu deaktivieren. Die folgenden Beispiele zeigen, wie Sie eine Bestandssuche für Files (Dateien) oder die Windows Registry (Windows-Registry) durchführen.

   Dateien

   - Unter Linux und macOS erfassen Sie Metadaten von .sh-Dateien im Verzeichnis `/home/ec2-user` ohne die Unterverzeichnisse.

   ```
   [{"Path":"/home/ec2-user","Pattern":["*.sh", "*.sh"],"Recursive":false}]
   ```

   - Unter Windows erfassen Sie rekursiv Metadaten aller ".exe" -Dateien im Ordner Programme, einschließlich der Unterverzeichnisse.

   ```
   [{"Path":"C:\Program Files","Pattern":["*.exe"],"Recursive":true}]
   ```

   - Unter Windows erfassen Sie Metadaten bestimmter Log-Muster.

```
[{"Path":"C:\ProgramData\Amazon","Pattern":["*amazon*.log"],"Recursive":true}]
```

- Beschränken Sie die Anzahl der Verzeichnisse, wenn Sie eine rekursive Erfassung durchführen.

```
[{"Path":"C:\Users","Pattern":["*.ps1"],"Recursive":true, "DirScanLimit": 1000}]
```

Windows-Registrierung

- Erfassen Sie alle Schlüssel und Werte rekursiv für einen bestimmten Pfad.

```
[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon","Recursive": true}]
```

- Erfassen Sie alle Schlüssel und Werte für einen bestimmten Pfad (die rekursive Suche ist deaktiviert).

```
[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Intel\PSIS\PSIS_DECODER", "Recursive":
 false}]
```

- Erfasst einen bestimmten Schlüssel unter Verwendung der Option `ValueNames`.

```
{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon\MachineImage","ValueNames":
["AMIName"]}
```

Weitere Informationen zur Erfassung von Datei- und Windows-Registry-Bestand finden Sie unter Arbeiten mit Datei- und Windows-Registrierungsbestand.

7. Wählen Sie im Bereich Advanced (Erweitert) die Option Sync inventory execution logs to an Amazon S3 bucket (Bestandsausführungsprotokolle mit einem Amazon S3-Bucket synchronisieren), wenn Sie den Ausführungsstatus der Zuordnung in einem S3-Bucket speichern möchten.
8. Wählen Sie die Option Setup Inventory. Systems Manager erstellt eine State Manager-Zuordnung und führt sofort Inventory auf den Knoten aus.
9. Wählen Sie im Navigationsbereich State Manager aus. Überprüfen Sie, ob eine neue Zuordnung erstellt wurde, die das **`AWS-GatherSoftwareInventory`**-Dokument verwendet. Der Zuordnungszeitplan verwendet einen Rate-Ausdruck. Überprüfen Sie auch, ob das Feld Status den Wert Success enthält. Wenn Sie die Option Sync inventory execution logs to an S3

bucket (Inventory-Ausführungsprotokolle mit einem S3-Bucket synchronisieren) ausgewählt haben, können Sie die Protokolldaten nach einigen Minuten in Amazon S3 anzeigen. Wenn Sie Bestandsdaten für einen bestimmten Knoten anzeigen möchten, klicken Sie auf Managed Instances (Verwaltete Instances) im Navigationsbereich.

10. Wählen Sie einen Knoten und anschließend View details (Details anzeigen).
11. Wählen Sie auf der Knoten-Detailseite Inventory aus. Verwenden Sie die Inventory type-Listen, um den Bestand zu filtern.

Arbeiten mit Systems Manager-Bestandsdaten

Dieser Abschnitt enthält Themen, in denen beschrieben wird, wie AWS Systems Manager-Bestandsdaten abgefragt und aggregiert werden.

Themen

- Abfragen von Bestandsdaten aus mehreren Regionen und Konten
- Abfragen der Bestandserfassung mithilfe von Filtern
- Aggregieren von Bestandsdaten

Abfragen von Bestandsdaten aus mehreren Regionen und Konten

AWS Systems Manager Inventory ist in Amazon Athena integriert, sodass Sie Inventardaten von mehreren AWS-Regionen und AWS-Konten abfragen können. Die Athena-Integration verwendet die Ressourcendatensynchronisierung, sodass Sie Inventardaten von all Ihren verwalteten Knoten auf der Detailansichtsseite in der AWS Systems Manager Konsole anzeigen können.

> **Important**
>
> Diese Funktion wird verwendet AWS Glue , um die Daten in Ihrem Amazon Simple Storage Service (Amazon S3) -Bucket zu crawlen, und Amazon Athena, um die Daten abzufragen. Abhängig davon, wie viele Daten durchsucht und abgefragt werden, werden Ihnen diese Services in Rechnung gestellt. Mit AWS Glue zahlen Sie einen Stundensatz, der sekundengenau abgerechnet wird, für Crawler (Erkennung von Daten) und ETL-Jobs (Verarbeitung und Laden von Daten). Bei Athena richtet sich die Gebühr nach der Menge der pro Abfrage durchsuchten Daten. Wir empfehlen Ihnen, die Preisrichtlinien für diese Services zu lesen, bevor Sie die Amazon Athena-Integration mit Systems Manager Inventory

verwenden. Weitere Informationen finden Sie unter Amazon Athena – Preise und AWS Glue - Preise.

Sie können Inventory-Daten auf der Seite Detailed View (Detailsansicht in allen AWS-Regionen anzeigen, in denen Amazon Athena verfügbar ist. Eine Liste der unterstützten Regionen finden Sie unter Amazon Athena-Service-Endpunkte im Allgemeine Amazon Web Services-Referenz.

Bevor Sie beginnen

Die Athena-Integration verwendet Resource Data Sync. Sie müssen Resource Data Sync einrichten und konfigurieren, um dieses Feature zu verwenden. Weitere Informationen finden Sie unter Konfigurieren von Resource Data Sync für Inventory.

Beachten Sie außerdem, dass die Detailed View (Detailansicht Bestandsdaten für den Besitzer des zentralen Amazon S3-Buckets anzeigt, der von Resource Data Sync verwendet wird. Wenn Sie nicht der Besitzer des zentralen Amazon S3-Buckets sind, werden Ihnen auf der Seite Detailed View (Detailansicht) keine Bestandsdaten angezeigt.

Konfigurieren des Zugriffs

Bevor Sie Daten aus mehreren Konten und Regionen auf der Seite Detailsansicht in der Systems-Manager-Konsole abfragen und anzeigen können, müssen Sie Ihre (IAM)-Entität mit Berechtigungen zur Ansicht der Daten konfigurieren.

Wenn die Inventardaten in einem Amazon S3 S3-Bucket gespeichert sind, der die Verschlüsselung AWS Key Management Service (AWS KMS) verwendet, müssen Sie auch Ihre IAM-Entität und die `Amazon-GlueServiceRoleForSSM` Servicerolle für die AWS KMS Verschlüsselung konfigurieren.

Themen

- Konfigurieren Ihrer IAM-Entität für den Zugriff auf die Seite Detailansicht
- (Optional) Konfigurieren Sie Berechtigungen für die Anzeige AWS KMS verschlüsselter Daten

Konfigurieren Ihrer IAM-Entität für den Zugriff auf die Seite Detailansicht

Im Folgenden werden die Mindestberechtigungen beschrieben, die zum Anzeigen von Bestandsdaten auf der Seite Detailansicht erforderlich sind.

Die von **`AWSQuicksightAthenaAccess`** verwaltete Richtlinie

Die folgende `PassRole` und der zusätzliche erforderliche Berechtigungsblock

```
{
    "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": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "glue.amazonaws.com"
                }
            }
        },
        {
            "Sid": "iamRoleCreation",
           "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:iam::account_ID:role/*"
        },
        {
            "Sid": "iamPolicyCreation",
            "Effect": "Allow",
            "Action": "iam:CreatePolicy",
            "Resource": "arn:aws:iam::account_ID:policy/*"
        }
    ]
```

```
}
```

(Optional) Wenn der Amazon S3 S3-Bucket, der zum Speichern von Inventardaten verwendet wird AWS KMS, mithilfe von verschlüsselt ist, müssen Sie der Richtlinie auch den folgenden Block hinzufügen.

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:Region:account_ID:key/key_ARN"
    ]
}
```

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter Erstellen einer Rolle für einen externen Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter Erstellen einer Rolle für einen IAM-Benutzer im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

(Optional) Konfigurieren Sie Berechtigungen für die Anzeige AWS KMS verschlüsselter Daten

Wenn der Amazon S3 S3-Bucket, der zum Speichern von Inventardaten verwendet wird, mithilfe von AWS Key Management Service (AWS KMS) verschlüsselt ist, müssen Sie Ihre IAM-Entität und die GlueServiceRoleForAmazon-SSM-Rolle mit `kms:Decrypt` Berechtigungen für den AWS KMS Schlüssel konfigurieren.

Bevor Sie beginnen

Um die `kms:Decrypt` Berechtigungen für den AWS KMS Schlüssel bereitzustellen, fügen Sie Ihrer IAM-Entität den folgenden Richtlinienblock hinzu:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:Region:account_ID:key/key_ARN"
    ]
}
```

Falls Sie dies noch nicht getan haben, schließen Sie dieses Verfahren ab und fügen Sie `kms:Decrypt` Berechtigungen für den AWS KMS Schlüssel hinzu.

Gehen Sie wie folgt vor, um die GlueServiceRoleForAmazon-SSM-Rolle mit `kms:Decrypt` Berechtigungen für den AWS KMS Schlüssel zu konfigurieren.

So konfigurieren Sie die GlueServiceRoleForAmazon-SSM-Rolle mit Berechtigungen **`kms:Decrypt`**

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus, und verwenden Sie dann das Suchfeld, um die GlueServiceRoleForAmazon-SSM-Rolle zu suchen. Die Seite Summary (Übersicht) wird geöffnet.
3. Verwenden Sie das Suchfeld, um die GlueServiceRoleForAmazon-SSM-Rolle zu finden. Wählen Sie den Rollennamen aus. Die Seite Summary (Übersicht) wird geöffnet.
4. Wählen Sie den Rollennamen aus. Die Seite Summary (Übersicht) wird geöffnet.
5. Wählen Sie Inline-Richtlinie hinzufügen. Die Seite Create policy (Richtlinie erstellen) wird geöffnet.
6. Wählen Sie den Tab JSON.
7. Löschen Sie den vorhandenen JSON-Text im Editor, kopieren Sie die folgende Richtlinie und fügen Sie sie in den JSON-Editor ein.

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

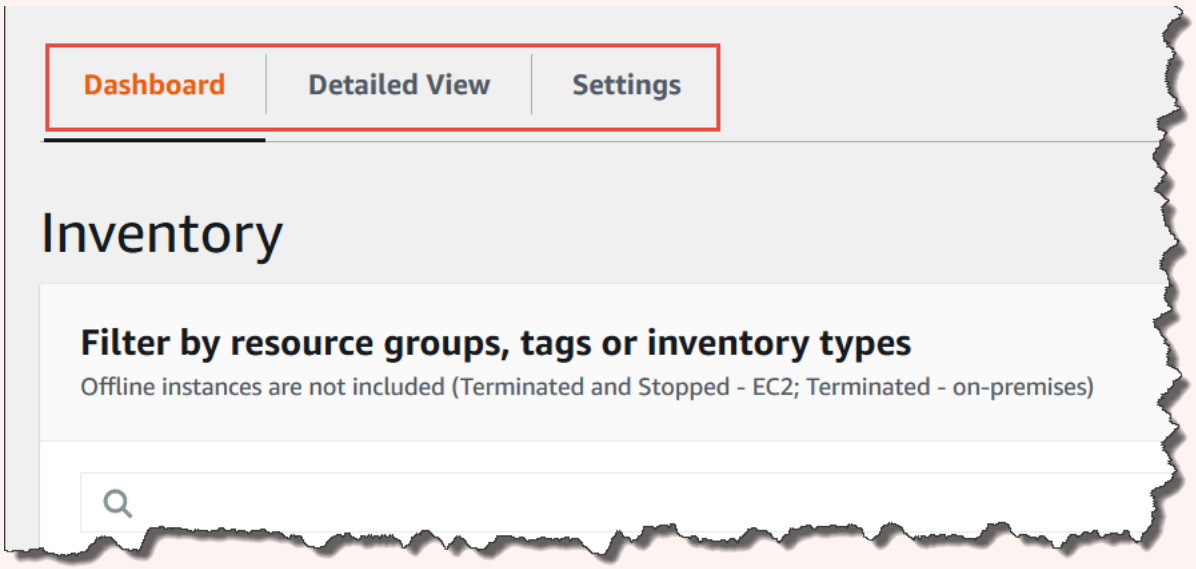
```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:Region:account_ID:key/key_ARN"
            ]
        }
    ]
}
```

8. Wählen Sie Review policy (Richtlinie überprüfen) aus.
9. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) im Feld Name einen Namen ein.
10. Wählen Sie Richtlinie erstellen aus.

Abfragen von Daten auf der Seite „Inventory Detailed View (Detaillierte Bestandsansicht)“

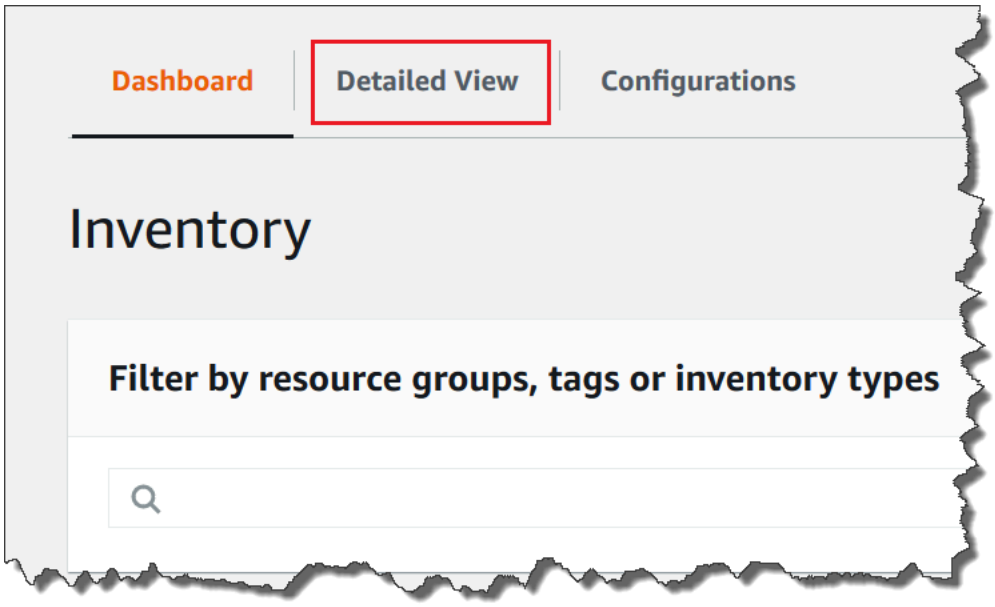
Gehen Sie wie folgt vor, um Inventardaten von mehreren AWS-Regionen und AWS-Konten auf der Seite Inventar Detailed View von Systems Manager anzuzeigen.

> **Important**
>
> Die Seite Inventory Detailed View (Detailansicht) ist nur in AWS-Regionen verfügbar, die Amazon Athena anbieten. Wenn die folgenden Registerkarten nicht auf der Seite Systems Manager Inventory angezeigt werden, bedeutet dies, dass Athena nicht in der Region verfügbar ist und Sie die Detailansicht nicht verwenden können, um Daten abzufragen.

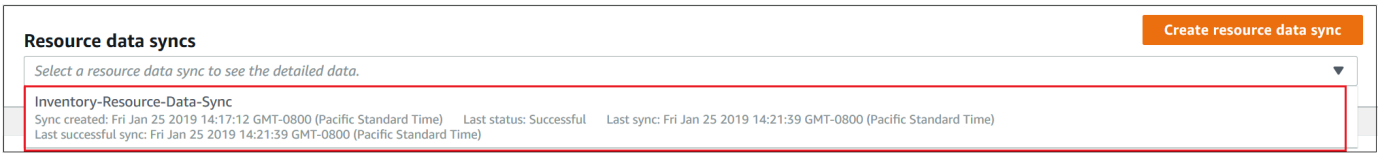


Bestandsdaten aus mehreren Regionen und Konten in der AWS Systems Manager -Konsole anzeigen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie die Registerkarte Detailed View (Detaillierte Ansicht) aus.

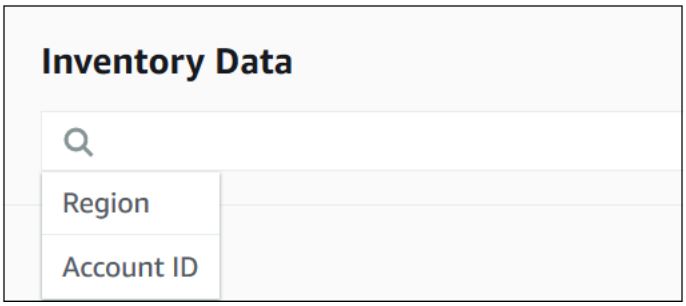


4. Wählen Sie die Resource Data Sync aus, für die Sie Daten abfragen möchten.



5. Wählen Sie in der Liste Inventory Type (Bestandstyp) den Typ der Bestandsdaten aus, die Sie abfragen möchten, und drücken Sie dann Enter.

6. Um die Daten zu filtern, wählen Sie die Filterleiste aus und wählen Sie dann eine Filteroption aus.



Sie können die Schaltfläche Export to CSV (Exportieren in CSV) verwenden, um die aktuelle Abfrage in einem Tabellenkalkulationsprogramm wie Microsoft Excel anzuzeigen. Sie können auch die Schaltflächen Query History (Abfrageverlauf) und Run Advanced Queries (Erweiterte Abfragen ausführen) verwenden, um mit Ihren Daten in Amazon Athena zu interagieren.

Bearbeiten des Zeitplans für den AWS Glue -Crawler

AWS Glue crawlt standardmäßig zweimal täglich die Inventardaten im zentralen Amazon S3 S3-Bucket. Wenn Sie häufig die Arten der auf Ihren Knoten zu erfassenden Daten ändern, möchten Sie möglicherweise Sie die Daten häufiger durchsuchen, wie im folgenden Verfahren beschrieben.

> **Important**
>
> AWS Glue AWS-Konto berechnet Ihnen für Crawler (Erkennung von Daten) und ETL-Jobs (Verarbeitung und Laden von Daten) einen Stundensatz, der sekundenweise abgerechnet wird. Bevor Sie den Crawler-Zeitplan anzeigen, rufen Sie die AWS Glue -Preisliste auf.

So ändern Sie den Bestandsdatencrawler-Zeitplan

1. Öffnen Sie die AWS Glue Konsole unter https://console.aws.amazon.com/glue/.
2. Wählen Sie im Navigationsbereich Crawlers (Crawler) aus.
3. Wählen Sie in der Liste der Crawler die Option neben dem Systems Manager Inventory-Crawler aus. Der Crawler-Name verwendet das folgende Format:

   `AWSSystemsManager-DOC-EXAMPLE-BUCKET-`*`Region`*`-`*`account_ID`*
4. Wählen Sie Action (Aktion) und Edit crawler (Crawler bearbeiten) aus.
5. Wählen Sie im Navigationsbereich Schedule (Zeitplan) aus.
6. Geben Sie im Feld Cron expression (cron-Ausdruck) einen neuen Zeitplan mit einem Cron-Format an. Weitere Informationen zum Cron-Format finden Sie unter Zeitpläne für Aufträge und Crawler im AWS Glue Developer Guide.

> **Important**
>
> Sie können den Crawler pausieren, damit keine Gebühren mehr von anfallen. AWS Glue Wenn Sie den Crawler aussetzen oder die Häufigkeit ändern, damit die Daten weniger häufig durchsucht werden, zeigt Inventory Detailed View (Detaillierte Ansicht) möglicherweise Daten an, die nicht aktuell sind.

Abfragen der Bestandserfassung mithilfe von Filtern

Nachdem Sie Bestandsdaten erfasst haben, können Sie den Filter-Funktionen in AWS Systems Manager nutzen, um eine Liste der verwalteten Knoten abzufragen, die bestimmte Filterkriterien erfüllen.

So fragen Sie Knoten basierend auf Bestandsfiltern ab

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie im Abschnitt Filter by resource groups, tags or inventory types die Filteroption. Eine Liste vordefinierter Filter wird angezeigt.

4. Wählen Sie ein Attribut, nach dem gefiltert werden soll. Wählen Sie zum Beispiel **`AWS:Application`** aus. Wenn Sie dazu aufgefordert werden, wählen Sie ein sekundäres Attribut, nach dem gefiltert werden soll. Wählen Sie zum Beispiel **`AWS:Application.Name`** aus.
5. Wählen Sie eine Begrenzung in der Liste aus. Wählen Sie z. B. Begin with. Im Filter wird ein Textfeld angezeigt.
6. Geben Sie einen Wert in das Textfeld ein. Geben Sie beispielsweise Amazon ein (SSM Agent ist als Amazon SSM Agent benannt).
7. Drücken Sie Enter. Das System gibt eine Liste der verwalteten Knoten zurück, die einen Anwendungsnamen haben, der mit dem Wort Amazon beginnt.

> **Note**
>
> Sie können mehrere Filter kombinieren, um die Suche zu verfeinern.

Aggregieren von Bestandsdaten

Nach der Konfiguration Ihrer verwalteten Knoten für AWS Systems Manager Inventory können Sie die aggregierte Anzahl der Bestandsdaten anzeigen. Nehmen wir beispielsweise an, Sie haben Dutzende oder Hunderte von verwalteten Knoten zur Erfassung des Bestandstyps `AWS:Application` konfiguriert. Mithilfe der Informationen in diesem Abschnitt können Sie die genaue Zahl der Knoten anzeigen, die zum Erfassen dieser Daten konfiguriert sind.

Sie können außerdem spezifische Bestandsdetails durch Aggregieren eines Datentyps sehen. Beispiel: Der Bestandstyp `AWS:InstanceInformation` erfasst Betriebssystem-Plattforminformationen mit dem Datentyp `Platform`. Durch die Aggregierung von Daten für den Datentyp `Platform` können Sie schnell sehen, wie viele Knoten Windows, wie viele Linux ausführen und wie viele macOS ausführen.

Die Verfahren in diesem Abschnitt beschreiben, wie Sie die aggregrierte Zahl der Bestandsdaten mithilfe der AWS Command Line Interface (AWS CLI) anzeigen. Außerdem können Sie vorkonfigurierte aggregierte Zählungen in der AWS Systems Manager-Konsole auf der Seite Inventory (Bestand) anzeigen. Diese vorkonfigurierten Dashboards werden als Inventory Insights (Bestandseinblicke) bezeichnet. Sie bieten eine 1-Klick-Wiederherstellung Ihrer Bestand-Konfigurationsprobleme.

Beachten Sie die folgenden wichtigen Details zu den Aggregationszählungen von Bestandsdaten:

- Wenn Sie einen verwalteten Knoten beenden, der zum Sammeln von Bestandsdaten konfiguriert ist, behält Systems Manager die Bestandsdaten 30 Tage lang bei und löscht sie anschließend. Für ausgeführte Knoten löscht das System alle Bestandsdaten, die älter als 30 Tage sind. Wenn Sie Bestandsdaten länger als 30 Tage speichern müssen, können Sie mit AWS Config einen Verlauf aufzeichnen oder die Daten regelmäßig abfragen und in einen Amazon Simple Storage Service (Amazon S3)-Bucket hochladen.
- Wenn ein Knoten zuvor konfiguriert wurde, um einen spezifischen Bestandsdatentyp zu melden, z. B. `AWS:Network`, und Sie die Konfiguration später so ändern, dass die Daten dieses Typs nicht mehr erfasst werden, zeigt die Aggregationszählung nach wie vor `AWS:Network`-Daten an, bis der Knoten beendet wurde und 30 Tage vergangen sind.

Weitere Informationen zum schnellen Konfigurieren und Erfassen von Bestandsdaten aus allen Knoten in einem bestimmten AWS-Konto (und allen künftigen Knoten, die in diesem Konto erstellt werden) finden Sie unter Konfigurieren der Erfassung über die Konsole.

Themen

- Aggregieren von Bestandsdaten zum Anzeigen der Anzahl von Knoten, die bestimmte Arten von Daten erfassen
- Aggregieren von Bestandsdaten mit Gruppen, um zu sehen, welche Knoten zur Erfassung eines Bestandstyps konfiguriert sind und welche nicht

Aggregieren von Bestandsdaten zum Anzeigen der Anzahl von Knoten, die bestimmte Arten von Daten erfassen

Sie können die AWS Systems Manager GetInventory-API-Operation verwenden, um aggregierte Zählungen der Knoten anzuzeigen, die einen oder mehrere Bestands- und Datentypen erfassen. Zum Beispiel, ermöglicht der `AWS:InstanceInformation`-Inventory-Typ das Anzeigen der aggregierten Zahl von Betriebssystemen mithilfe der GetInventory-API-Operation und dem `AWS:InstanceInformation.PlatformType`-Datentyp. Hier finden Sie ein Beispiel für den AWS CLI-Befehl und die Ausgabe.

```
aws ssm get-inventory --aggregators "Expression=AWS:InstanceInformation.PlatformType"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "Entities":[
      {
         "Data":{
            "AWS:InstanceInformation":{
               "Content":[
                  {
                     "Count":"7",
                     "PlatformType":"windows"
                  },
                  {
                     "Count":"5",
                     "PlatformType":"linux"
                  }
               ]
            }
         }
      }
   ]
}
```

Erste Schritte

Bestimmen Sie die Bestands- und Datentypen, für die Sie Zählungen anzeigen möchten. Sie können eine Liste der Bestandstypen und Datentypen anzeigen, die die Aggregation unterstützen, indem Sie den folgenden Befehl im Fenster AWS CLI ausführen.

```
aws ssm get-inventory-schema --aggregator
```

Der Befehl gibt eine JSON-Liste mit Bestands- und Datentypen zurück, die die Aggregation unterstützen. Das Feld TypeName enthält die unterstützten Bestandstypen. Das Feld Name zeigt die einzelnen Datentypen. Der Bestandstyp `AWS:Application` in der folgenden Listen enthält z. B. Datentypen für `Name` und `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"
                }
            ]
```

```
        }
    ]
}
```

Sie können Daten für einen der gelisteten Bestandstypen aggregieren, indem Sie einen Befehl erstellen, der die folgende Syntax verwendet.

```
aws ssm get-inventory --aggregators "Expression=InventoryType.DataType"
```

Hier sind einige Beispiele.

Beispiel 1

Dieses Beispiel aggregiert eine Zählung der Windows-Rollen, die von Ihren Knoten verwendet werden.

```
aws ssm get-inventory --aggregators "Expression=AWS:WindowsRole.Name"
```

Beispiel 2

Dieses Beispiel aggregiert eine Zählung der Anwendungen, die auf Ihren Knoten installiert sind.

```
aws ssm get-inventory --aggregators "Expression=AWS:Application.Name"
```

Kombinieren von mehreren Aggregatoren

Sie können auch mehrere Bestands- und Datentypen in einem Befehl kombinieren, um die Daten besser zu verstehen. Hier sind einige Beispiele.

Beispiel 1

Dieses Beispiel aggregiert eine Zählung der Betriebssystemtypen, die von Ihren Knoten verwendet werden. Außerdem wird der spezifische Name der Betriebssysteme zurückgegeben.

```
aws ssm get-inventory --aggregators '[{"Expression":
 "AWS:InstanceInformation.PlatformType", "Aggregators":[{"Expression":
 "AWS:InstanceInformation.PlatformName"}]}]'
```

Beispiel 2

Dieses Beispiel aggregiert eine Zählung der Anwendungen, die auf Ihren Knoten ausgeführt werden, sowie die spezifische Version der einzelnen Anwendungen.

```
aws ssm get-inventory --aggregators '[{"Expression": "AWS:Application.Name",
 "Aggregators":[{"Expression": "AWS:Application.Version"}]}]'
```

Wenn Sie möchten, können Sie einen Aggregationsausdruck mit einem oder mehreren Bestands- und Datentypen in einer JSON-Datei erstellen und die Datei über die AWS CLI aufrufen. Die JSON-Datei muss die folgende Syntax verwenden.

```
[
       {
           "Expression": "string",
           "Aggregators": [
               {
                  "Expression": "string"
               }
           ]
       }
]
```

Sie müssen die Datei mit der Erweiterung „.json” speichern.

Im folgenden Beispiel werden mehrere Bestands- und Datentypen verwendet.

```
[
       {
           "Expression": "AWS:Application.Name",
           "Aggregators": [
               {
                   "Expression": "AWS:Application.Version",
                   "Aggregators": [
                     {
                     "Expression": "AWS:InstanceInformation.PlatformType"
                     }
                   ]
               }
           ]
       }
]
```

Rufen Sie die Datei mit folgendem Befehl über die AWS CLI auf.

```
aws ssm get-inventory --aggregators file://file_name.json
```

Der Befehl gibt Informationen wie die folgenden zurück.


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

Aggregieren von Bestandsdaten mit Gruppen, um zu sehen, welche Knoten zur Erfassung eines Bestandstyps konfiguriert sind und welche nicht

Gruppen in Systems Manager Inventory ermöglichen es Ihnen, schnell eine Anzahl der verwalteten Knoten zu sehen, die für das Erfassen einer oder mehrerer Bestandstypen konfiguriert bzw. nicht konfiguriert sind. Mit Gruppen geben Sie einen oder mehrere Bestandstypen sowie einen Filter an, der den `exists`-Operator verwendet.

Beispiel: Angenommen, Sie haben vier verwaltete Knoten zum Erfassen der folgenden Bestandstypen konfiguriert:

- Knoten 1: `AWS:Application`
- Knoten 2: `AWS:File`
- Knoten 3: `AWS:Application`, `AWS:File`
- Knoten 4: `AWS:Network`

Sie können den folgenden Befehl in der AWS CLI ausführen, um zu sehen, wie viele Knoten zum Erfassen der Bestandstypen `AWS:Application` und `AWS:File inventory` konfiguriert sind. Die Antwort gibt auch die Anzahl der Knoten zurück, die nicht zum Erfassen dieser beiden Bestandstypen konfiguriert sind.

```
aws ssm get-inventory --aggregators
 'Groups=[{Name=ApplicationAndFile,Filters=[{Key=TypeName,Values=[AWS:Application],Type=Exists}
{Key=TypeName,Values=[AWS:File],Type=Exists}]}]'
```

Die Befehlsantwort zeigt, dass nur ein verwalteter Knoten zum Erfassen der beiden Bestandstypen `AWS:Application` und `AWS:File` konfiguriert ist.

```
{
   "Entities":[
      {
         "Data":{
            "ApplicationAndFile":{
               "Content":[
                  {
                     "notMatchingCount":"3"
                  },
                  {
                     "matchingCount":"1"
```

```
                }
            ]
        }
    }
  }
]
}
```

> **Note**
>
> Gruppen geben keine Datentypzahlen zurück. Außerdem können Sie die Ergebnisse nicht aufschlüsseln, um die IDs von Knoten zu sehen, die zum Erfassen des Bestandstyps konfiguriert bzw. nicht konfiguriert sind.

Wenn Sie möchten, können Sie einen Aggregationsausdruck mit einem oder mehreren Bestandstypen in einer JSON-Datei erstellen und die Datei über die AWS CLI aufrufen. Die JSON-Datei muss die folgende Syntax verwenden:

```
{
   "Aggregators":[
      {
         "Groups":[
            {
               "Name":"Name",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
                        "Inventory_type"
                     ],
                     "Type":"Exists"
                  },
                  {
                     "Key":"TypeName",
                     "Values":[
                        "Inventory_type"
                     ],
                     "Type":"Exists"
                  }
               ]
            }
```

```
         ]
      }
   ]
}
```

Sie müssen die Datei mit der Erweiterung „.json” speichern.

Rufen Sie die Datei mit folgendem Befehl über die AWS CLI auf.

```
aws ssm get-inventory --cli-input-json file://file_name.json
```

Weitere Beispiele

Die folgenden Beispiele zeigen Ihnen, wie Sie Bestandsdaten aggregieren, um zu sehen, welche verwalteten Knoten zum Erfassen der angegebenen Bestandstypen konfiguriert bzw. nicht konfiguriert sind. Diese Beispiele verwenden die AWS CLI. Jedes Beispiel enthält einen vollständigen Befehl mit Filtern, die Sie über die Befehlszeile ausführen können, und eine input.json-Beispieldatei, falls Sie es vorziehen, die Informationen in eine Datei einzugeben.

Beispiel 1

Dieses Beispiel aggregiert eine Zählung der Knoten, die zum Erfassen entweder des Bestandstyps `AWS:Application` oder des Typs `AWS:File` konfiguriert bzw. nicht konfiguriert sind.

Führen Sie den folgenden Befehl über die AWS CLI aus.

```
aws ssm get-inventory --aggregators
 'Groups=[{Name=ApplicationORFile,Filters=[{Key=TypeName,Values=[AWS:Application,
 AWS:File],Type=Exists}]}]'
```

Wenn Sie eine Datei verwenden möchten, kopieren Sie das folgende Beispiel in eine Datei und speichern Sie sie unter dem Namen input.json.

```
{
   "Aggregators":[
      {
         "Groups":[
            {
               "Name":"ApplicationORFile",
               "Filters":[
                  {
                     "Key":"TypeName",
```

```
                    "Values":[
                       "AWS:Application",
                       "AWS:File"
                    ],
                    "Type":"Exists"
                 }
              ]
           }
        ]
     }
  ]
}
```

Führen Sie den folgenden Befehl über die AWS CLI aus.

```
aws ssm get-inventory --cli-input-json file://input.json
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
   "Entities":[
      {
         "Data":{
            "ApplicationORFile":{
               "Content":[
                  {
                     "notMatchingCount":"1"
                  },
                  {
                     "matchingCount":"3"
                  }
               ]
            }
         }
      }
   ]
}
```

Beispiel 2

Dieses Beispiel aggregiert eine Zählung der Knoten, die zum Erfassen des Bestandstyps `AWS:Application`, `AWS:File` und `AWS:Network` konfiguriert bzw. nicht konfiguriert sind.

Führen Sie den folgenden Befehl über die AWS CLI aus.

```
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}]}]'
```

Wenn Sie eine Datei verwenden möchten, kopieren Sie das folgende Beispiel in eine Datei und speichern Sie sie unter dem Namen 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"
                 }
              ]
           }
        ]
     }
  ]
}
```

Führen Sie den folgenden Befehl über die AWS CLI aus.

```
aws ssm get-inventory --cli-input-json file://input.json
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
   "Entities":[
      {
         "Data":{
            "Application":{
               "Content":[
                  {
                     "notMatchingCount":"2"
                  },
                  {
                     "matchingCount":"2"
                  }
               ]
            },
            "File":{
               "Content":[
                  {
                     "notMatchingCount":"2"
                  },
                  {
                     "matchingCount":"2"
                  }
               ]
            },
            "Network":{
```

```
                "Content":[
                   {
                      "notMatchingCount":"3"
                   },
                   {
                      "matchingCount":"1"
                   }
                ]
             }
          }
       }
    ]
}
```

Arbeiten mit benutzerdefiniertem Bestand

Ordnen Sie Ihren Knoten alle möglichen Metadaten zu, indem Sie einen benutzerdefinierten AWS Systems Manager-Inventory-Bestand erstellen. Nehmen wir z. B. an, dass Sie eine große Anzahl von Servern in Racks in Ihrem Rechenzentrum verwalten; und diese Server als von Systems Manager verwaltete Knoten konfiguriert wurden. Derzeit speichern Sie die Informationen zu den Standorten der Server-Racks in einer Tabelle. Mit einem benutzerdefinierten Bestand können Sie die Rack-Standorte der einzelnen Knoten als Metadaten auf dem Knoten angeben. Wenn Sie den Bestand mit Systems Manager erfassen, werden die Metadaten zusammen mit den anderen Bestandsmetadaten erfasst. Anschließend können Sie alle Bestandsmetadaten in einen zentralen Amazon S3-Bucket portieren, indem Sie Resource Data Sync (Ressourcendaten-Synchronisation) verwenden und die Daten abfragen.

> **Note**
>
> Systems Manager unterstützt maximal 20 benutzerdefinierte Bestandstypen pro AWS-Konto.

Sie können einem Knoten einen benutzerdefinierten Bestand zuordnen, indem Sie die PutInventory-API-Operation von Systems Manager verwenden, wie in Walkthrough: Zuweisen benutzerdefinierter Bestands-Metadaten zu einem verwalteten Knoten beschrieben. Oder Sie können eine JSON-Datei für den benutzerdefinierten Bestand erstellen und diese auf den Knoten hochladen. In diesem Abschnitt wird beschrieben, wie Sie die JSON-Datei erstellen.

Die folgende Beispiel-JSON-Datei mit benutzerdefiniertem Bestand gibt Rack-Informationen zu einem On-Premises-Server an. Dieses Beispiel gibt einen Typ von benutzerdefinierten Bestandsdaten

("TypeName": "Custom:RackInformation") an, mit mehreren Einträgen unter Content, die die Daten beschreiben.

```
{
    "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"
      }
 }
```

Sie können wie im folgenden Beispiel gezeigt auch verschiedene Einträge im Abschnitt Content angeben.

```
{
"SchemaVersion": "1.0",
"TypeName": "Custom:PuppetModuleInfo",
    "Content": [{
        "Name": "puppetlabs/aws",
        "Version": "1.0"
      },
      {
        "Name": "puppetlabs/dsc",
        "Version": "2.0"
      }
    ]
}
```

Das JSON-Schema für den benutzerdefinierten Bestand erfordert die Abschnitte SchemaVersion, TypeName und Content, aber Sie können die Informationen in diesen Abschnitten definieren.

```
{
    "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"
      }
}
```

Der `TypeName`-Wert ist auf 100 Zeichen begrenzt. Außerdem muss der `TypeName`-Wert mit dem großgeschriebenen Wort `Custom` beginnen. Zum Beispiel `Custom:PuppetModuleInfo`. Daher würden die folgenden Beispiele zu einer Ausnahme führen: `CUSTOM:PuppetModuleInfo`, `custom:PuppetModuleInfo`.

Der Abschnitt `Content` enthält Attribute und *`Daten`*. Beachten Sie, dass bei diesen Elementen Groß- und Kleinschreibung nicht berücksichtigt wird. Wenn Sie jedoch ein Attribut definieren (z. B: "`Vendor`": "DELL") müssen Sie dieses Attribut in Ihren Dateien für den benutzerdefinierten Bestand konsistent referenzieren. Wenn Sie in einer Datei "`Vendor`": "DELL" (mit einem großen „V” in `Vendor`) und in einer anderen Datei "`vendor`": "DELL" (mit einem kleinen „v” in `vendor`) angeben, gibt das System ein Fehler zurück.

> **Note**
>
> Sie müssen die Datei mit der Erweiterung `.json` speichern und die von Ihnen definierte Bestandsliste darf nur aus Zeichenfolgenwerten bestehen.

Wenn Sie die Datei erstellt haben, müssen Sie sie auf dem Knoten speichern. In der folgenden Tabelle wird der jeweilige Speicherort angezeigt, an dem die JSON-Dateien für den benutzerdefinierten Bestand auf dem Knoten gespeichert werden müssen.

Betriebssystem	Pfad
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

Ein Beispiel für die Verwendung des benutzerdefinierten Bestands finden Sie unter Abrufen der Laufwerkauslastung Ihrer Flotte mit benutzerdefinierten Bestandstypen in EC2 Systems Manager.

Löschen eines benutzerdefinierten Bestands

Sie können die DeleteInventory-API-Operation zum Löschen eines benutzerdefinierten Bestandstyp und der Daten verwenden, die diesem zugeordnet sind. Sie rufen den Befehl delete-inventory unter Verwendung der AWS Command Line Interface (AWS CLI) auf, um alle Daten für einen Bestandstyp zu löschen. Sie rufen den Befehl delete-inventory mit der `SchemaDeleteOption` auf, um einen benutzerdefinierten Bestandstyp zu löschen.

> **Note**
>
> Ein Bestandstyp wird auch als Bestandsschema bezeichnet.

Der Parameter `SchemaDeleteOption` umfasst die folgenden Optionen:

- DeleteSchema: Diese Option löscht den angegebenen benutzerdefinierten Typ und alle seine zugeordneten Daten. Sie können das Schema später bei Bedarf erneut erstellen.
- DisableSchema: Wenn Sie diese Option auswählen, deaktiviert das System die aktuelle Version, löscht alle Daten und ignoriert alle neuen Daten, wenn die Version kleiner oder gleich der deaktivierten Version ist. Um diesen Bestand für eine höhere als die deaktivierte Version erneut zu aktivieren, führen Sie die Aktion PutInventory aus.

So löschen oder deaktivieren Sie einen benutzerdefinierten Bestand mit der AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um mit der Option `dry-run` anzuzeigen, welche Daten aus dem System gelöscht werden. Mit diesem Befehl werden keine Daten gelöscht.

   ```
   aws ssm delete-inventory --type-name "Custom:custom_type_name" --dry-run
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
      "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"
}
```

Informationen zur Interpretation der Übersicht für gelöschten Bestand finden Sie unter Interpretieren der Übersicht für gelöschten Bestand.

3. Führen Sie den folgenden Befehl aus, um alle Daten für einen benutzerdefinierten Bestandstyp zu löschen.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name"
```

> **Note**
>
> Der Fortschritt des Löschvorgangs wird in der Ausgabe dieses Befehls nicht angezeigt. Daher sind TotalCount und RemainingCount immer identisch, da das System noch nichts gelöscht hat. Verwenden Sie den Befehl "describe-inventory-deletions", um den Fortschritt des Löschvorgangs anzuzeigen, wie später in diesem Thema beschrieben.

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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"
}
```

Das System löscht alle Daten für den angegebenen benutzerdefinierten Bestandstyp aus dem Systems Manager Inventory-Service.

4. Führen Sie den folgenden Befehl aus. Der Befehl führt die folgenden Aktionen für die aktuelle Version des Bestandstyps aus: Deaktivieren der aktuellen Version, Löschen aller Daten daraus und Ignorieren aller neuen Daten, wenn die Version kleiner oder gleich der deaktivierten Version ist.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name" --schema-delete-
option "DisableSchema"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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"
}
```

Sie können einen deaktivierten Bestandstyp mit dem folgenden Befehl anzeigen.

```
aws ssm get-inventory-schema --type-name Custom:custom_type_name
```

5. Führen Sie den folgenden Befehl aus, um einen Bestandstyp zu löschen.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name" --schema-delete-
option "DeleteSchema"
```

Das System löscht das Schema und alle Bestandsdaten für den angegebenen benutzerdefinierten Typ.

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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"
```

```
}
```

Anzeigen des Löschstatus

Sie können den Status eines Löschvorgangs mit dem `describe-inventory-deletions` AWS CLI-Befehl überprüfen. Sie können eine Lösch-ID angeben, um den Status eines bestimmten Löschvorgangs anzuzeigen. Wenn Sie keine Lösch-ID angeben, wird eine Liste aller Löschvorgänge der letzten 30 Tage angezeigt.

1. Führen Sie den folgenden Befehl aus, um den Status eines Löschvorgangs anzuzeigen. Das System gibt in der Übersicht über gelöschten Bestand die Lösch-ID zurück.

   ```
   aws ssm describe-inventory-deletions --deletion-id system_generated_deletion_ID
   ```

   Das System gibt den aktuellen Status zurück. Der Löschvorgang ist möglicherweise noch nicht abgeschlossen. Das System gibt unter anderem folgende Informationen zurück

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

   Wenn der Löschvorgang abgeschlossen ist, wird in `LastStatusMessage` die Meldung "Deletion is successful" (Löschvorgang erfolgreich) angezeigt.

```
{"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. Führen Sie den folgenden Befehl aus, um eine Liste aller Löschvorgänge der letzten 30 Tage anzuzeigen.

```
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"
```

Interpretieren der Übersicht für gelöschten Bestand

Sehen Sie sich das folgende Beispiel an, um den Inhalt der Übersicht für gelöschten Bestand besser zu verstehen. Ein Benutzer hat dem Bestand Custom:RackSpace drei Knoten zugewiesen. Die Bestandsartikel 1 und 2 verwenden die benutzerdefinierte Typversion 1.0 ("SchemaVersion":"1.0"). Bestandsartikel 3 verwendet die benutzerdefinierte Typversion 2.0 ("SchemaVersion":"2.0").

Benutzerdefinierter RackSpace-Bestand 1

```
{
   "CaptureTime":"2018-02-19T10:48:55Z",
   "TypeName":"CustomType:RackSpace",
   "InstanceId":"i-1234567890",
   "SchemaVersion":"1.0"   "Content":[
      {
         content of custom type omitted
      }
   ]
}
```

Benutzerdefinierter RackSpace-Bestand 2

```
{
   "CaptureTime":"2018-02-19T10:48:55Z",
   "TypeName":"CustomType:RackSpace",
   "InstanceId":"i-1234567891",
   "SchemaVersion":"1.0"   "Content":[
      {
         content of custom type omitted
      }
   ]
}
```

Benutzerdefinierter RackSpace-Bestand 3

```
{
   "CaptureTime":"2018-02-19T10:48:55Z",
   "TypeName":"CustomType:RackSpace",
   "InstanceId":"i-1234567892",
   "SchemaVersion":"2.0"   "Content":[
      {
         content of custom type omitted
      }
   ]
}
```

Der Benutzer führt den folgenden Befehl aus, um eine Vorschau der zu löschenden Daten anzuzeigen.

```
aws ssm delete-inventory --type-name "Custom:RackSpace" --dry-run
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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"
}
```

Der Benutzer führt den folgenden Befehl aus, um den Custom:RackSpace-Bestand zu löschen.

> **Note**
>
> Der Fortschritt des Löschvorgangs wird in der Ausgabe dieses Befehls nicht angezeigt. Daher sind `TotalCount` und `RemainingCount` immer identisch, da das System noch nichts gelöscht hat. Sie können den `describe-inventory-deletions`-Befehl verwenden, um den Fortschritt des Löschvorgangs anzuzeigen.

```
aws ssm delete-inventory --type-name "Custom:RackSpace"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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"
}
```

Anzeigen von Löschaktionen für einen Bestand in EventBridge

Sie können Amazon EventBridge so konfigurieren, dass es immer dann ein Ereignis erstellt, wenn ein Benutzer einen benutzerdefinierten Bestand löscht. EventBridge bietet drei Arten von Ereignissen für benutzerdefinierte Inventory-Löschoperationen:

- Löschaktion für eine Instance: Ob der benutzerdefinierte Bestand für einen bestimmten verwalteten Knoten erfolgreich gelöscht wurde oder nicht.
- Löschaktion-Übersicht: Eine Übersicht über die Löschaktion.
- Warnung für einen deaktivierten benutzerdefinierten Bestandstyp: Ein Warnereignis, wenn ein Benutzer die API-Operation PutInventory für eine benutzerdefinierte Bestandstypversion aufgerufen hat, die zuvor deaktiviert wurde.

Hier finden Sie Beispiele für jedes Ereignis.

Löschaktion für eine Instance


```
{
   "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"
   }
}
```


Löschaktion-Übersicht


```
{
   "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"
   }
}
```

Warnung für einen deaktivierten benutzerdefinierten Bestandstyp

```
{
   "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"
   }
}
```

Führen Sie das folgende Verfahren aus, um eine EventBridge-Regel für Löschvorgänge an benutzerdefiniertem Bestand zu erstellen. In diesem Verfahren wird gezeigt, wie Sie eine Regel erstellen, die Benachrichtigungen für Löschvorgänge an benutzerdefiniertem Bestand an ein Amazon SNS-Thema sendet. Bevor Sie beginnen, stellen Sie sicher, dass Sie ein Amazon SNS-Thema haben oder ein neues erstellen. Weitere Informationen dazu erhalten Sie unter Erste Schritte im Amazon Simple Notification Service Entwicklerhandbuch.

So konfigurieren Sie EventBridge für Bestandslöschvorgänge

1. Öffnen Sie die Amazon EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.

3. Wählen Sie Regel erstellen.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
5. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf übereinstimmende Ereignisse reagiert, die von Ihrem eigenen AWS-Konto stammen, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es stets an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus.
7. Wählen Sie Next (Weiter).
8. Wählen Sie für Event source (Ereignisquelle) AWS events or EventBridge partner events (-Ereignisse oder EventBridge-Partnerereignisse).
9. Wählen Sie im Abschnitt Ereignismuster die Option Ereignismusterformular aus.
10. Als Event source (Ereignisquelle) wählen Sie AWS-Services aus.
11. Wählen Sie für AWS service (-Service), die Option Systems Manager aus.
12. Wählen Sie Inventory für Event type (Ereignistyp).
13. Für Specific detail type(s) (Spezifische(r) Detail-Typ(en)), wählen Sie Inventory Resource State Change (Inventar-Ressourcen-Statusänderung).
14. Wählen Sie Next (Weiter).
15. Bei Target types (Zieltypen) wählen Sie AWS-Service aus.
16. Wählen Sie für Select a target (Ziel auswählen), die Option SNS topic (SNS-Thema), und dann für Topic (Thema), Ihr Thema aus.
17. Vergewissern Sie sich, dass im Abschnitt Additional settings (Zusätzliche Einstellungen) für Configure target input (Zieleingabe konfigurieren) die Option Matched event (Übereinstimmendes Ereignis) ausgewählt ist.
18. Wählen Sie Next (Weiter).
19. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Tagging Your Amazon EventBridge Resources (Taggen Ihrer Amazon EventBridge Resources) im Amazon EventBridge-Benutzerhandbuch.
20. Wählen Sie Next (Weiter).
21. Überprüfen Sie die Details der Regel und wählen Sie dann Create rule (Regel erstellen) aus.

Anzeigen von Bestandsverlauf und Änderungsnachverfolgung

Sie können den AWS Systems Manager-Inventory-Verlauf sowie die Änderungsnachverfolgung für alle Ihre verwalteten Knoten mit AWS Config anzeigen. AWS Config stellt eine detaillierte Ansicht der Konfiguration von AWS-Ressourcen in Ihrem AWS-Konto bereit. Dazu gehört auch, wie die Ressourcen jeweils zueinander in Beziehung stehen und wie sie in der Vergangenheit konfiguriert wurden, damit Sie sehen können, wie sich die Konfigurationen und Beziehungen im Laufe der Zeit verändern. Um den Bestandsverlauf und die Änderungsverfolgung anzuzeigen, müssen Sie die folgenden Ressourcen in AWS Config aktivieren:

- SSM:ManagedInstanceInventory
- SSM:PatchCompliance
- SSM:AssociationCompliance
- SSM:FileData

> **Note**
>
> Beachten Sie die folgenden wichtigen Hinweise zum Inventory-Verlauf und der Änderungsverfolgung:
>
> - Wenn Sie AWS Config verwenden, um Änderungen in Ihrem System zu verfolgen, müssen Sie Systems Manager Inventory so konfigurieren, dass `AWS:File`-Metadaten gesammelt werden, damit Sie Dateiänderungen in AWS Config(`SSM:FileData`) anzeigen können. Wenn Sie das nicht tun, dann verfolgt AWS Config keine Dateiänderungen auf Ihrem System.
> - Durch die Aktivierung von SSM:PatchCompliance und SSM:AssociationCompliance können Sie Systems Manager Patch Manager-Patches und den Systems Manager State Manager-Zuordnungs-Compliance-Verlauf sowie die Änderungsnachverfolgung anzeigen. Weitere Informationen über die Compliance-Verwaltung für diese Ressourcen finden Sie unter Arbeiten mit Compliance.

Im folgenden Verfahren wird beschrieben, wie Sie die Erfassung des Bestandsverlaufs und der Änderungsnachverfolgung in AWS Config mithilfe von AWS Command Line Interface (AWS CLI) aktivieren. Weitere Informationen dazu, wie Sie diese Ressourcen in AWS Config auswählen und konfigurieren, finden Sie unter Selecting Which Resources AWS Config Records im AWS Config-

Entwicklerleitfaden. Informationen zu AWS Config-Preisen erhalten Sie unter Pricing (Preise für WAF).

Bevor Sie beginnen

AWS Config erfordert AWS Identity and Access Management (IAM)-Berechtigungen, um Konfigurationsdetails zu Systems Manager-Ressourcen abzurufen. Im folgenden Verfahren müssen Sie einen Amazon-Ressourcennamen (ARN) für eine IAM-Rolle angeben, die AWS Config-Berechtigung für Systems Manager-Ressourcen gewährt. Sie können die verwaltete `AWS_ConfigRole`-Richtlinie der IAM-Rolle hinzufügen, die Sie AWS Config zuweisen. Weitere Informationen über diese Rolle finden Sie unter von AWS verwaltete Richtlinie: AWS_ConfigRole im AWS Config-Entwicklerhandbuch. Weitere Informationen zum Erstellen einer IAM-Rolle und dem Zuweisen der `AWS_ConfigRole`-verwalteten Richtlinie zu dieser Rolle finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

So aktivieren Sie die Erfassung des Bestandsverlaufs und der Änderungsnachverfolgung in AWS Config.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Kopieren Sie das folgende JSON-Beispiel in einen einfachen Texteditor und speichern Sie die Datei unter dem Namen recordingGroup.json.

   ```
   {
      "allSupported":false,
      "includeGlobalResourceTypes":false,
      "resourceTypes":[
         "AWS::SSM::AssociationCompliance",
         "AWS::SSM::PatchCompliance",
         "AWS::SSM::ManagedInstanceInventory",
         "AWS::SSM::FileData"
      ]
   }
   ```

3. Führen Sie den folgenden Befehl aus, um die Datei recordingGroup.json in AWS Config zu laden.

```
aws configservice put-configuration-recorder --configuration-recorder
 name=myRecorder,roleARN=arn:aws:iam::123456789012:role/myConfigRole --recording-
group file://recordingGroup.json
```

4. Führen Sie den folgenden Befehl aus, um die Erfassung des Bestandsverlaufs und der Änderungsnachverfolgung zu starten.

```
aws configservice start-configuration-recorder --configuration-recorder-
name myRecorder
```

Nachdem Sie die Verlaufs- und Änderungsverfolgung konfiguriert haben, können Sie weitere Details des Verlaufs für einen bestimmten verwalteten Knoten mit der Schaltfläche AWS Config in der Systems-Manager-Konsole anzeigen. Sie können entweder von der Seite Managed Instances (Verwaltete Instances) oder der Inventory(Inventory)-Seite aus auf die Schaltfläche AWS Config zugreifen. Je nach Bildschirmgröße müssen Sie möglicherweise einen Bildlauf nach rechts auf der Seite ausführen, um die Schaltfläche anzuzeigen.

Anhalten der Datenerfassung und Löschen von Bestandsdaten

Wenn Sie AWS Systems Manager Inventar nicht mehr verwenden möchten, um Metadaten zu Ihren AWS Ressourcen anzuzeigen, können Sie die Datenerfassung beenden und bereits gesammelte Daten löschen. Dieser Abschnitt enthält folgende Informationen.

Themen

- Beenden der Datensammlung
- Löschen einer Inventory Resource Data Sync

Beenden der Datensammlung

Wenn Sie Systems Manager zunächst so konfigurieren, dass Bestandsdaten erfasst werden, erstellt das System eine State Manager-Zuordnung, die den Zeitplan und die Ressourcen definiert, aus denen Metadaten gesammelt werden sollen. Sie können die Datenerfassung stoppen, indem Sie alle State Manager-Zuordnungen löschen, die das `AWS-GatherSoftwareInventory`-Dokument verwenden.

Löschen einer Bestandszuordnung

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie eine Zuordnung aus, die das `AWS-GatherSoftwareInventory`-Dokument nutzt und wählen Sie dann Delete (Löschen).
4. Wiederholen Sie Schritt drei für alle verbleibenden Zuordnungen, die das `AWS-GatherSoftwareInventory`-Dokument verwenden.

Löschen einer Inventory Resource Data Sync

Wenn Sie AWS Systems Manager Inventar nicht mehr verwenden möchten, um Metadaten zu Ihren AWS Ressourcen anzuzeigen, empfehlen wir außerdem, die für die Erfassung von Inventardaten verwendeten Ressourcendatensynchronisationen zu löschen.

Löschen einer Inventory Resource Data Sync

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie Resource Data Syncs (Ressourcen-Datensynchronisierung).
4. Wählen Sie eine Synchronisierung aus der Liste aus.

> **Important**
>
> Stellen Sie sicher, dass Sie die Synchronisierung für Inventory auswählen. Systems Manager unterstützt die Resource Data Sync für mehrere Funktionen. Wenn Sie die falsche Synchronisierung wählen, könnten Sie die Datenaggregation für Systems Manager Explorer oder Systems Manager Compliance unterbrechen.

5. Wählen Sie Delete (Löschen)
6. Wiederholen Sie diese Schritte für alle verbleibenden Resource Data Syncs, die Sie löschen möchten.

7. Löschen Sie den Amazon Simple Storage Service (Amazon S3)-Bucket, in dem die Daten gespeichert wurden. Weitere Informationen zum Löschen eines Amazon S3-Buckets finden Sie unter Deleting a bucket (Löschen eines Buckets).

Walkthroughs zu Systems Manager Inventory

Verwenden Sie die folgenden Walkthroughs für die Erfassung und Verwaltung von Bestandsdaten mit AWS Systems Manager Inventory. Wir empfehlen, dass Sie diese Walkthroughs zunächst für verwaltete Knoten in einer Testumgebung ausführen.

Bevor Sie beginnen

Bevor Sie mit diesen Walkthroughs beginnen, führen Sie die folgenden Aufgaben aus:

- Aktualisieren Sie AWS Systems Manager SSM Agent auf den Knoten, für die Sie eine Bestandsaufnahme ausführen möchten. Durch Ausführen der neuesten Version von SSM Agent stellen Sie sicher, dass Sie Metadaten für alle unterstützten Bestandstypen sammeln können. Informationen zur Aktualisierung von SSM Agent mithilfe von State Manager finden Sie unter Anleitung: Automatische Aktualisierung von SSM Agent (CLI).
- Überprüfen Sie, ob Sie die Einrichtungsanforderungen für Ihre Instances der Amazon Elastic Compute Cloud (Amazon EC2) und Nicht-EC2-Geräte in einer Hybrid- und Multi-Cloud-Umgebung erfüllt haben. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.
- (Optional) Erstellen Sie eine JSON-Datei für das Erfassen des benutzerdefinierten Bestands. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefiniertem Bestand.

Inhalt

- Walkthrough: Zuweisen benutzerdefinierter Bestands-Metadaten zu einem verwalteten Knoten
- Walkthrough: Konfigurieren Ihrer verwalteten Knoten für Inventory mithilfe der CLI
- Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten

Walkthrough: Zuweisen benutzerdefinierter Bestands-Metadaten zu einem verwalteten Knoten

Das folgende Verfahren führt Sie durch die Schritte zur Verwendung der PutInventory-API-Operation von AWS Systems Manager, mit der Sie einem verwalteten Knoten benutzerdefinierte Bestands-

Metadaten zuweisen können. In diesem Beispiel werden einem Knoten Informationen zum Rack-Standort zugewiesen. Weitere Informationen zum benutzerdefinierten Bestand finden Sie unter Arbeiten mit benutzerdefiniertem Bestand

So weisen Sie benutzerdefinierte Bestands-Metadaten zu einem verwalteten Knoten zu

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um einem Knoten Informationen zum Rack-Standort zuzuweisen.

   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. Führen Sie den folgenden Befehl aus, um die Einträge eines benutzerdefinierten Bestands für diesen Knoten anzuzeigen.

   ```
   aws ssm list-inventory-entries --instance-id ID --type-name "Custom:RackInfo"
   ```

   Das System gibt die folgenden Informationen zurück.

   ```
   {
       "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. Führen Sie den folgenden Befehl aus, um das benutzerdefinierte Bestandsschema anzuzeigen.

```
aws ssm get-inventory-schema --type-name Custom:RackInfo
```

Das System gibt die folgenden Informationen zurück.

```
{
    "Schemas": [
        {
            "TypeName": "Custom:RackInfo",
            "Version": "1.0",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "RackLocation"
                }
            ]
        }
    ]
}
```

Walkthrough: Konfigurieren Ihrer verwalteten Knoten für Inventory mithilfe der CLI

Die folgenden Verfahren führen Sie durch die Schritte zur Konfiguration von AWS Systems Manager Inventory für das Erfassen von Metadaten aus Ihren verwalteten Knoten. Wenn Sie die Erfassung durch Inventory konfigurieren, erstellen Sie zuerst eine Systems Manager State Manager-Zuordnung. Systems Manager erfasst die Bestandsdaten, wenn der Zuordnungsstatus ausgeführt wird. Wenn Sie den Zuordnungsstatus nicht zuerst erstellen und versuchen, das `aws:softwareInventory`-Plug-In z. B. mit Systems Manager Run Command aufzurufen, gibt das System den folgenden Fehler aus:

`The aws:softwareInventory plugin can only be invoked via ssm-associate.`

> **Note**
>
> Pro Knoten kann nur jeweils eine Bestandszuordnungs konfiguriert werden. Wenn Sie einen Knoten mit zwei oder mehr Bestandszuordnungen konfigurieren, wird die Zuordnung nicht ausgeführt und es werden keine Bestandsdaten erfasst.

Schnelle Konfiguration aller Ihrer verwalteten Knoten für Inventory (CLI)

Sie können schnell alle verwalteten Knoten in Ihrer AWS-Konto und in der aktuellen Region konfigurieren, um Inventardaten zu sammeln. Dieser Vorgang wird als „Erstellen einer globalen Bestandszuordnung“ bezeichnet. Um eine globale Bestandszuordnung mithilfe der AWS CLI zu erstellen, verwenden Sie die Platzhalteroption für den `instanceIds`-Wert, wie im folgenden Beispiel gezeigt:

So konfigurieren Sie das Inventar für alle verwalteten Knoten in Ihrer AWS-Konto und in der aktuellen Region (CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus.

   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**
>
> Mit diesem Befehl kann Inventory keine Metadaten für die Windows-Registrierung oder Dateien sammeln. Um diese Datentypen in den Bestand aufzunehmen, fahren Sie mit dem nächsten Schritt fort.

Manuelle Konfiguration von Inventory auf Ihren verwalteten Knoten (CLI)

Gehen Sie wie folgt vor, um AWS Systems Manager Inventar auf Ihren verwalteten Knoten mithilfe von Knoten-IDs oder Tags manuell zu konfigurieren.

So konfigurieren Sie Ihre verwalteten Knoten manuell für Inventory (CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um eine State Manager-Zuordnung zu erstellen, die Systems Manager Inventory auf dem Knoten ausführt. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Mit diesem Befehl wird der Service so konfiguriert, dass er alle sechs Stunden ausgeführt wird und Metadaten über Netzwerkkonfigurationen, Windows Update und Anwendungen aus einem Knoten erfasst.

   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\": \"DOC-EXAMPLE-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\": \"DOC-EXAMPLE-BUCKET\",
 \"OutputS3KeyPrefix\": \"Test\" } }" ^
--parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
```

Das System gibt die folgenden Informationen zurück.

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

Sie können dies für große Gruppen von Knoten durchführen, indem Sie den `Targets`-Parameter in Verbindung mit EC2-Tags verwenden. Sehen Sie sich das folgende -Beispiel an.

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\": \"DOC-EXAMPLE-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\": \"DOC-EXAMPLE-BUCKET\", \"OutputS3KeyPrefix\": \"Test
\" } }" ^
--parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
```

Sie können auch Dateien und Windows-Registry-Schlüssel auf einem Windows Server-Knoten inventarisieren, indem Sie die Bestandstypen `files` und `windowsRegistry` mit Ausdrücken

verwenden. Weitere Informationen zu diesen Bestandstypen finden Sie unter Arbeiten mit Datei- und Windows-Registrierungsbestand.

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. Führen Sie den folgenden Befehl aus, um den Zuordnungsstatus anzuzeigen.

```
aws ssm describe-instance-associations-status --instance-id an_instance_ID
```

Das System gibt die folgenden Informationen zurück.

```
{
"InstanceAssociationStatusInfos": [
         {
            "Status": "Pending",
            "DetailedStatus": "Associated",
            "Name": "reInvent2016PolicyDocumentTest",
            "InstanceId": "i-1a2b3c4d5e6f7g",
            "AssociationId": "1a2b3c4d5e6f7g-1a2b3c-1a2b3c-1a2b3c-1a2b3c4d5e6f7g",
            "DocumentVersion": "1"
        }
]
```

```
}
```

Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten

In der folgenden exemplarischen Vorgehensweise wird beschrieben, wie Sie mithilfe von AWS Command Line Interface (AWS CLI) eine Konfiguration für die AWS Systems Manager Ressourcendatensynchronisierung für Inventar erstellen. Eine Ressourcen-Datensynchronisierung portiert alle Bestandsdaten aus verwalteten Knoten automatisch in einen zentralen Amazon Simple Storage Service (Amazon S3)-Bucket. Die Synchronisierung aktualisiert die Daten in dem zentralen Amazon S3-Bucket automatisch, sobald neue Bestandsdaten erfasst werden.

In dieser exemplarischen Vorgehensweise wird auch beschrieben, wie Sie Amazon Athena und Amazon verwenden QuickSight , um die aggregierten Daten abzufragen und zu analysieren. Informationen zum Erstellen einer Ressourcendatensynchronisierung mithilfe von Systems Manager finden Sie unterKonfigurieren von Resource Data Sync für Inventory. AWS Management Console Informationen zum Abfragen von Inventar von mehreren AWS-Regionen Konten mithilfe von Systems Manager finden Sie AWS Management Console unterAbfragen von Bestandsdaten aus mehreren Regionen und Konten.

> **Note**
>
> Diese Anleitung enthält Informationen dazu, wie die Synchronisierung mit AWS Key Management Service (AWS KMS) verschlüsselt werden kann. Inventory erfasst keine personenbezogenen, geschützten oder vertraulichen Daten, d. h. die Verschlüsselung ist optional. Weitere Informationen zu AWS KMS finden Sie im AWS Key Management Service Entwicklerhandbuch.

Bevor Sie beginnen

Überprüfen oder erledigen Sie die folgenden Aufgaben, bevor Sie mit dem Walkthrough in diesem Abschnitt beginnen:

- Sammeln Sie Bestandsdaten von Ihren verwalteten Knoten. Für die Zwecke der QuickSight Abschnitte Amazon Athena und Amazon in dieser exemplarischen Vorgehensweise empfehlen wir Ihnen, Anwendungsdaten zu sammeln. Weitere Informationen zur Erfassung von Bestandsdaten

finden Sie unter Konfigurieren der Bestandserfassung oder Walkthrough: Konfigurieren Ihrer verwalteten Knoten für Inventory mithilfe der CLI.

- (Optional) Wenn die Inventardaten in einem Amazon Simple Storage Service (Amazon S3) - Bucket gespeichert werden, der die Verschlüsselung AWS Key Management Service (AWS KMS) verwendet, müssen Sie auch Ihr IAM-Konto und die `Amazon-GlueServiceRoleForSSM` Servicerolle für die AWS KMS Verschlüsselung konfigurieren. Wenn Sie Ihr IAM-Konto und diese Rolle nicht konfigurieren, wird Systems Manage `Cannot load Glue tables` anzeigen, wenn Sie die Registerkarte Detailed View (Detailansicht) in der Konsole wählen. Weitere Informationen finden Sie unter (Optional) Konfigurieren Sie Berechtigungen für die Anzeige AWS KMS verschlüsselter Daten.
- (Optional) Wenn Sie die Ressourcendatensynchronisierung mithilfe von verschlüsseln möchten AWS KMS, müssen Sie entweder einen neuen Schlüssel erstellen, der die folgende Richtlinie enthält, oder Sie müssen einen vorhandenen Schlüssel aktualisieren und diese Richtlinie hinzufügen.

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

```
}
```

So erstellen Sie eine Resource Data Sync für Inventory

1. Öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie einen Bucket zum Speichern der aggregierten Bestandsdaten. Weitere Informationen finden Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service. Notieren Sie sich den Namen des Buckets und den AWS-Region Ort, an dem Sie ihn erstellt haben.
3. Wenn Sie den Bucket erstellt haben, wählen Sie die Registerkarte Permissions aus und wählen Sie dann die Option Bucket Policy.
4. Kopieren Sie die folgende Bucket-Richtlinie in den Richtlinien-Editor. Ersetzen Sie DOC-EXAMPLE-BUCKET und *Account-ID* durch den Namen des Amazon S3 S3-Buckets, den Sie erstellt haben, und durch eine gültige ID. AWS-Konto Wenn Sie mehrere Konten hinzufügen, fügen Sie für jedes Konto eine zusätzliche Bedingungszeichenfolge und einen ARN hinzu. Entfernen Sie die zusätzlichen Platzhalter aus dem Beispiel, wenn Sie einzelne Konten hinzufügen. Sie können auch das *bucket-prefix* durch den Namen eines Amazon S3-Präfixes (Unterverzeichnis) ersetzen. Wenn Sie kein Präfix erstellt haben, entfernen Sie das *bucket-prefix/* aus dem ARN in der Richtlinie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": " SSMBucketDelivery",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-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:*:account-id1:resource-data-sync/*",
            "arn:aws:ssm:*:account-id2:resource-data-sync/*",
            "arn:aws:ssm:*:account-id3:resource-data-sync/*",
            "arn:aws:ssm:*:account-id4:resource-data-sync/*"
          ]
        }
      }
    }
  ]
}
```

5. (Optional) Wenn Sie die Synchronisierung verschlüsseln möchten, müssen Sie der im vorherigen Schritt aufgeführten Richtlinie die folgenden Bedingungen hinzufügen. Fügen Sie diese zum Abschnitt `StringEquals` hinzu.

```
"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"
```

   Ein Beispiel:

```
"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. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

7. (Optional) Wenn Sie die Synchronisation verschlüsseln möchten, führen Sie den folgenden Befehl aus, um zu überprüfen, ob die Bucket-Richtlinie die AWS KMS Schlüsselanforderung durchsetzt. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws s3 cp ./A_file_in_the_bucket s3://DOC-EXAMPLE-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://DOC-EXAMPLE-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. Führen Sie den folgenden Befehl aus, um eine Resource Data Sync-Konfiguration mit dem zu Beginn dieses Verfahrens erstellten Amazon S3-Bucket zu erstellen. Dieser Befehl erstellt eine Synchronisation aus dem, bei dem AWS-Region Sie angemeldet sind.

   > **Note**
   >
   > Wenn sich der Synchronisierungs- und der Amazon S3-Ziel-Bucket in verschiedenen Regionen befinden, können Kosten für Datenübertragung anfallen. Weitere Informationen finden Sie unter Amazon S3 – Preise.

   Linux & macOS

   ```
   aws ssm create-resource-data-sync \
   --sync-name a_name \
   --s3-destination "BucketName=DOC-EXAMPLE-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=DOC-EXAMPLE-BUCKET,Prefix=prefix_name,
 if_specified,SyncFormat=JsonSerDe,Region=bucket_region"
```

Sie können über den `region`-Parameter angeben, wo die Synchronisierungskonfiguration erstellt werden soll. Im folgenden Beispiel werden Bestandsdaten aus der Region us-west-1 in dem Amazon S3-Bucket in der Region us-west-2 synchronisiert.

Linux & macOS

```
aws ssm create-resource-data-sync \
    --sync-name InventoryDataWest \
    --s3-destination "BucketName=DOC-EXAMPLE-
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=DOC-EXAMPLE-
BUCKET,Prefix=HybridEnv,SyncFormat=JsonSerDe,Region=us-west-2" ^ --region us-
west-1
```

(Optional) Wenn Sie die Synchronisation mit verschlüsseln möchten, führen Sie den folgenden Befehl aus AWS KMS, um die Synchronisierung zu erstellen. Wenn Sie die Synchronisierung verschlüsseln, müssen sich der AWS KMS -Schlüssel und der Amazon S3-Bucket in derselben Region befinden.

Linux & macOS

```
aws ssm create-resource-data-sync \
--sync-name sync_name \
--s3-destination "BucketName=DOC-EXAMPLE-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=DOC-EXAMPLE-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. Führen Sie den folgenden Befehl aus, um den Status der Synchronisierungskonfiguration anzuzeigen.

   ```
   aws ssm list-resource-data-sync
   ```

   Wenn Sie die Synchronisierungskonfiguration in einer anderen Region erstellt haben, müssen Sie den `region`-Parameter angeben, wie im folgenden Beispiel gezeigt.

   ```
   aws ssm list-resource-data-sync --region us-west-1
   ```

10. Wenn die Synchronisierungskonfiguration erfolgreich erstellt wurde, prüfen Sie den Ziel-Bucket in Amazon S3. Die Bestandsdaten sollten in der Regel nach nur wenige Minuten angezeigt werden.

Arbeiten mit den Daten in Amazon Athena

Im folgenden Abschnitt wird beschrieben, wie Sie die Daten in Amazon Athena anzeigen und abfragen können. Bevor Sie beginnen, empfehlen wir Ihnen, sich mit Athena vertraut zu machen. Weitere Informationen finden Sie unter Was ist Amazon Athena? und Arbeiten mit Daten im Benutzerhandbuch für Amazon Athena.

Anzeigen und Abfragen von Daten in Amazon Athena

1. Öffnen Sie die Athena-Konsole unter https://console.aws.amazon.com/athena/.
2. Kopieren Sie die folgende Anweisung, fügen Sie sie in den Abfrage-Editor ein und wählen Sie dann Run Query.

   ```
   CREATE DATABASE ssminventory
   ```

   Das System erstellt eine Datenbank mit dem Namen ssminventory.

3. Kopieren Sie die folgende Anweisung, fügen Sie sie in den Abfrage-Editor ein und wählen Sie dann Run Query. Ersetzen Sie DOC-EXAMPLE-BUCKET und *bucket_prefix durch den Namen und das Präfix* des Amazon S3 S3-Ziels.

```
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://DOC-EXAMPLE-BUCKET/bucket_prefix/AWS:Application/'
```

4. Kopieren Sie die folgende Anweisung, fügen Sie sie in den Abfrage-Editor ein und wählen Sie dann Run Query.

```
MSCK REPAIR TABLE ssminventory.AWS_Application
```

   Das System partitioniert die Tabelle.

   > **Note**
   >
   > Wenn Sie Ressourcendatensynchronisationen aus zusätzlichen AWS-Regionen oder erstellen, müssen Sie diesen Befehl erneut ausführen AWS-Konten, um die Partitionen zu aktualisieren. Sie müssen möglicherweise auch Ihre Amazon S3-Bucket-Richtlinie aktualisieren.

5. Sie können Ihre Daten in der Vorschau anzeigen, indem Sie das Ansichtssymbol neben der Tabelle `AWS_Application` wählen.

6. Kopieren Sie die folgende Anweisung, fügen Sie sie in den Abfrage-Editor ein und wählen Sie dann 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
```

   Die Abfrage gibt die Anzahl der verschiedenen Versionen von zurück`aws-cfn-bootstrap`, wobei es sich um eine AWS Anwendung handelt, die auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances für Linux vorhanden istmacOS, undWindows Server.

7. Kopieren Sie die folgenden Anweisungen einzeln und fügen Sie sie in den Abfrage-Editor ein, ersetzen Sie DOC-EXAMPLE-BUCKET und ***Bucket-Präfix*** durch Informationen für Amazon S3 und wählen Sie dann Abfrage ausführen. Diese Anweisungen richten zusätzliche Bestandstabellen in Athena ein.

```
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://DOC-EXAMPLE-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://DOC-EXAMPLE-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://DOC-EXAMPLE-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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/bucket-prefix/AWS:PatchSummary/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_PatchSummary
```

Mit den Daten in Amazon arbeiten QuickSight

Der folgende Abschnitt bietet eine Übersicht mit Links zum Erstellen einer Visualisierung in Amazon QuickSight.

Um eine Visualisierung in Amazon zu erstellen QuickSight

1. Melden Sie sich bei Amazon an QuickSight und melden Sie sich dann an der QuickSight Konsole an.
2. Erstellen Sie einen Datensatz aus der Tabelle `AWS_Application` sowie aus allen anderen Tabellen, die Sie erstellt haben. Weitere Informationen finden Sie unter Erstellen eines Datasets mit Amazon Athena-Daten.
3. Verknüpfen Sie Tabellen. Sie können z. B. die Spalte `instanceid` aus `AWS_InstanceInformation` verknüpfen, da sie der Spalte `resourceid` in anderen Bestandstabellen entspricht. Weitere Informationen zum Verknüpfen von Tabellen finden Sie unter Verknüpfen von Tabellen.
4. Erstellen Sie eine Visualisierung. Weitere Informationen finden Sie unter Arbeiten mit Amazon QuickSight Visuals.

Fehlerbehebung bei Problemen mit Systems Manager Inventory

Dieses Thema enthält Informationen zum Beheben gängiger Fehler oder Probleme mit AWS Systems Manager Inventory. Informationen zur Behebung von Problemen bei der Anzeige Ihrer Knoten in Systems Manager finden Sie unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten.

Themen

- Mehrere Anwenden aller Zuordnungen mit Dokument 'AWS-GatherSoftwareInventory' werden nicht unterstützt
- Der Inventory-Ausführungsstatus verlässt nie den Status „ausstehend“.
- Das AWS-ListWindowsInventory-Dokument kann nicht ausgeführt werden
- Konsole zeigt das Inventory-Dashboard nicht an | Detailed View (Detailansicht) | Registerkarte „Settings“ (Einstellungen)
- UnsupportedAgent
- Übersprungen
- Fehlgeschlagen
- Inventark-Compliance für eine Amazon-EC2-Instance fehlgeschlagen
- S3-Bucket-Objekt enthält alte Daten

Mehrere Anwenden aller Zuordnungen mit Dokument '**AWS-GatherSoftwareInventory**' werden nicht unterstützt

Ein Fehler `Multiple apply all associations with document 'AWS-GatherSoftwareInventory' are not supported`, der bedeutet, dass eine oder mehrere AWS-Regionen, in denen Sie versuchen, eine Bestandszuordnung für alle Knoten zu konfigurieren, sind bereits mit einer Bestandszuordnung für alle Knoten konfiguriert. Falls erforderlich, können Sie die vorhandene Bestandszuordnung für alle Knoten löschen und anschließend eine neue erstellen. Um vorhandene Bestandszuordnungen anzuzeigen, wählen Sie State Manager in der Systems Manager Konsole und suchen Sie dann nach Zuordnungen, die das `AWS-GatherSoftwareInventory`-SSM-Dokument verwenden. Wenn die vorhandene Bestandszuordnung für alle Knoten in mehreren Regionen erstellt wurde und Sie eine neue erstellen möchten, müssen Sie die vorhandene Zuordnung aus jeder Region löschen, in der sie vorhanden ist.

Der Inventory-Ausführungsstatus verlässt nie den Status „ausstehend“.

Es gibt zwei Gründe, warum die Bestandsammlung niemals den `Pending` Status annimmt:

- Keine Knoten in der ausgewählten AWS-Region:

  Wenn Sie eine globale Bestandszuordnung mithilfe von Systems Manager Quick Setup erstellen, zeigt der Status der Bestandszuordnung (`AWS-GatherSoftwareInventory`-Dokument) `Pending` an, wenn in der ausgewählten Region keine Knoten verfügbar sind.
- Unzureichende Berechtigungen:

  Eine Bestandszuordnung zeigt `Pending` an, wenn eine oder mehrere Knoten nicht über die Berechtigung zum Ausführen von Systems Manager Inventory verfügen. Stellen Sie sicher, dass das AWS Identity and Access Management (IAM)-Instance-Profil die verwaltete Richtlinie AmazonSSMManagedInstanceCore enthält. Weitere Informationen zum Hinzufügen dieser Richtlinie zu einem Instance-Profil finden Sie unter Alternative Konfiguration für EC2-Instanzberechtigungen.

  Das Instance-Profil muss mindestens über die folgenden IAM-Berechtigungen verfügen.

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

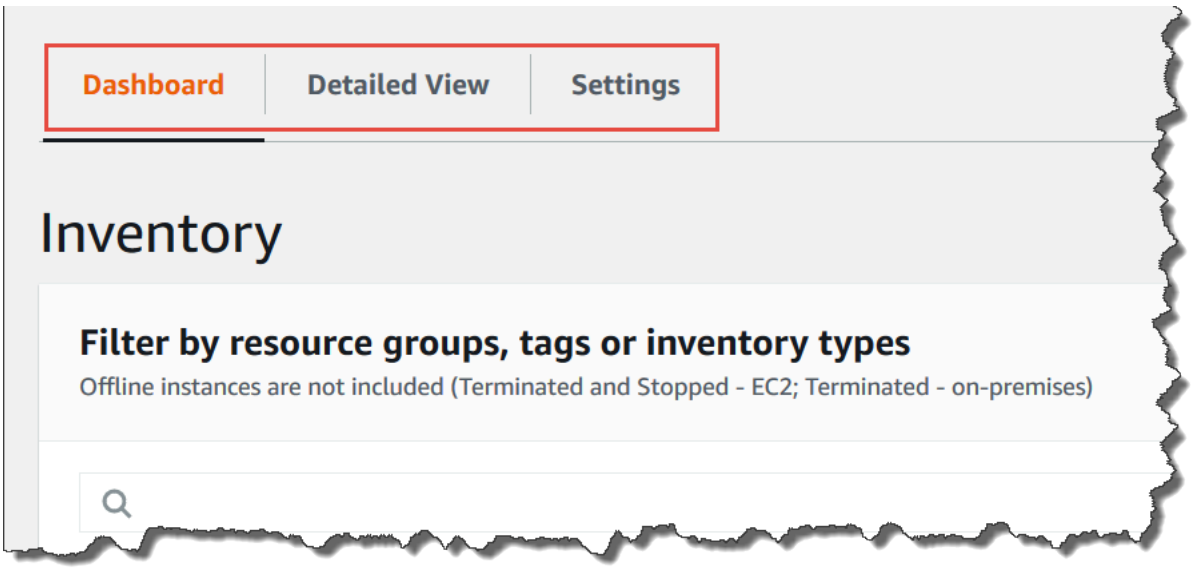
```
            "Action": [
                "ssm:DescribeAssociation",
                "ssm:ListAssociations",
                "ssm:ListInstanceAssociations",
                "ssm:PutInventory",
                "ssm:PutComplianceItems",
                "ssm:UpdateAssociationStatus",
                "ssm:UpdateInstanceAssociationStatus",
                "ssm:UpdateInstanceInformation",
                "ssm:GetDocument",
                "ssm:DescribeDocument"
            ],
            "Resource": "*"
        }
    ]
}
```

Das **AWS-ListWindowsInventory**-Dokument kann nicht ausgeführt werden

Das `AWS-ListWindowsInventory`-Dokument ist veraltet. Verwenden Sie dieses Dokument nicht zur Bestandserfassung. Verwenden Sie stattdessen einen der unter Konfigurieren der Bestandserfassung beschriebenen Prozesse.

Konsole zeigt das Inventory-Dashboard nicht an | Detailed View (Detailansicht) | Registerkarte „Settings“ (Einstellungen)

Die Seite Inventory Detailed View (Detailansicht) ist nur in AWS-Regionen verfügbar, die Amazon Athena anbieten. Wenn die folgenden Registerkarten nicht auf der Seite Systems Manager Inventory angezeigt werden, bedeutet dies, dass Athena nicht in der Region verfügbar ist und Sie die Detailansicht nicht verwenden können, um Daten abzufragen.



UnsupportedAgent

Wenn der detaillierte Status einer Bestandszuordnung UnsupportedAgent (Nicht unterstützter Agent) und der Association status (Zuordnungsstatus) Failed (Fehlgeschlagen) anzeigt, ist die Version von AWS Systems Manager SSM Agent auf dem verwalteten Knoten nicht korrekt. Um eine globale Bestandszuordnung (zur Bestandsaufnahme aller Knoten in Ihrem AWS-Konto) zu erstellen, müssen Sie beispielsweise SSM Agent Version 2.0.790.0 oder höher verwenden. Sie können die Ausführung der Agenten-Version auf jedem Ihrer Knoten auf der Seite Managed Instances (Verwaltete Instances) in der Spalte Agent version (Agent-Version) anzeigen. Weitere Informationen zur Aktualisierung von SSM Agent auf Ihren Knoten finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

Übersprungen

Wenn der Status der Bestandszuordnung für einen Knoten Skipped (Übersprungen) anzeigt, bedeutet dies, dass Sie eine globale Bestandszuordnung (Zum Sammeln Bestand von allen Knoten) erstellt haben, der übersprungene Knoten jedoch bereits über eine ihm zugewiesene Bestandszuordnung verfügte. Die globale Bestandszuordnung wurde diesem Knoten nicht zugewiesen und es wurde kein Bestand durch die globale Bestandszuordnung erfasst. Allerdings meldet der Knoten nach wie vor Bestandsdaten, wenn die spezifische Bestandszuordnung ausgeführt wird.

Wenn Sie nicht möchten, dass der Knoten von der globalen Bestandszuordnung übersprungen wird, müssen Sie die vorhandene Bestandszuordnung löschen. Um vorhandene Bestandszuordnungen

anzuzeigen, wählen Sie State Manager in der Systems Manager Konsole und suchen Sie dann nach Zuordnungen, die das `AWS-GatherSoftwareInventory`-SSM-Dokument verwenden.

Fehlgeschlagen

Wenn der Status der Bestandszuordnung für einen Knoten Failed (Fehlgeschlagen) anzeigt, könnte dies bedeuten, dass der Knoten über mehrere ihm zugewiesene Bestandszuordnungen verfügt. Ein Knoten kann jeweils nur über eine zugewiesene Bestandszuordnung verfügen. Eine Bestandszuordnung verwendet das `AWS-GatherSoftwareInventory` AWS Systems Manager-Dokument (SSM-Dokument). Sie können den folgenden Befehl ausführen, indem Sie die AWS Command Line Interface (AWS CLI) verwenden, um eine Liste der Zuordnungen für einen Knoten anzuzeigen.

```
aws ssm describe-instance-associations-status
                --instance-id instance ID
```

Inventark-Compliance für eine Amazon-EC2-Instance fehlgeschlagen

Die Inventar-Compliance für eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance kann fehlschlagen, wenn Sie der Instance mehrere Inventarzuordnungen zuweisen.

Um dieses Problem zu beheben, löschen Sie eine oder mehrere Inventarzuordnungen, die der Instance zugewiesen sind. Weitere Informationen finden Sie unter Löschen einer Zuordnung.

> **Note**
>
> Beachten Sie das folgende Verhalten, wenn Sie mehrere Bestandszuordnungen für einen verwalteten Knoten erstellen:
>
> - Jedem Knoten kann eine Bestandszuordnung zugewiesen werden, die auf alle Knoten abzielt (--targets "Key=InstanceIds,Values=*").
> - Jedem Knoten kann auch eine bestimmte Zuordnung zugewiesen werden, die entweder Tag-Schlüssel-Wert-Paare oder eine AWS-Ressourcengruppe verwendet.
> - Wenn einem Knoten mehrere Bestandszuordnungen zugewiesen sind, zeigt der Status Skipped (Übersprungen) für die Zuordnung an, die nicht ausgeführt wurde. Die zuletzt durchgeführte Zuordnung zeigt den aktuellen Status der Bestandszuordnung an.
> - Wenn einem Knoten mehrere Bestandszuordnungen zugewiesen sind und jede ein Tag-Schlüssel-Wert-Paar verwendet, können diese Bestandszuordnungen aufgrund des Tag-

Konflikts nicht auf dem Knoten ausgeführt werden. Die Zuordnung wird weiterhin auf Knoten ausgeführt, bei denen der Tag-Schlüssel-Wert-Konflikt nicht besteht.

S3-Bucket-Objekt enthält alte Daten

Die Daten im Amazon-S3-Bucket-Objekt werden aktualisiert, wenn die Zuordnung zum Bestand erfolgreich ist und neue Daten entdeckt werden. Das Amazon-S3-Bucket-Objekt wird für jeden Knoten aktualisiert, wenn die Zuordnung läuft und fehlschlägt, aber die Daten innerhalb des Objekts werden in diesem Fall nicht aktualisiert. Die Daten im Amazon-S3-Bucket-Objekt werden nur dann aktualisiert, wenn die Zuordnung erfolgreich verläuft. Wenn die Bestandszuordnung fehlschlägt, sehen Sie alte Daten in dem Amazon-S3-Bucket-Objekt.

AWS Systems Manager Hybride Aktivierungen

Um Nicht-EC2-Computer für die Verwendung AWS Systems Manager in einer Hybrid- und Multicloud-Umgebung zu konfigurieren, erstellen Sie eine Hybrid-Aktivierung. Zu den Nicht-EC2-Maschinentypen, die als verwaltete Knoten unterstützt werden, gehören die folgenden:

- Server in Ihren eigenen Räumlichkeiten (On-Premises-Server)
- AWS IoT Greengrass Kerngeräte
- AWS IoT und Geräte, die nicht zu den AWS Edge-Geräten gehören
- Virtuelle Maschinen (VMs), einschließlich VMs in anderen Cloud-Umgebungen

Wenn Sie den Befehl create-activation ausführen, um einen Hybrid-Aktivierungsprozess zu starten, erhalten Sie in der Befehlsantwort einen Aktivierungscode und eine ID. Anschließend fügen Sie den Aktivierungscode und die ID dem Befehl zur Installation von SSM Agent auf der Maschine bei, wie in Schritt 3 von Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen beschrieben. Dieser Aktivierungsprozess gilt für alle Maschinentypen, die nicht zu EC2 gehören, mit Ausnahme von AWS IoT Greengrass Kerngeräten. Informationen zur Konfiguration von AWS IoT Greengrass Kerngeräten für Systems Manager finden Sie unterVerwaltung von Edge-Geräten mit Systems Manager.

> **Note**
>
> Für Nicht-EC2-macOS-Maschinen wird derzeit kein Support bereitgestellt.

Üner Systems Manager Instances-Kontingente

AWS Systems Manager bietet eine Stufe „Standard-Instances“ und eine Stufe „Advanced-Instances“. Beide unterstützen verwaltete Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung. Die Stufe „Standard-Instances“ ermöglicht es Ihnen, maximal 1.000 Maschinen pro Person zu registrieren. AWS-Konto AWS-Region Wenn Sie mehr als 1 000 Maschinen in einem einzigen Konto und einer Region anmelden müssen, verwenden Sie das Advanced-Instances-Kontingent. Sie können im Advanced-Instances-Kontingent so viele verwaltete Knoten erstellen, wie Sie möchten. Alle verwalteten Knoten, die für Systems Manager konfiguriert sind, werden auf pay-per-use Basis von Preisen berechnet. Weitere Informationen über das Aktivieren des Advanced-Instances-Kontingent finden Sie unter Aktivieren des Kontingents für erweiterte Instances. Weitere Informationen über die Preise finden Sie unter AWS Systems Manager – Preise.

> **Note**
>
> - Mithilfe von Advanced Instances können Sie in einer Hybrid- und Multi-Cloud-Umgebung auch eine Verbindung zu Ihren Nicht-EC2-Knoten herstellen. AWS Systems Manager Session Manager Session Managerbietet interaktiven Shell-Zugriff auf Ihre Instances. Weitere Informationen finden Sie unter AWS Systems Manager Session Manager.
> - Das standardmäßige Instance-Kontingent gilt auch für EC2-Instances, die eine On-Premises-Aktivierung von Systems Manager verwenden (was kein übliches Szenario ist).
> - Aktivieren Sie das Advanced-Instances-Kontingent, um Anwendungen, die von Microsoft auf virtuellen Maschinen (VMs) On-Premises-Instances veröffentlicht werden, zu patchen. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Für Patch-Anwendungen, die von Microsoft auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances veröffentlicht wurden, fallen keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden.

AWS Systems Manager Session Manager

Session Managerist eine vollständig verwaltete Funktion. AWS Systems Manager Mit Session Manager können Sie Ihre Amazon Elastic Compute Cloud (Amazon EC2)-Instances, Edge-Geräte, On-Premises-Server und virtuelle Maschinen (VM) verwalten. Sie können entweder eine interaktive browserbasierte Shell mit einem Klick oder die AWS Command Line Interface ()AWS CLI verwenden. Session Managerbietet sicheres und überprüfbares Knotenmanagement, ohne

dass eingehende Ports geöffnet, Bastion-Hosts verwaltet oder SSH-Schlüssel verwaltet werden müssen. Session Managerermöglicht Ihnen außerdem die Einhaltung von Unternehmensrichtlinien, die einen kontrollierten Zugriff auf verwaltete Knoten, strenge Sicherheitspraktiken und vollständig überprüfbare Protokolle mit Knotenzugriffsdetails vorschreiben, und bietet Endbenutzern gleichzeitig einen einfachen plattformübergreifenden Zugriff mit nur einem Klick auf Ihre verwalteten Knoten. Um mit Session Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Session Manager aus.

Welche Vorteile bietet Session Manager meiner Organisation?

Session Manager bietet die folgenden Vorteile:

- Zentralisierte Kontrolle des Zugriffs auf verwaltete Knoten mit IAM-Richtlinien

  Administratoren erhalten eine zentrale Stelle zum Erteilen und Widerrufen des Zugriffs auf verwaltete Knoten. Wenn Sie ausschließlich AWS Identity and Access Management (IAM-) Richtlinien verwenden, können Sie steuern, welche einzelnen Benutzer oder Gruppen in Ihrem Unternehmen Zugriff darauf haben Session Manager und auf welche verwalteten Knoten sie zugreifen können.
- Keine offenen Ports für eingehenden Datenverkehr und keine Notwendigkeit für die Verwaltung von Bastion-Hosts oder SSH-Ports

  Wenn Sie SSH-Ports für eingehenden Datenverkehr und PowerShell-Remote Ports auf Ihren verwalteten Knoten geöffnet lassen, besteht ein höheres Risiko, dass juristische Stellen nicht autorisierte oder böswillige Befehle auf den verwalteten Knoten ausführen. Session Manager unterstützt Sie bei der Verbesserung des Sicherheitsstatus, da Sie diese Ports für eingehenden Datenverkehr schließen können. Dies befreit Sie von der Notwendigkeit, SSH-Schlüssel und -Zertifikate, Bastion-Hosts und Jumpboxes verwalten zu müssen.
- One-Click-Zugriff auf verwaltete Knoten über die Konsole und die CLI

  Mit der AWS Systems Manager Konsole oder der Amazon EC2 EC2-Konsole können Sie eine Sitzung mit einem einzigen Klick starten. Mit dem AWS CLI können Sie auch eine Sitzung starten, die einen einzelnen Befehl oder eine Befehlsfolge ausführt. Da Berechtigungen für verwaltete Knoten durch IAM-Richtlinien und nicht von SSH-Schlüsseln oder anderen Mechanismen bereitgestellt werden, wird die Verbindungszeit stark reduziert.
- Herstellen einer Verbindung mit Amazon-EC2-Instances und Nicht-EC2-verwalteten Knoten in Hybrid- und Multi-Cloud-Umgebungen

Sie können sich sowohl mit Instances der Amazon Elastic Compute Cloud (Amazon EC2) als auch mit Nicht-EC2-Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung verbinden.

Um über Session Manager eine Verbindung zu Nicht-EC2-Knoten herzustellen, müssen Sie zunächst die Advanced-Instances-Kontingente aktivieren. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Für die Verbindung mit EC2-Instances über Session Manager fallen jedoch keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter Konfigurieren von Instance-Kontingenten.

- Port-Weiterleitung

  Leiten Sie jeden Port in Ihrem verwalteten Knoten an einen lokalen Port auf einem Client um. Stellen Sie danach eine Verbindung mit dem lokalen Port her und greifen Sie auf die Serveranwendung zu, die in dem Knoten ausgeführt wird.

- Plattformübergreifende Unterstützung für Windows, Linux und macOS

  Session Manager unterstützt in einem einzigen Tool sowohl Windows, Linux als auch macOS. Sie müssen beispielsweise keinen SSH-Client für Linux- und macOS-verwaltete Knoten oder eine RDP-Verbindung für Windows Server-verwaltete Knoten verwenden.

- Protokollierung und Prüfung von Sitzungsaktivitäten

  Um betriebs- oder sicherheitsbezogene Anforderungen in Ihrer Organisation zu erfüllen, müssen Sie möglicherweise eine Aufzeichnung der Verbindungen bereitstellen, die mit Ihren verwalteten Knoten hergestellt wurden, und der Befehle, die auf ihnen ausgeführt wurden. Sie können auch Benachrichtigungen empfangen, wenn ein Benutzer in Ihrer Organisation Sitzungsaktivitäten startet oder beendet.

  Die Funktionen für Protokollierung und Prüfung werden durch die Integration mit den folgenden AWS-Services bereitgestellt:

  - AWS CloudTrail— AWS CloudTrail erfasst Informationen über Session Manager API-Aufrufe in Ihrem AWS-Konto und schreibt sie in Protokolldateien, die in einem von Ihnen angegebenen Amazon Simple Storage Service (Amazon S3) -Bucket gespeichert werden. Ein Bucket wird für alle CloudTrail Protokolle Ihres Kontos verwendet. Weitere Informationen finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.
  - Amazon Simple Storage Service – Sie können Sitzungsprotokolldaten zu Debugging-Zwecken in einem Amazon S3-Bucket Ihrer Wahl speichern. Protokolldaten können mit oder ohne Verschlüsselung über Ihren AWS KMS key an Ihren Amazon S3-Bucket gesendet werden.

Weitere Informationen finden Sie unter Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole).

- Amazon CloudWatch Logs — CloudWatch Logs ermöglicht es Ihnen, Protokolldateien aus verschiedenen Quellen zu überwachen, zu speichern und darauf zuzugreifen AWS-Services. Sie können Sitzungsprotokolldaten zu Debugging- und Fehlerbehebungszwecken an eine CloudWatch Logs-Protokollgruppe senden. Protokolldaten können mit oder ohne AWS KMS Verschlüsselung mit Ihrem KMS-Schlüssel an Ihre Protokollgruppe gesendet werden. Weitere Informationen finden Sie unter Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole).
- Amazon EventBridge und Amazon Simple Notification Service — EventBridge ermöglicht es Ihnen, Regeln einzurichten, um zu erkennen, wann Änderungen an den von Ihnen angegebenen AWS Ressourcen vorgenommen werden. Sie können eine Regel erstellen, um zu erkennen, wenn ein Benutzer in Ihrer Organisation eine Sitzung startet oder anhält. Anschließend können Sie über Amazon SNS eine Benachrichtigung (z. B. eine Text- oder E-Mail-Nachricht) über das Ereignis erhalten. Sie können ein CloudWatch Ereignis auch so konfigurieren, dass es andere Antworten auslöst. Weitere Informationen finden Sie unter Überwachung der Sitzungsaktivität mit Amazon EventBridge (Konsole).

> **Note**
>
> Protokollieren ist für Session Manager-Sitzungen, die eine Verbindung über Port-Weiterleitung oder SSH herstellen, nicht verfügbar. Dies liegt daran, dass SSH alle Sitzungsdaten verschlüsselt und Session Managernur als Tunnel für SSH-Verbindungen dient.

An wen richtet sich Session Manager?

- Jeder AWS Kunde, der seine Sicherheits- und Auditsituation verbessern, den betrieblichen Aufwand durch die Zentralisierung der Zugriffskontrolle auf verwalteten Knoten reduzieren und den eingehenden Knotenzugriff reduzieren möchte.
- Datensicherheitsexperten, die den Zugriff auf und die Aktivität von verwalteten Knoten überwachen und verfolgen möchten, Ports für eingehenden Datenverkehr auf verwalteten Knoten schließen möchten oder Verbindungen mit verwalteten Knoten ohne öffentliche IP-Adresse ermöglichen möchten.

- Administratoren, die den Zugriff über eine zentrale Stelle gewähren und widerrufen möchten und Benutzern eine einzige Lösung für von Linux, macOS und Windows Server verwaltete Knoten bereitstellen möchten.
- Benutzer, die mit nur einem Klick im Browser oder AWS CLI ohne Angabe von SSH-Schlüsseln eine Verbindung zu einem verwalteten Knoten herstellen möchten.

Was sind die Hauptfeatures von Session Manager?

- Support für von Windows Server-, Linux- und macOS- verwaltete Knoten

  Mit Session Manager können Sie sichere Verbindungen zu Amazon Elastic Compute Cloud (EC2)-Instances, Edge-Geräten, On-Premises-Servern und virtuellen Maschinen (VM) herstellen. Eine Liste der unter den jeweiligen Betriebssystemen unterstützten Typen finden Sie unter Einrichten von Session Manager.

  > **Note**
  >
  > Session Manager-Support für On-Premises-Server wird nur für das Advanced-Instances-Kontingent bereitgestellt. Weitere Informationen finden Sie unter Aktivieren des Kontingents für erweiterte Instances.

- Zugriff auf Session Manager-Funktionen über Konsole, CLI und SDK

  Sie können Session Manager wie folgt nutzen:

  Die AWS Systems Manager -Konsole bietet sowohl Administratoren als auch Endbenutzern Zugriff auf alle Session Manager-Funktionen. Sie können über die Systems Manager-Konsole jede Aufgabe im Zusammenhang mit Ihren Sitzungen ausführen.

  Die Amazon-EC2-Konsole bietet Endbenutzern die Möglichkeit, eine Verbindung zu den EC2-Instances herzustellen, für die sie Sitzungsberechtigungen erhalten haben.

  Die AWS CLI beinhaltet den Zugriff auf Session Manager-Funktionen für Endbenutzer. Sie können eine Sitzung starten, eine Liste von Sitzungen anzeigen und eine Sitzung dauerhaft beenden, indem Sie die AWS CLI verwenden.

> **Note**
>
> Um die Befehle AWS CLI to run session verwenden zu können, müssen Sie Version 1.16.12 der CLI (oder höher) verwenden und das Session Manager Plugin auf Ihrem lokalen Computer installiert haben. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI. Informationen zum Anzeigen des Plug-ins finden Sie unter GitHub session-manager-plugin.

- IAM-Zugriffskontrolle

  Mithilfe von IAM-Richtlinien können Sie steuern, welche Mitglieder Ihrer Organisation Sitzungen mit verwalteten Knoten starten können und auf welche Knoten sie zugreifen können. Sie können auch einen temporären Zugriff auf Ihre verwalteten Knoten bereitstellen. Beispielsweise könnten Sie einem Techniker auf Aufruf (oder einer Gruppe von Technikern auf Abruf) nur für die Dauer ihrer Schicht Zugriff auf Produktionsserver geben.

- Support für Protokollierungs- und Prüfungsfunktionen

  Session Managerbietet Ihnen Optionen für die Prüfung und Protokollierung von Sitzungsverläufen in Ihrer AWS-Konto durch die Integration mit einer Reihe von anderen. AWS-Services Weitere Informationen finden Sie unter Prüfen von Sitzungsaktivitäten und Protokollierung von Sitzungsaktivitäten aktivieren und deaktivieren.

- Konfigurierbare Shell-Profile

  Session Manager bietet Ihnen Optionen zum Konfigurieren von Voreinstellungen innerhalb von Sitzungen. Mit diesen anpassbaren Profilen können Sie Voreinstellungen wie Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnisse und das Ausführen mehrerer Befehle definieren, wenn eine Sitzung gestartet wird.

- Support für die Datenverschlüsselung mit dem Kundenschlüssel

  Sie können so konfigurierenSession Manager, dass die Sitzungsdatenprotokolle, die Sie an einen Amazon Simple Storage Service (Amazon S3) -Bucket senden oder in eine CloudWatch Logs-Protokollgruppe streamen, verschlüsselt werden. Sie können Session Manager auch so konfigurieren, dass die Daten, die zwischen Client-Maschinen und Ihren verwalteten Knoten während der Sitzungen übertragen werden, weiter verschlüsselt werden. Weitere Informationen finden Sie unter Protokollierung von Sitzungsaktivitäten aktivieren und deaktivieren und Konfigurieren von Sitzungspräferenzen.

- AWS PrivateLink Unterstützung für verwaltete Knoten ohne öffentliche IP-Adressen

  Sie können auch VPC-Endpunkte für Systems Manager einrichten, um Ihre Sitzungen weiter AWS PrivateLink zu sichern. AWS PrivateLink begrenzt den gesamten Netzwerkverkehr zwischen Ihren verwalteten Knoten, Systems Manager und Amazon EC2 auf das Amazon-Netzwerk. Weitere Informationen finden Sie unter Verbessern der Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

- Tunneling

  Verwenden Sie in einer Sitzung ein Dokument vom Typ Sitzung AWS Systems Manager (SSM), um Datenverkehr wie HTTP oder ein benutzerdefiniertes Protokoll zwischen einem lokalen Port auf einem Client-Computer und einem Remote-Port auf einem verwalteten Knoten zu tunneln.

- Interaktive Befehle

  Erstellen Sie ein SSM-Dokument vom Typ Session, das eine Sitzung verwendet, um interaktiv einen einzelnen Befehl auszuführen, sodass Sie verwalten können, was Benutzer auf einem verwalteten Knoten tun können.

Was ist eine Sitzung?

Eine Sitzung ist eine Verbindung zu einem verwalteten Knoten mit Session Manager. Sitzungen basieren auf einem sicheren bidirektionalen Kommunikationskanal zwischen dem Client (Sie) und dem remote verwalteten Knoten, der Eingaben und Ausgaben für Befehle streamt. Der Datenverkehr zwischen einem Client und einem verwalteten Knoten wird mit TLS 1.2 verschlüsselt. Anforderungen zum Aufbau der Verbindung werden mit Sigv4 signiert. Diese bidirektionale Kommunikation ermöglicht interaktive Bash und PowerShell den Zugriff auf verwaltete Knoten. Sie können auch einen AWS Key Management Service (AWS KMS) verwenden, um Daten über die standardmäßige TLS-Verschlüsselung hinaus zu verschlüsseln.

Angenommen, John ist ein Techniker auf Abruf in Ihrer IT-Abteilung. Er erhält eine Benachrichtigung zu einem Problem, für dessen Bearbeitung er eine Remote-Verbindung mit einem verwalteten Knoten herstellen muss, beispielsweise einem Ausfall, der behoben werden muss, oder eine Anweisung, eine einfache Konfigurationsoption für einen Knoten zu ändern. Mithilfe der AWS Systems Manager Konsole, der Amazon EC2 EC2-Konsole oder der startet John eine Sitzung AWS CLI, die ihn mit dem verwalteten Knoten verbindet, führt Befehle auf dem Knoten aus, die für die Ausführung der Aufgabe erforderlich sind, und beendet dann die Sitzung.

Wenn John den Befehl zum Starten der Sitzung sendet, authentifiziert der Session Manager-Service seine ID, überprüft die ihm von einer IAM-Richtlinie gewährten Berechtigungen, prüft Konfigurationseinstellungen (z. B. die zulässigen Limits für die Sitzungen) und sendet eine Nachricht an, SSM Agent, um die Zwei-Wege-Verbindung zu öffnen. Nach der Herstellung der Verbindung und der Eingabe des nächsten Befehls durch John wird die Befehlsausgabe aus SSM Agent zu diesem Kommunikationskanal hochgeladen und zurück an Johns lokalen Computer gesendet.

Themen

- Einrichten von Session Manager
- Arbeiten mit Session Manager
- Prüfen von Sitzungsaktivitäten
- Protokollierung von Sitzungsaktivitäten aktivieren und deaktivieren
- Schema des Sitzungsdokuments
- Fehlerbehebung für Session Manager

Einrichten von Session Manager

Führen Sie AWS Systems Manager Session Manager die Schritte in den folgenden Themen aus, bevor Sie eine Verbindung zu den verwalteten Knoten in Ihrem Konto herstellen.

Themen

- Schritt 1: Erfüllen der Session Manager-Voraussetzungen
- Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager
- Schritt 3: Steuern des Sitzungs-Zugriffs auf verwaltete Knoten
- Schritt 4: Konfigurieren von Sitzungspräferenzen
- Schritt 5: (Optional) Beschränken des Zugriffs auf Befehle in einer Sitzung
- Schritt 6: (Optional) Verwenden von AWS PrivateLink zum Einrichten eines VPC-Endpunkts für Session Manager
- Schritt 7: (Optional) Deaktivieren oder Aktivieren der Administratorberechtigungen für das SSM-Benutzerkonto
- Schritt 8: (Optional) Erlauben und Steuern von Berechtigungen für SSH-Verbindungen über Session Manager

Schritt 1: Erfüllen der Session Manager-Voraussetzungen

Stellen Sie vor der Verwendung von Session Manager sicher, dass Ihre Umgebung den folgenden Anforderungen entspricht.

Session Manager-Voraussetzungen

Anforderung	Beschreibung
Unterstützte Betriebssysteme	Session Manager unterstützt die Verbindung zu Amazon Elastic Compute Cloud (Amazon EC2)-Instances und Nicht-EC2-Maschine n in Ihrer Hybrid- und Multi-Cloud-Umgebu ng, die das Advanced-Instances-Kontingent verwenden. Session Manager unterstützt die folgenden Betriebssystemversionen: **Note** Session Manager unterstützt EC2-Insta nces, Edge-Geräte und On-Premis es-Server und virtuelle Maschinen (VMs) in Ihrer Hybrid-und Multi-Clo ud-Umgebung, die das Advanced-Instances-Kontingent verwenden. Weitere Informationen über erweitert e Instances finden Sie unter Konfiguri eren von Instance-Kontingenten. Linux und macOS Session Managerunterstützt alle Versionen von Linux und macOS die von unterstützt werden AWS Systems Manager. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Maschinentypen.

Anforderung	Beschreibung
	Windows Session Manager unterstützt Windows Server 2012 bis Windows Server 2022. **Note** Microsoft Windows Server 2016 Nano wird nicht unterstützt.

Anforderung	Beschreibung
SSM Agent	Auf den verwalteten Knoten, zu denen Sie über Sitzungen eine Verbindung herstellen möchten, muss mindestens AWS Systems Manager SSM Agent Version 2.3.68.0 oder höher installiert sein. Um die Option zum Verschlüsseln von Sitzungsdaten mithilfe eines in AWS Key Management Service (AWS KMS) erstellten Schlüssels verwenden zu können, SSM Agent muss Version 2.3.539.0 oder höher auf dem verwalteten Knoten installiert sein. Um Shell-Profile in einer Sitzung zu verwenden, muss SSM Agent Version 3.0.161.0 oder höher auf dem verwalteten Knoten installiert sein. Um eine Session Manager-Port-Weiterleitung oder SSH-Sitzung zu starten, muss SSM Agent Version 3.0.222.0 oder höher auf dem verwaltet en Knoten installiert sein. Um Sitzungsdaten mit Amazon CloudWatch Logs zu streamen, muss SSM Agent Version 3.0.284.0 oder höher auf dem verwalteten Knoten installiert sein. Informationen zum Ermitteln der auf einer Instance ausgeführten Versionsnummer finden Sie unter Überprüfen der SSM Agent-Ver sionsnummer. Informationen über das manuelle Installieren oder automatische Aktualisieren von SSM Agent finden Sie unter Arbeiten mit SSM Agent. Über das ssm-user-Konto

Anforderung	Beschreibung
	Beginnend mit Version 2.3.50.0 von SSM Agent erstellt der Agent unter Verwendung der Root- oder Administratorberechtigungen ein Benutzerkonto auf dem verwalteten Knoten, das `ssm-user` genannt wird. (Auf Versionen vor 2.3.612.0 wird das Konto erstellt, wenn SSM Agent startet oder neu startet. Auf Version 2.3.612.0 und höher wird `ssm-user` erstellt, wenn eine Sitzung auf dem verwalteten Knoten zum ersten Mal gestartet wird.) Sitzungen werden mittels der Anmeldeinformationen für dieses Benutzerkonto gestartet. Weitere Informationen zum Einschränken der administrativen Kontrolle für dieses Konto finden Sie unter Deaktivieren oder Aktivieren der Administratorberechtigungen für das SSM-Benutzerkonto. ssm-user auf Windows Server-Domain-Controller Ab SSM Agent Version 2.3.612.0 wird das `ssm-user`-Konto nicht automatisch auf verwalteten Knoten erstellt, die als Windows Server-Domain-Controller verwendet werden. Um Session Manager auf einer Windows Server-Maschine zu verwenden, die als Domain-Controller verwendet wird, erstellen Sie das `ssm-user`-Konto manuell, sofern es noch nicht vorhanden ist, und weisen dem Benutzer Domain-Administratorberechtigungen zu. Unter Windows Server legt SSM Agent bei jedem Start einer Sitzung ein neues Passwort für das `ssm-user`-Konto fest. Es ist also nicht erforderl

Anforderung	Beschreibung
	ich, ein Passwort anzugeben, wenn Sie das Konto erstellen.
Konnektivität mit Endpunkten	In diesem Fall müssen die verwalteten Knoten auch ausgehenden HTTPS-Datenverkehr (Port 443) zu den folgenden Endpunkten zulassen: • ec2messages.*region*.amazonaws.com • ssm.*region*.amazonaws.com • ssmmessages.*region*.amazonaws.com Weitere Informationen finden Sie unter den folgenden Themen: • Referenz: ec2messages, ssmmessages und andere API-Operationen • Wie erstelle ich VPC-Endpoints, sodass ich Systems Manager verwenden kann, um private EC2-Instances ohne Internetzugang zu verwalten? im Knowledge Center AWS re:Post . Alternativ können Sie sich über Schnittst ellenendpunkte mit den erforderlichen Endpunkten verbinden. Weitere Informati onen finden Sie unter Schritt 6: (Optional) Verwenden von AWS PrivateLink zum Einrichte n eines VPC-Endpunkts für Session Manager.

Anforderung	Beschreibung
AWS CLI	(Optional) Wenn Sie AWS Command Line Interface (AWS CLI) verwenden, um Ihre Sitzungen zu starten (anstatt die AWS Systems Manager Konsole oder die Amazon EC2 EC2-Konsole zu verwenden), muss Version 1.16.12 oder höher der CLI auf Ihrem lokalen Computer installiert sein. Zum Überprüfen der Version können Sie den Befehl `aws --version` aufrufen. Wenn Sie die CLI installieren oder aktualisieren müssen, finden Sie weitere Informationen unter Installation von AWS Command Line Interface im AWS Command Line Interface Benutzerhandbuch. **Important** Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates

Anforderung	Beschreibung
	für SSM Agent. Abonnieren Sie die Seite mit den SSM AgentVersionshinweisenGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten. Um die CLI zur Verwaltung Ihrer Knoten mit Session Manager verwenden zu können, müssen Sie zunächst das Session Manager-Plug-In auf Ihrer lokalen Maschine installieren. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.
Aktivieren des Advanced-Instances-Kontingents (Hybrid- und Multi-Cloud-Umgebungen)	Um eine Verbindung zu Nicht-EC2-Computern herzustellenSession Manager, müssen Sie die Stufe „Advanced-Instances“ in dem Bereich aktivieren, in AWS-Region dem AWS-Konto Sie Hybrid-Aktivierungen erstellen, um Nicht-EC2-Computer als verwaltete Knoten zu registrieren. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Weitere Informationen zum Aktivieren des Advanced-Instances-Kontingent finden Sie unter Konfigurieren von Instance-Kontingenten.

Anforderung	Beschreibung
Überprüfen der Berechtigungen für IAM-Servicerollen (Hybrid- und Multi-Cloud-Umgebungen)	Hybrid-aktivierte Knoten verwenden die in der Hybrid-Aktivierung angegebene Dienstrolle AWS Identity and Access Management (IAM), um mit Systems Manager Manager-API-Vorgängen zu kommunizieren. Diese Servicerolle muss die Berechtigungen enthalten, die zum Herstellen einer Verbindung mit Ihren Hybrid- und Multi-Cloud-Maschinen mit Session Manager erforderlich sind. Wenn Ihre Servicerolle die AWS verwaltete Richtlinie enthält`AmazonSSMManagedInstanceCore`, Session Manager sind die erforderlichen Berechtigungen für bereits bereitgestellt. Wenn Sie feststellen, dass die Servicerolle nicht die erforderlichen Berechtigungen enthält, müssen Sie die verwaltete Instance abmelden und sie bei einer neuen Hybrid-Aktivierung registrieren, die eine IAM-Servicerolle mit den erforderlichen Berechtigungen verwendet. Informationen über das Abmelden verwalteter Instances finden Sie unter Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung. Weitere Informationen zum Erstellen von IAM-Richtlinien mit Session Manager-Berechtigungen finden Sie unter Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager.

Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager

Hat standardmäßig AWS Systems Manager keine Berechtigung, Aktionen auf Ihren Instances durchzuführen. Sie können Instance-Berechtigungen auf Kontoebene mithilfe einer AWS Identity and Access Management (IAM)-Rolle oder auf Instance-Ebene mithilfe eines Instance-Profils bereitstellen. Wenn Ihr Anwendungsfall dies zulässt, empfehlen wir, mithilfe der Standardkonfiguration für die Host-Verwaltung Zugriff auf Kontoebene zu gewähren. Wenn Sie die Standardkonfiguration für die Host-Verwaltung für Ihr Konto bereits mithilfe der `AmazonSSMManagedEC2InstanceDefaultPolicy`-Richtlinie eingerichtet haben, können Sie mit dem nächsten Schritt fortfahren. Weitere Informationen über die Standardkonfiguration für die Host-Verwaltung finden Sie unter Verwenden der Standardeinstellung für die Host-Management-Konfiguration.

Alternativ können Sie auch Instance-Profile verwenden, um Ihren Instances die erforderlichen Berechtigungen zu erteilen. Ein Instance-Profil übergibt eine IAM-Rolle an eine Amazon-EC2-Instance. Sie können ein IAM-Instance-Profil einer Amazon-EC2-Instance beim Starten anfügen oder einer zuvor gestarteten Instance anfügen. Weitere Informationen finden Sie unter Verwenden von Instance-Profilen.

Für On-Premises-Server oder virtuelle Maschinen (VMs) werden Berechtigungen von der IAM-Servicerolle bereitgestellt, die der Hybrid-Aktivierung zugeordnet ist, die zum Anmelden Ihrer On-Premises-Server und VMs bei Systems Manager verwendet wird. On-Premises-Server und VMs verwenden keine Instance-Profile.

Wenn Sie bereits andere Systems-Manager-Funktionen wie Run Command oder Parameter Store verwenden, ist Ihren Amazon-EC2-Instances möglicherweise bereits ein Instance-Profil mit den für Session Manager erforderlichen Grundberechtigungen angefügt. Wenn Ihren Instances bereits ein Instance-Profil angefügt ist, das die AWS -verwaltete Richtlinie `AmazonSSMManagedInstanceCore` enthält, sind die erforderlichen Berechtigungen für Session Manager bereits vorhanden. Dies gilt auch, wenn die bei Ihrer Hybrid-Aktivierung verwendete IAM-Servicerolle die verwaltete Richtlinie `AmazonSSMManagedInstanceCore` enthält.

> **Important**
>
> Sie können die IAM-Servicerolle, die einer Hybrid-Aktivierung zugeordnet ist, nicht ändern. Wenn Sie feststellen, dass die Servicerolle nicht die erforderlichen Berechtigungen enthält, müssen Sie die verwaltete Instance abmelden und sie bei einer neuen Hybrid-Aktivierung

registrieren, die eine Servicerolle mit den erforderlichen Berechtigungen verwendet. Informationen über das Abmelden verwalteter Instances finden Sie unter Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung. Weitere Informationen zum Erstellen einer IAM-Dienstrolle für lokale Maschinen finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Dienstrolle in Hybrid- und Multicloud-Umgebungen.

In einigen Fällen müssen Sie jedoch möglicherweise die Berechtigungen ändern, die Ihrem Instance-Profil zugeordnet sind. Sie möchten beispielsweise einen engeren Satz von Instance-Berechtigungen bereitstellen, Sie haben eine benutzerdefinierte Richtlinie für Ihr Instance-Profil erstellt oder Sie möchten die Verschlüsselungsoptionen Amazon Simple Storage Service (Amazon S3) oder AWS Key Management Service (AWS KMS) zur Sicherung von Sitzungsdaten verwenden. Führen Sie in diesen Fällen einen der folgenden Schritte aus, um Session Manager-Aktionen auf Ihren Instances auszuführen:

- Einbetten von Berechtigungen für Session Manager-Aktionen in einer benutzerdefinierten IAM-Rolle

  Um einer vorhandenen IAM-Rolle, die nicht auf der AWS bereitgestellten Standardrichtlinie basiert, Berechtigungen für Session Manager Aktionen hinzuzufügen`AmazonSSMManagedInstanceCore`, folgen Sie den Schritten unter. Hinzufügen von Session Manager-Berechtigungen für eine vorhandene IAM-Rolle
- Erstellen einer benutzerdefinierten IAM-Rolle, die ausschließlich Session Manager-Berechtigungen besitzt

  Um eine IAM-Rolle zu erstellen, die ausschließlich Berechtigungen für Session Manager-Aktionen enthält, befolgen Sie die Schritte in Erstellen einer benutzerdefinierten IAM-Rolle für Session Manager.
- Erstellen und Verwenden einer neuen IAM-Rolle mit Berechtigungen für alle Systems-Manager-Aktionen

  Um eine IAM-Rolle für von Systems Manager verwaltete Instanzen zu erstellen, die eine Standardrichtlinie verwendet, die bereitgestellt wird, AWS um allen Systems Manager-Berechtigungen zu gewähren, folgen Sie den Schritten unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

Themen

- Hinzufügen von Session Manager-Berechtigungen für eine vorhandene IAM-Rolle
- Erstellen einer benutzerdefinierten IAM-Rolle für Session Manager

Hinzufügen von Session Manager-Berechtigungen für eine vorhandene IAM-Rolle

Gehen Sie wie folgt vor, um einer vorhandenen AWS Identity and Access Management (IAM)-Rolle Session Manager-Berechtigungen hinzuzufügen. Indem Sie einer bestehenden Rolle Berechtigungen hinzufügen, können Sie die Sicherheit Ihrer Computerumgebung erhöhen, ohne die AWS `AmazonSSMManagedInstanceCore` Richtlinie für Instance-Berechtigungen verwenden zu müssen.

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - Dieses Verfahren setzt voraus, dass Ihre vorhandene Rolle bereits andere Systems-Manager-`ssm`-Berechtigungen für Aktionen enthält, für die Sie den Zugriff erlauben möchten. Diese Richtlinie reicht allein nicht aus, um Session Manager verwenden zu können.
> - Das folgende Richtlinienbeispiel beinhaltet eine `s3:GetEncryptionConfiguration`-Aktion. Diese Aktion ist erforderlich, wenn Sie in den Session Manager-Protokollierungseinstellungen die Option S3-Protokollverschlüsselung erzwingen gewählt haben.

So fügen Sie Session Manager-Berechtigungen einer vorhandenen Rolle hinzu (Konsole)

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Roles.
3. Wählen Sie den Namen der Rolle aus, zu der Sie die Berechtigungen hinzufügen möchten.
4. Wählen Sie die Registerkarte Permissions (Berechtigungen).
5. Wählen Sie Berechtigungen hinzufügen und dann Eingebundene Richtlinie hinzufügen aus.
6. Wählen Sie den Tab JSON.

7. Ersetzen Sie den Inhalt der Standardrichtlinie durch den folgenden Inhalt. Ersetzen Sie *key-name* durch den Amazon-Ressourcennamen (ARN) des AWS Key Management Service-Schlüssels (AWS KMS key), den Sie verwenden möchten.

```
{
    "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": "key-name"
        }
    ]
}
```

Weitere Informationen über die Verwendung eines KMS-Schlüssels zum Verschlüsseln von Sitzungsdaten finden Sie unter So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Wenn Sie die AWS KMS-Verschlüsselung für Ihre Sitzungsdaten nicht verwenden, entfernen Sie die folgenden Inhalte aus der Richtlinie.

```
,
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "key-name"
        }
```

8. Wählen Sie Next: Markierungen (Weiter: Markierungen).
9. (Optional) Fügen Sie Tags hinzu, indem SieAdd tag (Tag hinzufügen) auswählen und die bevorzugten Tags für die Richtlinie eingeben.
10. Wählen Sie Weiter: Prüfen aus.
11. Geben Sie auf der Seite Review Policy (Richtlinie prüfen) im Feld Name (Name) einen Namen für die Inline-Richtlinie ein, z. B. **`SessionManagerPermissions`**.
12. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für die Richtlinie ein.

    Wählen Sie Create Policy (Richtlinie erstellen) aus.

Weitere Informationen über die `ssmmessages`-Aktionen finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.

Erstellen einer benutzerdefinierten IAM-Rolle für Session Manager

Sie können eine AWS Identity and Access Management (IAM-) Rolle erstellen, die Ihnen Session Manager die Berechtigung erteilt, Aktionen auf Ihren von Amazon EC2 verwalteten Instances durchzuführen. Sie können auch eine Richtlinie hinzufügen, um die Berechtigungen zu gewähren, die für das Senden von Sitzungsprotokollen an Amazon Simple Storage Service (Amazon S3) und Amazon CloudWatch Logs erforderlich sind.

Nachdem Sie die IAM-Rolle erstellt haben, finden Sie Informationen dazu, wie Sie die Rolle an eine Instance anhängen oder ersetzen können, auf der AWS re:Post Website unter Ein Instance-Profil anhängen oder ersetzen. Weitere Informationen über IAM-Instance-Profile und -Rollen finden Sie unter Verwendung von Instance-Profilen im IAM-Benutzerhandbuch und IAM-Rollen für Amazon EC2 im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances. Weitere Informationen zum Erstellen einer IAM-Dienstrolle für lokale Maschinen finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Dienstrolle in Hybrid- und Multicloud-Umgebungen.

Themen

- Erstellen einer IAM-Rolle mit geringstmöglichen Session Manager-Berechtigungen (Konsole)
- Erstellen einer IAM-Rolle mit Berechtigungen für Amazon S3 Session Manager und CloudWatch Logs (Konsole)

Erstellen einer IAM-Rolle mit geringstmöglichen Session Manager-Berechtigungen (Konsole)

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte IAM-Rolle mit einer Richtlinie zu erstellen, die ausschließlich Berechtigungen für Session Manager-Aktionen auf Ihren Instances bereitstellt.

So erstellen Sie ein Instance-Profil mit den geringstmöglichen Session Manager-Berechtigungen (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create policy (Richtlinie erstellen). (Wenn die Schaltfläche Get Started (Erste Schritte) angezeigt wird, klicken Sie darauf und wählen Sie anschließend Create Policy (Richtlinie erstellen) aus.)
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie den Standardinhalt durch folgende Richtlinie. Um Sitzungsdaten mit AWS Key Management Service (AWS KMS) zu verschlüsseln, ersetzen Sie *key-name* durch den Amazon-Ressourcennamen (ARN) des AWS KMS key , den Sie verwenden möchten.

```
{
    "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": "key-name"
        }
    ]
}
```

Weitere Informationen über die Verwendung eines KMS-Schlüssels zum Verschlüsseln von Sitzungsdaten finden Sie unter So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Wenn Sie keine AWS KMS Verschlüsselung für Ihre Sitzungsdaten verwenden, können Sie den folgenden Inhalt aus der Richtlinie entfernen.

```
,
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "key-name"
        }
```

5. Wählen Sie Weiter: Markierungen.
6. (Optional) Fügen Sie Tags hinzu, indem SieAdd tag (Tag hinzufügen) auswählen und die bevorzugten Tags für die Richtlinie eingeben.
7. Wählen Sie Weiter: Prüfen aus.
8. Geben Sie auf der Seite Review Policy (Richtlinie prüfen) im Feld Name (Name) einen Namen für die Inline-Richtlinie ein, z. B. **`SessionManagerPermissions`**.
9. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für die Richtlinie ein.
10. Wählen Sie Richtlinie erstellen aus.
11. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
12. Auf der Seite Create Role (Rolle erstellen) wählen Sie AWS service (-Service), und fürUse case (Anwendungsfall), wählen Sie EC2 aus.
13. Wählen Sie Weiter aus.

14. Aktivieren Sie auf der Seite Attached permissions policy (Richtlinie für angefügte Berechtigungen) das Kontrollkästchen links neben dem Namen der Richtlinie, die Sie gerade erstellt haben, z. B. **`SessionManagerPermissions`**.
15. Wählen Sie Weiter aus.
16. Geben Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen) für Role name (Rollenname) einen Namen für die IAM-Rolle ein, z. B. **`MySessionManagerRole`**.
17. (Optional) Geben Sie in Role description (Beschreibung der Rolle) eine Beschreibung für das Instance-Profil ein.
18. (Optional) Fügen Sie Tags hinzu, indem SieAdd tag (Tag hinzufügen) auswählen und die bevorzugten Tags für die Richtlinie eingeben.

    Wählen Sie Rolle erstellen aus.

Weitere Informationen zu `ssmmessages`-Aktionen finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.

Erstellen einer IAM-Rolle mit Berechtigungen für Amazon S3 Session Manager und CloudWatch Logs (Konsole)

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte IAM-Rolle mit einer Richtlinie zu erstellen, die Berechtigungen für Session Manager-Aktionen auf Ihren Instances bereitstellt. Die Richtlinie bietet auch die erforderlichen Berechtigungen für die Speicherung von Sitzungsprotokollen in Amazon Simple Storage Service (Amazon S3) -Buckets und Amazon CloudWatch Logs-Protokollgruppen.

> **Important**
>
> Um Sitzungsprotokolle an einen Amazon S3-Bucket auszugeben, der zu einem anderen AWS-Konto gehört, müssen Sie die `s3:PutObjectAcl`-Berechtigung dieser IAM-Rollen-Richtlinie hinzufügen. Außerdem müssen Sie sicherstellen, dass die Bucket-Richtlinie kontenübergreifenden Zugriff auf die IAM-Rolle gewährt, die vom besitzenden Konto verwendet wird, um dem Systems Manager Berechtigungen für verwaltete Instances zu gewähren. Wenn der Bucket die Verschlüsselung des Key Management Service (KMS) verwendet, muss die KMS-Richtlinie des Buckets diesen kontoübergreifenden Zugriff ebenfalls gewähren. Weitere Informationen zur Konfiguration von kontoübergreifenden Bucket-Berechtigungen in Amazon S3 finden Sie unter Gewährung von kontoübergreifenden Bucket-Berechtigungen im Benutzerhandbuch zu Amazon Simple Storage Service. Wenn

die kontoübergreifenden Berechtigungen nicht hinzugefügt werden, kann das Konto, das Eigentümer des Amazon-S3-Buckets ist, nicht auf die Sitzungsausgabeprotokolle zugreifen.

Informationen zum Angeben von Präferenzen für das Speichern von Sitzungsprotokollen finden Sie unter Protokollierung von Sitzungsaktivitäten aktivieren und deaktivieren.

So erstellen Sie eine IAM-Rolle mit Berechtigungen für Session Manager Amazon S3 und CloudWatch Logs (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create policy (Richtlinie erstellen). (Wenn die Schaltfläche Get Started (Erste Schritte) angezeigt wird, klicken Sie darauf und wählen Sie anschließend Create Policy (Richtlinie erstellen) aus.)
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie den Standardinhalt durch folgende Richtlinie. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
{
    "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:::DOC-EXAMPLE-BUCKET/s3-prefix/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "key-name"
        },
        {
            "Effect": "Allow",
            "Action": "kms:GenerateDataKey",
            "Resource": "*"
        }
    ]
}
```

5. Wählen Sie Weiter: Markierungen.
6. (Optional) Fügen Sie Tags hinzu, indem SieAdd tag (Tag hinzufügen) auswählen und die bevorzugten Tags für die Richtlinie eingeben.
7. Wählen Sie Weiter: Prüfen aus.
8. Geben Sie auf der Seite Review Policy (Richtlinie prüfen) im Feld Name (Name) einen Namen für die Inline-Richtlinie ein, z. B. **`SessionManagerPermissions`**.
9. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für die Richtlinie ein.
10. Wählen Sie Richtlinie erstellen aus.

11. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
12. Auf der Seite Create Role (Rolle erstellen) wählen Sie AWS service (-Service), und fürUse case (Anwendungsfall), wählen Sie EC2 aus.
13. Wählen Sie Weiter aus.
14. Aktivieren Sie auf der Seite Attached permissions policy (Richtlinie für angefügte Berechtigungen) das Kontrollkästchen links neben dem Namen der Richtlinie, die Sie gerade erstellt haben, z. B. **`SessionManagerPermissions`**.
15. Wählen Sie Weiter aus.
16. Geben Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen) für Role name (Rollenname) einen Namen für die IAM-Rolle ein, z. B. **`MySessionManagerRole`**.
17. (Optional) Geben Sie im Feld Role description (Rollenbeschreibung) eine Beschreibung für die Rolle ein.
18. (Optional) Fügen Sie Tags hinzu, indem SieAdd tag (Tag hinzufügen) auswählen und die bevorzugten Tags für die Richtlinie eingeben.
19. Wählen Sie Rolle erstellen aus.

Schritt 3: Steuern des Sitzungs-Zugriffs auf verwaltete Knoten

Sie gewähren oder entziehen den Session Manager-Zugriff auf verwaltete Knoten mit Hilfe von AWS Identity and Access Management (IAM)-Richtlinien. Sie können eine Richtlinie erstellen und sie einem IAM-Benutzer oder einer IAM-Gruppe zuordnen, die festlegt, mit welchen verwalteten Knoten sich der Benutzer oder die Gruppe verbinden kann. Sie können auch die Session Manager-API-Operationen festlegen, die der Benutzer oder die Gruppe auf diesen verwalteten Knoten durchführen kann.

Um Ihnen den Einstieg in die IAM-Berechtigungsrichtlinien für Session Manager zu erleichtern, haben wir Beispielrichtlinien für einen Endbenutzer und einen Administrator erstellt. Sie können diese Richtlinien mit nur geringfügigen Änderungen verwenden. Oder verwenden Sie sie als Leitfaden für die Erstellung benutzerdefinierter IAM-Richtlinien. Weitere Informationen finden Sie unter Muster-IAM-Richtlinien für Session Manager. Informationen dazu, wie Sie IAM-Richtlinien erstellen und diese Benutzern oder Gruppen anfügen, finden Sie unter Erstellen von IAM-Richtlinien und Hinzufügen und Entfernen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Informationen zu Sitzungs-ID-ARN-Formaten

Beim Erstellen einer IAM-Richtlinie für den Session Manager-Zugriff geben Sie eine Sitzungs-ID als Teil des Amazon-Ressourcennamens (ARN) an. Die Sitzungs-ID enthält den Benutzernamen als Variable. Um dies zu veranschaulichen, finden Sie hier das Format eines Session Manager-ARN und ein Beispiel:

```
arn:aws:ssm:region-id:account-id:session/session-id
```

Beispielsweise:

```
arn:aws:ssm:us-east-2:123456789012:session/JohnDoe-1a2b3c4d5eEXAMPLE
```

Weitere Informationen zur Verwendung von Variablen in IAM-Richtlinien finden Sie unter IAM-Richtlinienelemente: Variablen.

Themen

- Starten Sie eine Standard-Shell-Sitzung, indem Sie das Standard-Sitzungsdokument in den IAM-Richtlinien angeben
- Starten Sie eine Sitzung mit einem Dokument, indem Sie die Sitzungsdokumente in IAM-Richtlinien angeben
- Muster-IAM-Richtlinien für Session Manager
- Zusätzliche IAM-Beispielrichtlinien für Session Manager

Starten Sie eine Standard-Shell-Sitzung, indem Sie das Standard-Sitzungsdokument in den IAM-Richtlinien angeben

Wenn Sie die Konfiguration Session Manager für Ihre Sitzung vornehmen AWS-Konto oder wenn Sie die Sitzungseinstellungen in der Systems Manager Manager-Konsole ändern, erstellt das System ein SSM-Sitzungsdokument mit dem Namen`SSM-SessionManagerRunShell`. Dies ist das Standard-Sitzungsdokument. Session Manager verwendet dieses Dokument, um Ihre Sitzungseinstellungen zu speichern, die Informationen wie die folgenden enthalten:

- Ein Ort, an dem Sie Sitzungsdaten speichern möchten, z. B. ein Amazon Simple Storage Service (Amazon S3) -Bucket oder eine Amazon CloudWatch Logs-Protokollgruppe.
- Eine AWS Key Management Service (AWS KMS) Schlüssel-ID zum Verschlüsseln von Sitzungsdaten.
- Ob die Unterstützung von Run As für Ihre Sitzungen erlaubt ist.

Hier sehen Sie ein Beispiel für die Informationen, die im `SSM-SessionManagerRunShell`-Dokument Sitzungseinstellungen enthalten sind.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
    "s3BucketName": "DOC-EXAMPLE-BUCKET",
    "s3KeyPrefix": "MyS3Prefix",
    "s3EncryptionEnabled": true,
    "cloudWatchLogGroupName": "MyCWLogGroup",
    "cloudWatchEncryptionEnabled": false,
    "kmsKeyId": "1a2b3c4d",
    "runAsEnabled": true,
    "runAsDefaultUser": "RunAsUser"
  }
}
```

Standardmäßig verwendet Session Manager das Standard-Sitzungsdokument, wenn ein Benutzer eine Sitzung von AWS Management Console aus startet. Dies gilt entweder für Fleet Manager oder Session Manager in der Systems Manager Manager-Konsole oder für EC2 Connect in der Amazon EC2 EC2-Konsole. Session Managerverwendet auch das Standardsitzungsdokument, wenn ein Benutzer eine Sitzung mit einem AWS CLI Befehl wie dem folgenden Beispiel startet:

```
aws ssm start-session \
    --target i-02573cafcfEXAMPLE
```

Um eine Standard-Shell-Sitzung zu starten, müssen Sie das Standardsitzungsdokument in der IAM-Richtlinie angeben, wie im folgenden Beispiel gezeigt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnableSSMSession",
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
```

```
                "arn:aws:ec2:us-west-2:123456789012:instance/i-02573cafcfEXAMPLE",
                "arn:aws:ssm:us-west-2:123456789012:document/SSM-
SessionManagerRunShell"
            ]
        }
    ]
}
```

Starten Sie eine Sitzung mit einem Dokument, indem Sie die Sitzungsdokumente in IAM-Richtlinien angeben

Wenn Sie den AWS CLI-Befehl start-session mit dem Standard-Sitzungsdokument verwenden, können Sie den Dokumentnamen auslassen. Das System ruft automatisch das `SSM-SessionManagerRunShell`-Sitzungsdokument auf.

In allen anderen Fällen müssen Sie einen Wert für den `document-name`-Parameter angeben. Wenn ein Benutzer den Namen eines Sitzungsdokuments in einem Befehl angibt, überprüft das System seine IAM-Richtlinie, um sicherzustellen, dass er berechtigt ist, auf das Dokument zuzugreifen. Wenn sie nicht berechtigt sind, schlägt die Verbindungsanforderung fehl. In den folgenden Beispielen ist der `document-name`-Parameter im `AWS-StartPortForwardingSession`-Sitzungsdokument enthalten.

```
aws ssm start-session \
    --target i-02573cafcfEXAMPLE \
    --document-name AWS-StartPortForwardingSession \
    --parameters '{"portNumber":["80"], "localPortNumber":["56789"]}'
```

Erzwingen Sie beim Starten einer Sitzung eine Berechtigungsprüfung für das Sitzungsdokument

Um den Zugriff auf das `AWS-StartPortForwardingSession`-Sitzungsdokument zu beschränken, können Sie der IAM-Richtlinie des Benutzers ein Bedingungselement hinzufügen, das überprüft, ob der Benutzer expliziten Zugriff auf ein Sitzungsdokument hat. Wenn diese Bedingung angewendet wird, muss der Benutzer einen Wert für die Option `document-name` des `start-session`-Befehls angeben. Das folgende Bedingungselement führt, wenn es der Aktion `ssm:StartSession` in der IAM-Richtlinie hinzugefügt wird, eine Prüfung des Sitzungsdokumentzugriffs durch.

```
"Condition": {
    "BoolIfExists": {
        "ssm:SessionDocumentAccessCheck": "true"
    }
```

```
}
```

Wenn dieses Bedingungselement auf `true` festgelegt ist, muss expliziter Zugriff auf ein Sitzungsdokument in der IAM-Richtlinie gewährt werden, damit der Benutzer eine Sitzung starten kann. Um sicherzustellen, dass das Bedingungselement durchgesetzt wird, muss es in allen Richtlinienanweisungen enthalten sein, die die `ssm:StartSession`-Aktion erlauben. Ein Beispiel.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnableSSMSession",
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:us-west-2:123456789012:instance/i-02573cafcfEXAMPLE",
                "arn:aws:ssm:us-west-2::document/AWS-StartPortForwardingSession"
            ],
            "Condition": {
                "BoolIfExists": {
                    "ssm:SessionDocumentAccessCheck": "true"
                }
            }
        }
    ]
}
```

Wenn diese IAM-Richtlinie vorhanden ist und das `SessionDocumentAccessCheck`-Bedingungselement auf `true` gesetzt ist, müssen Benutzer den `document-name`-Parameter in ihren Befehl eingeben, wenn sie eine Sitzung mit der AWS CLI starten. Der Wert von `document-name` muss dem Dokument entsprechen, das im `Resource`-Abschnitt der IAM-Richtlinie angegeben ist. Wenn der Benutzer einen anderen Dokumentnamen eingibt oder den `document-name`-Parameter nicht angibt, schlägt die Anfrage fehl.

Wenn das `SessionDocumentAccessCheck`-Bedingungselement auf `false` eingestellt ist, hat dies keinen Einfluss auf die Evaluierung der IAM-Richtlinie.

Ein Beispiel für die Angabe eines Session Manager-Sitzungsdokuments in einer IAM-Richtlinie finden Sie unter Kurzeinführung in Endbenutzerrichtlinien für Session Manager.

Andere Szenarien

Zum Starten einer Sitzung mit SSH müssen sowohl auf dem anvisierten verwalteten Knoten als auch auf der lokalen Maschine des Benutzers bestimmte Konfigurationsschritte vorgenommen werden. Weitere Informationen finden Sie unter (Optional) Erlauben und Steuern von Berechtigungen für SSH-Verbindungen über Session Manager.

Muster-IAM-Richtlinien für Session Manager

Verwenden Sie die Beispiele in diesem Abschnitt, um Ihnen bei der Erstellung von AWS Identity and Access Management (IAM-) Richtlinien zu helfen, die die am häufigsten benötigten Zugriffsberechtigungen Session Manager bereitstellen.

> **Note**
>
> Sie können auch eine AWS KMS key Richtlinie verwenden, um zu kontrollieren, welche IAM-Entitäten (Benutzer oder Rollen) Zugriff auf Ihren KMS-Schlüssel erhalten. AWS-Konten Weitere Informationen finden Sie im AWS Key Management Service Entwicklerhandbuch unter Überblick über die Verwaltung des Zugriffs auf Ihre AWS KMS Ressourcen und die Verwendung wichtiger Richtlinien. AWS KMS

Themen

- Kurzeinführung in Endbenutzerrichtlinien für Session Manager
- Kurzeinführung in Administratorrichtlinien für Session Manager

Kurzeinführung in Endbenutzerrichtlinien für Session Manager

Verwenden Sie die folgenden Beispiele, um IAM-Endbenutzerrichtlinien für Session Manager zu erstellen.

Sie können eine Richtlinie erstellen, die es Benutzern ermöglicht, Sitzungen nur von der Session Manager Konsole und AWS Command Line Interface (AWS CLI), nur von der Amazon Elastic Compute Cloud (Amazon EC2) -Konsole oder von allen drei aus zu starten.

Diese Richtlinien bieten Endbenutzern die Möglichkeit, eine Sitzung zu einem bestimmten verwalteten Knoten zu starten und nur ihre eigenen Sitzungen zu beenden. Beispiele für Anpassungen, die Sie möglicherweise für die Richtlinie ausführen sollten, finden Sie unter Zusätzliche IAM-Beispielrichtlinien für Session Manager.

Ersetzen Sie in den folgenden Beispielrichtlinien jeden *Beispiel-Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

Lesen Sie die folgenden Abschnitte, um Beispielrichtlinien für den Bereich des Sitzungszugriffs anzuzeigen, den Sie bereitstellen möchten.

Session Manager and Fleet Manager

Verwenden Sie diese Beispielrichtlinie, um Benutzern die Möglichkeit zu geben, Sitzungen nur über die Fleet Manager Konsolen Session Manager und wieder aufzunehmen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:region:account-id:instance/instance-id",
                "arn:aws:ssm:region:account-id:document/SSM-
SessionManagerRunShell" (1)
            ],
            "Condition": {
                "BoolIfExists": {
                    "ssm:SessionDocumentAccessCheck":
 "true" (2)
                }
            }
        },
        {
            "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" (3)
            ],
            "Resource": "key-name"
        }
    ]
}
```

Amazon EC2

Verwenden Sie diese Beispielrichtlinie für Provider-Benutzer, die Sitzungen nur über die Amazon EC2-Konsole starten und wiederaufnehmen können. Diese Richtlinie bietet nicht alle Berechtigungen, die zum Starten von Sitzungen über die Session ManagerKonsole und die AWS CLI Konsole erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",

 "ssm:SendCommand" (4)
            ],
            "Resource": [
                "arn:aws:ec2:region:account-id:instance/instance-id",
                "arn:aws:ssm:region:account-id:document/SSM-
SessionManagerRunShell" (1)
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        }
    ]
}
```

AWS CLI

Verwenden Sie diese Beispielrichtlinie, um Benutzern die Möglichkeit zu geben, Sitzungen von der aus zu starten und fortzusetzen AWS CLI.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",

 "ssm:SendCommand" 4
            ],
            "Resource": [
                "arn:aws:ec2:region:account-id:instance/instance-id",
                "arn:aws:ssm:region:account-id:document/SSM-
SessionManagerRunShell" 1
            ],
            "Condition": {
                "BoolIfExists": {
```

```
                    "ssm:SessionDocumentAccessCheck":
 "true" 2
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [

 "kms:GenerateDataKey" 3
            ],
            "Resource": "key-name"
        }
    ]
}
```

[1] `SSM-SessionManagerRunShell` ist der Standardname des SSM-Dokuments, das Session Manager zum Speichern Ihrer Sitzungskonfiguration erstellt. Sie können stattdessen ein benutzerdefiniertes Sitzungsdokument erstellen und es in dieser Richtlinie angeben. Sie können das AWS bereitgestellte Dokument auch `AWS-StartSSHSession` für Benutzer angeben, die Sitzungen mit SSH starten. Informationen zu den Konfigurationsschritten, die zur Unterstützung von SSH-Sitzungen erforderlich sind, finden Sie unter (Optional) Zulassen und Steuern von Berechtigungen für SSH-Verbindungen über. Session Manager

[2] Wenn Sie das Bedingungselement `ssm:SessionDocumentAccessCheck` auf `true` festlegen, prüft das System, ob ein Benutzer expliziten Zugriff auf das definierte Sitzungsdokument, in diesem Beispiel `SSM-SessionManagerRunShell`, hat, bevor eine Sitzung eingerichtet wird. Weitere Informationen finden Sie unter Erzwingen Sie beim Starten einer Sitzung eine Berechtigungsprüfung für das Sitzungsdokument.

[3] Die `kms:GenerateDataKey`-Berechtigung ermöglicht die Erstellung eines Datenverschlüsselungsschlüssels, der zur Verschlüsselung von Sitzungsdaten verwendet wird. Wenn Sie die AWS Key Management Service (AWS KMS) -Verschlüsselung für Ihre Sitzungsdaten verwenden, ersetzen Sie *key-name* durch den Amazon Resource Name (ARN) des KMS-Schlüssels, den Sie verwenden möchten, im Format. `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-12345EXAMPLE` Wenn Sie keine KMS-Schlüsselverschlüsselung für Ihre Sitzungsdaten verwenden möchten, entfernen Sie den folgenden Inhalt aus der Richtlinie.

```
,
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "key-name"
        }
```

Informationen zur Verwendung AWS KMS zur Verschlüsselung von Sitzungsdaten finden Sie unter. So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole)

[4] Die Genehmigung für SendCommandist für Fälle erforderlich, in denen ein Benutzer versucht, eine Sitzung von der Amazon EC2 EC2-Konsole aus zu starten, die jedoch Session Manager zunächst auf die erforderliche Mindestversion aktualisiert werden SSM Agent muss. Run Commandwird verwendet, um einen Befehl an die Instance zu senden, um den Agenten zu aktualisieren.

Kurzeinführung in Administratorrichtlinien für Session Manager

Verwenden Sie die folgenden Beispiele, um IAM-Administratorrichtlinien für Session Manager zu erstellen.

Diese Richtlinien bieten Administratoren die Möglichkeit, eine Sitzung für verwaltete Knoten zu starten, die mit `Key=Finance,Value=WebServers` markiert sind, sowie die Berechtigung zum Erstellen, Aktualisieren und Löschen von Einstellungen und die Berechtigung, nur ihre eigenen Sitzungen zu beenden. Beispiele für Anpassungen, die Sie möglicherweise für die Richtlinie ausführen sollten, finden Sie unter Zusätzliche IAM-Beispielrichtlinien für Session Manager.

Sie können eine Richtlinie erstellen, die es Administratoren ermöglicht, diese Aufgaben nur von der Session Manager Konsole und AWS CLI nur von der Amazon EC2 EC2-Konsole aus oder von allen drei aus auszuführen.

Ersetzen Sie in den folgenden Beispielrichtlinien jeden *Beispiel-Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

Lesen Sie die folgenden Abschnitte, um Beispielrichtlinien für die drei Berechtigungsszenarien anzuzeigen.

Session Manager and CLI

Verwenden Sie diese Beispielrichtlinie für Provider-Administratoren, die sitzungsbezogene Aufgaben nur über die Session Manager-Konsole und die AWS CLI ausführen können. Diese Richtlinie bietet nicht alle Berechtigungen, die für die Ausführung von sitzungsbezogenen Aufgaben über die Amazon EC2-Konsole erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:region:account-id:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/Finance": [
                        "WebServers"
                    ]
                }
            }
        },
        {
            "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:region:account-id:document/SSM-
SessionManagerRunShell"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        }
    ]
}
```

Amazon EC2

Verwenden Sie diese Beispielrichtlinie für Provider-Administratoren, die sitzungsbezogene Aufgaben nur über die Amazon EC2-Konsole ausführen können. Diese Richtlinie bietet nicht alle Berechtigungen, die zum Ausführen von sitzungsbezogenen Aufgaben über die Session Manager-Konsole und die AWS CLI-Konsole erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",

 "ssm:SendCommand" 1
            ],
            "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": [
                "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

Verwenden Sie diese Beispielrichtlinie für Provider-Administratoren, die sitzungsbezogene Aufgaben über die Session Manager-Konsole, die AWS CLI und die Amazon EC2-Konsole ausführen können.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",

 "ssm:SendCommand" (1)
            ],
            "Resource": [
                "arn:aws:ec2:region:account-id:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/tag-key": [
                        "tag-value"
                    ]
                }
            }
        },
        {
            "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:region:account-id:document/SSM-
SessionManagerRunShell"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        }
    ]
}
```

[1] Die Berechtigung für SendCommand wird für Fälle benötigt, in denen ein Benutzer versucht, eine Sitzung von der Amazon-EC2-Konsole aus zu starten, aber zuerst ein Befehl zur SSM Agent-Aktualisierung gesendet werden muss.

Zusätzliche IAM-Beispielrichtlinien für Session Manager

Die folgenden Beispielrichtlinien helfen Ihnen beim Erstellen einer benutzerdefinierten AWS Identity and Access Management (IAM)-Richtlinien für alle Session Manager-Benutzerzugriffsszenarien, die Sie unterstützen müssen.

Themen

- Beispiel 1: Zugriff auf Dokumente in der Konsole gewähren
- Beispiel 2: Beschränken des Zugriffs auf bestimmte verwaltete Knoten
- Beispiel 3: Beschränken des Zugriffs anhand von Tags
- Beispiel 4: Benutzern erlauben, ausschließlich von ihnen gestartete Sitzungen zu beenden
- Beispiel 5: Benutzer erhalten vollständigen (administrativen) Zugriff auf alle Sitzungen

Beispiel 1: Zugriff auf Dokumente in der Konsole gewähren

Sie können Benutzern erlauben, ein benutzerdefiniertes Dokument anzugeben, wenn sie eine Sitzung über die Session-Manager-Konsole starten. Das folgende Beispiel für eine IAM-Richtlinie gewährt die Erlaubnis, auf Dokumente zuzugreifen, deren Namen mit **`SessionDocument-`** den angegebenen AWS-Region und AWS-Konto beginnen.

Zum Verwenden dieses Beispiels ersetzen Sie jeden *Platzhalter für Beispielressourcen* durch Ihre eigenen Informationen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
                "ssm:ListDocuments"
            ],
            "Resource": [
                "arn:aws:ssm:region:account-id:document/SessionDocument-*"

            ],
            "Condition": {
                "BoolIfExists": {
                    "ssm:SessionDocumentAccessCheck": "true"
                }
            }
        }
    ]
}
```

> **Note**
>
> Die Session-Manager-Konsole unterstützt nur Sitzungsdokumente mit einem `sessionType` von `Standard_Stream`, die zur Definition von Sitzungseinstellungen verwendet werden. Weitere Informationen finden Sie unter Schema des Sitzungsdokuments.

Beispiel 2: Beschränken des Zugriffs auf bestimmte verwaltete Knoten

Sie können eine IAM-Richtlinie erstellen, die definiert, mit welchen verwalteten Knoten ein Benutzer mithilfe von Session Manager eine Verbindung herstellen darf. Die folgende Richtlinie gewährt einem Benutzer beispielsweise die Berechtigung, seine Sitzungen auf drei bestimmten Knoten zu starten, zu beenden und fortzusetzen. Die Richtlinie schränkt den Benutzer ein, eine Verbindung zu anderen als den angegebenen Knoten herzustellen.

> **Note**
>
> Informationen zu Verbundbenutzern finden Sie unterBeispiel 4: Benutzern erlauben, ausschließlich von ihnen gestartete Sitzungen zu beenden.

```
{
    "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": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        }
    ]
}
```

Beispiel 3: Beschränken des Zugriffs anhand von Tags

Sie können den Zugriff auf verwaltete Knoten anhand bestimmter Tags einschränken. Im folgenden Beispiel darf der Benutzer Sitzungen (`Effect: Allow, Action: ssm:StartSession, ssm:ResumeSession`) auf jedem verwalteten Knoten (`Resource: arn:aws:ec2:`*`region`*`:`*`987654321098`*`:instance/*`) starten und fortsetzen, sofern es

sich bei dem Knoten um einen Finanzknoten WebServer (`ssm:resourceTag/Finance: WebServer`) handelt. Wenn der Benutzer einen Befehl an einen verwalteten Knoten sendet, der nicht markiert ist oder einen anderen Tag als `Finance: WebServer` hat, enthält das Befehlsergebnis `AccessDenied`.

```
{
    "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": [
                "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"
            ]
```

```
        }
    ]
}
```

Sie können IAM-Richtlinien erstellen, mit denen ein Benutzer Sitzungen mit verwalteten Knoten starten kann, die mit mehreren Tags markiert sind. Die folgende Richtlinie ermöglicht dem Benutzer das Starten von Sitzungen mit verwalteten Knoten, auf denen beide angegebenen Tags angewendet wurden. Wenn ein Benutzer einen Befehl an einen verwalteten Knoten sendet, der nicht mit beiden Tags markiert ist, enthält das Befehlsergebnis `AccessDenied`.

```
{
   "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": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-2:123456789012:document/SSM-
SessionManagerRunShell"
            ]
      }
   ]
}
```

Weitere Informationen zum Erstellen von IAM-Richtlinien finden Sie unter Verwaltete Richtlinien und eingebundene Richtlinien im IAM-Benutzerhandbuch. Weitere Informationen zum Markieren von verwalteten Knoten finden Sie unter Markieren verwalteter Knoten und Taggen Ihrer Amazon EC2 EC2-Ressourcen im Amazon EC2 EC2-Benutzerhandbuch (der Inhalt bezieht sich auf Windows und Linux verwaltete Knoten). Weitere Informationen zur Steigerung des Sicherheitsstatus in Bezug auf nicht autorisierte Befehle auf Root-Ebene auf Ihren verwalteten Knoten finden Sie unter Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent

Beispiel 4: Benutzern erlauben, ausschließlich von ihnen gestartete Sitzungen zu beenden

Session Managerbietet zwei Methoden, um zu steuern, welche Sitzungen ein verbundener Benutzer in Ihrem AWS-Konto Netzwerk beenden darf.

- Verwenden Sie die Variable `{aws:userid}` in einer AWS Identity and Access Management (IAM-) Berechtigungsrichtlinie. Verbundbenutzer können nur von ihnen gestartete Sitzungen beenden. Verwenden Sie für nicht-Verbundbenutzer die Variable `{aws:username}` anstelle von `{aws:userid}`.
- Verwenden Sie Tags, die von AWS Tags in einer IAM-Berechtigungsrichtlinie bereitgestellt werden. Sie nehmen eine Bedingung in die Richtlinie auf, die es Benutzern erlaubt, nur Sitzungen zu beenden, die mit bestimmten Tags versehen sind, die von AWS bereitgestellt wurden. Diese Methode funktioniert für alle Konten, auch solche, die über Verbundkennungen Zugriff auf AWS gewähren.

Methode 1: Gewähren Sie TerminateSession Berechtigungen mithilfe der Variablen **`{aws:username}`**

Mit der folgenden IAM-Richtlinie können Benutzer die IDs aller Sitzungen in Ihrem Konto anzeigen. Benutzer können jedoch nur über von ihnen gestartete Sitzungen mit verwalteten Knoten interagieren. Ein Benutzer, dem die folgende Richtlinie zugewiesen wurde, kann keine Verbindungen mit Sitzungen anderer Benutzer herstellen oder diese beenden. Die Richtlinie verwendet die Variable `{aws:username}`, um dies zu erreichen.

> **Note**
>
> Diese Methode funktioniert nicht für Konten, die mithilfe von Verbundkennungen Zugriff auf AWS gewähren.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:DescribeSessions"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "ssm:TerminateSession"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:username}-*"
            ]
        }
    ]
}
```

Methode 2: Gewähren Sie TerminateSession Berechtigungen mithilfe von Tags, die bereitgestellt werden von AWS

Sie können steuern, welche Sitzungen ein Benutzer beenden kann, indem Sie eine Bedingung mit bestimmten Tag-Schlüsselvariablen in einer IAM-Richtlinie verwenden. Die Bedingung gibt an, dass der Benutzer nur Sitzungen beenden kann, die mit einer oder beiden dieser spezifischen Tag-Schlüsselvariablen und einem angegebenen Wert gekennzeichnet sind.

Wenn ein Benutzer in Ihrem Umfeld eine Sitzung AWS-Konto startet, Session Manager wendet er zwei Ressourcen-Tags auf die Sitzung an. Das erste Ressourcen-Tag ist `aws:ssmmessages:target-id`, mit dem Sie die ID des Ziels angeben, das der Benutzer beenden darf. Das andere Ressourcen-Tag ist `aws:ssmmessages:session-id`, mit einem Wert im Format *`role-id`*`:`*`caller-specified-role-name`*.

> **Note**
>
> Session Manager unterstützt keine benutzerdefinierten Tags für diese IAM-Zugriffssteuerungsrichtlinie. Sie müssen die unten beschriebenen Resource-Tags verwenden AWS, die von bereitgestellt werden.

`aws:ssmmessages:target-id`

Mit diesem Tag-Schlüssel schließen Sie die ID des verwalteten Knotens als Wert in die Richtlinie ein. Im folgenden Richtlinienblock lässt die Bedingungsanweisung einen Benutzer nur den Knoten i-02573cafcfEXAMPLE beenden.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                "ssm:TerminateSession"
             ],
             "Resource": "*",
             "Condition": {
                 "StringLike": {
                     "ssm:resourceTag/aws:ssmmessages:target-id": [
                        "i-02573cafcfEXAMPLE"
                     ]
                 }
             }
         }
     ]
}
```

Wenn der Benutzer versucht, eine Sitzung zu beenden, für die ihm diese `TerminateSession`-Berechtigung nicht erteilt wurde, wird eine `AccessDeniedException`-Fehlermeldung angezeigt.

`aws:ssmmessages:session-id`

Dieser Tag-Schlüssel enthält als Wert in der Anforderung zum Starten einer Sitzung eine Variable für die Sitzungs-ID.

Das folgende Beispiel zeigt eine Richtlinie für Fälle, in denen der Aufrufertyp `User` ist. Der Wert, für den Sie für `aws:ssmmessages:session-id` angeben, ist die ID des Benutzers. In diesem Beispiel stellt `AIDIODR4TAW7CSEXAMPLE` die ID eines Benutzers in Ihrem AWS-Konto dar. Um die ID für einen Benutzer in Ihrem abzurufen AWS-Konto, verwenden Sie den IAM-Befehl,`get-user`. Weitere Informationen finden Sie unter get-user im AWS Identity and Access Management Abschnitt des IAM-Benutzerhandbuchs.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                "ssm:TerminateSession"
             ],
             "Resource": "*",
             "Condition": {
                 "StringLike": {
                     "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "AIDIODR4TAW7CSEXAMPLE"
                     ]
                 }
             }
         }
     ]
}
```

Das folgende Beispiel zeigt eine Richtlinie für Fälle, in denen der Aufrufertyp `AssumedRole` ist. Sie können die Variable `{aws:userid}` für den Wert verwenden, den Sie für `aws:ssmmessages:session-id` angeben. Alternativ können Sie eine Rollen-ID für den Wert, den Sie für `aws:ssmmessages:session-id` angeben, fest codieren. Wenn Sie eine Rollen-ID fest codieren, müssen Sie den Wert im Format *`role-id`*`:`*`caller-specified-role-name`* angeben. z. B. `AIDIODR4TAW7CSEXAMPLE:MyRole`.

> **Important**
>
> Damit System-Tags angewendet werden können, darf die von Ihnen bereitzustellende Rollen-ID nur folgende Zeichen enthalten: Unicode-Buchstaben, 0-9, Leerzeichen, _, ., :, /, =, +, -, @ und \.

Verwenden Sie den Befehl, um die Rollen-ID für eine Rolle in Ihrem AWS-Konto abzurufen. `get-caller-identity` Weitere Informationen finden Sie unter get-caller-identity in der AWS CLI Befehlsreferenz.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                "ssm:TerminateSession"
             ],
             "Resource": "*",
             "Condition": {
                 "StringLike": {
                     "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userid}*"
                     ]
                 }
             }
         }
     ]
}
```

Wenn ein Benutzer versucht, eine Sitzung zu beenden, für die ihm diese `TerminateSession`-Berechtigung nicht erteilt wurde, wird eine `AccessDeniedException`-Fehlermeldung angezeigt.

`aws:ssmmessages:target-id` und **`aws:ssmmessages:session-id`**

Sie können auch IAM-Richtlinien erstellen, die es einem Benutzer ermöglichen, Sitzungen zu beenden, die mit beiden System-Tags gekennzeichnet sind, wie in diesem Beispiel dargestellt.

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

Beispiel 5: Benutzer erhalten vollständigen (administrativen) Zugriff auf alle Sitzungen

Die folgende IAM-Richtlinie ermöglicht Benutzern die vollständige Interaktion mit allen verwalteten Knoten und allen Sitzungen, die von allen Benutzern für alle Knoten erstellt wurden. Diese Berechtigung sollte nur einem Administrator gewährt werden, der vollständige Kontrolle über die Session Manager-Aktivitäten Ihrer Organisation benötigt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:StartSession",
                "ssm:TerminateSession",
                "ssm:ResumeSession",
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Schritt 4: Konfigurieren von Sitzungspräferenzen

Benutzer, denen in ihrer AWS Identity and Access Management (IAM-) Richtlinie Administratorberechtigungen gewährt wurden, können Sitzungseinstellungen konfigurieren, darunter die folgenden:

- Aktivieren Sie den Run-As-Support für Linux-verwaltete Knoten. Dadurch ist es möglich, Sitzungen mit den Anmeldeinformationen eines bestimmten Betriebssystembenutzers zu starten, anstatt mit den Anmeldeinformationen eines vom System generierten `ssm-user` Kontos, das auf einem verwalteten AWS Systems Manager Session Manager Knoten erstellt werden kann.
- Konfigurieren Sie Session Manager die Konfiguration so, dass AWS KMS key Verschlüsselung verwendet wird, um die zwischen Client-Computern und verwalteten Knoten übertragenen Daten zusätzlich zu schützen.
- Konfigurieren Sie Session Manager die Konfiguration, um Sitzungsverlaufsprotokolle zu erstellen und an einen Amazon Simple Storage Service (Amazon S3) -Bucket oder eine Amazon CloudWatch Logs-Protokollgruppe zu senden. Die gespeicherten Protokolldaten können anschließend verwendet werden, um die Sitzungsverbindungen mit Ihren verwalteten Knoten und die auf diesen während der Sitzungen ausgeführten Befehle zu prüfen oder zu melden.
- Konfigurieren Sie Sitzungs-Timeouts. Mit dieser Einstellung können Sie festlegen, wann eine Sitzung nach einem Zeitraum der Inaktivität beendet werden soll.
- Konfigurieren Sie Session Manager so, dass es konfigurierbare Shell-Profile verwendet. Mit diesen anpassbaren Profilen können Sie Einstellungen in Sitzungen wie Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnissen und das Ausführen mehrerer Befehle definieren, wenn eine Sitzung gestartet wird.

Weitere Informationen zu den Berechtigungen, die zum Konfigurieren von Session Manager-Einstellungen erforderlich sind, finden Sie unter the section called “Gewähren oder Verweigern von Benutzerberechtigungen zum Aktualisieren von Session Manager-Einstellungen”.

Themen

- Gewähren oder Verweigern von Benutzerberechtigungen zum Aktualisieren von Session Manager-Einstellungen
- Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen
- Angeben der maximalen Sitzungsdauer
- Konfigurierbare Shell-Profile zulassen

- Aktivieren Sie „Als ausführen“ -Unterstützung für Linux und macOS verwaltete Knoten
- So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole)
- Erstellen eines Dokuments mit Session Manager-Einstellungen (Befehlszeile)
- Aktualisieren von Session Manager-Einstellungen (Befehlszeile)

Weitere Informationen zur Verwendung der Systems Manager-Konsole zum Konfigurieren von Optionen für die Protokollierung von Sitzungsdaten finden Sie in den folgenden Themen:

- Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)
- Streaming-Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)
- Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

Gewähren oder Verweigern von Benutzerberechtigungen zum Aktualisieren von Session Manager-Einstellungen

Kontoeinstellungen werden als AWS Systems Manager (SSM)-Dokumente für jede AWS-Region gespeichert. Bevor Benutzer die Kontoeinstellungen für Sitzungen in Ihrem Konto aktualisieren können, müssen ihnen die notwendigen Berechtigungen für den Zugriff auf die Art des SSM-Dokuments gewährt werden, in denen diese Einstellungen gespeichert werden. Diese Berechtigungen werden über eine AWS Identity and Access Management (IAM)-Richtlinie erteilt.

Administratorrichtlinie, die das Erstellen und Aktualisieren von Richtlinien zulässt

Ein Administrator kann die folgende Richtlinie zum jederzeitigen Erstellen und Aktualisieren von Einstellungen besitzen. Die folgende Richtlinie gewährt die Berechtigung für Zugriff und Aktualisierung des Dokuments `SSM-SessionManagerRunShell` im Konto 123456789012 in der Region us-east-2.

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

Benutzerrichtlinie, die das Aktualisieren von Einstellungen verhindert

Mittels der folgenden Richtlinie verhindern Sie das Aktualisieren oder Überschreiben von Session Manager-Einstellungen durch Endbenutzer in Ihrem Konto.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:CreateDocument",
                "ssm:GetDocument",
                "ssm:UpdateDocument",
                "ssm:DeleteDocument"
            ],
            "Effect": "Deny",
            "Resource": [
                "arn:aws:ssm:us-east-2:123456789012:document/SSM-
SessionManagerRunShell"
            ]
        }
    ]
}
```

Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen

Mit Session Manager, einer Funktion von AWS Systems Manager, können Sie festlegen, wie lange Benutzer inaktiv sein können, bevor das System eine Sitzung beendet. Standardmäßig wird eine Sitzung nach 20 Minuten Inaktivität beendet. Sie können diese Einstellung ändern und eine Zeitüberschreitung zwischen 1 und 60 Minuten Inaktivität festlegen. Einige professionelle Agenturen für Computersicherheit empfehlen, Timeouts für inaktive Sitzungen auf maximal 15 Minuten festzulegen.

So lassen Sie Zeitüberschreitungen für Leerlaufsitzungen zu (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Geben Sie im Feld minutes unter Zeitüberschreitung bei Leerlaufsitzung an, wie lange ein Benutzer inaktiv sein kann, bevor eine Sitzung beendet wird.
5. Wählen Sie Save (Speichern).

Angeben der maximalen Sitzungsdauer

Session Manager, eine Fähigkeit von AWS Systems Manager, ermöglicht es Ihnen, die maximale Dauer einer Sitzung anzugeben, bevor sie endet. Standardmäßig haben Sitzungen keine maximale Dauer. Der Wert, den Sie für die maximale Sitzungsdauer angeben, muss zwischen 1 und 1 440 Minuten liegen.

So geben Sie die maximale Sitzungsdauer an (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Aktivieren Sie das Kontrollkästchen neben Enable maximum session duration (Aktivieren der maximalen Sitzungsdauer).
5. Geben Sie die maximale Sitzungsdauer in dem Feld minutes (Minuten) unter Maximum session duration (Maximale Sitzungsdauer) an.
6. Klicken Sie auf Speichern.

Konfigurierbare Shell-Profile zulassen

Standardmäßig starten Sitzungen auf EC2-Instances für Linux, die Bourne-Shell (sh) zu verwenden. Sie könnten jedoch eine andere Shell wie bash vorziehen. Indem Sie konfigurierbare Shell-Profile

zulassen, können Sie Einstellungen in Sitzungen wie Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnissen und das Ausführen mehrerer Befehle anpassen, wenn eine Sitzung gestartet wird.

> **Important**
>
> Systems Manager überprüft die Befehle oder Skripts in Ihrem Shell-Profil nicht, bevor sie ausgeführt werden, um zu sehen, welche Änderungen sie an einer Instance vornehmen würden. Um die Fähigkeit eines Benutzers, Befehle oder Skripte zu ändern, die in seinem Shell-Profil eingegeben wurden, einzuschränken, wird Folgendes empfohlen:
>
> - Erstellen Sie ein angepasstes Sitzungsdokument für Ihre AWS Identity and Access Management (IAM)-Benutzer und -Rollen. Ändern Sie dann die IAM-Richtlinie für diese Benutzer und Rollen so, dass die `StartSession` API-Operation nur das Sitzungstyp-Dokument verwenden kann, das Sie für sie erstellt haben. Weitere Informationen finden Sie unter Erstellen eines Dokuments mit Session Manager-Einstellungen (Befehlszeile) und Kurzeinführung in Endbenutzerrichtlinien für Session Manager.
> - Ändern Sie die IAM-Richtlinie für Ihre IAM-Benutzer und -Rollen, um den Zugriff auf die `UpdateDocument` API-Operation für die von Ihnen erstellte Sitzungstyp-Dokumentressource zu verweigern. Auf diese Weise können Benutzer und Rollen das von Ihnen erstellte Dokument für ihre Sitzungseinstellungen verwenden, ohne dass sie die Einstellungen ändern können.

So aktivieren Sie konfigurierbare Shell-Profile

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Geben Sie die Umgebungsvariablen, Shell-Einstellungen oder Befehle, die beim Start der Sitzung ausgeführt werden sollen, in den Feldern der entsprechenden Betriebssysteme an.
5. Wählen Sie Speichern.

Im Folgenden sehen Sie einige Beispielbefehle, die Ihrem Shell-Profil hinzugefügt werden können.

Wechseln Sie zur bash-Shell und wechseln Sie in das Verzeichnis /usr auf Linux-Instances.

```
exec /bin/bash
cd /usr
```

Geben Sie einen Zeitstempel und eine Begrüßungsnachricht zu Beginn einer Sitzung aus.

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."
```

Zeigen Sie die dynamische Systemaktivität zu Beginn einer Sitzung an.

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 "-------  ------    -----      ----- -----   ------     -- -----------"}
```

Aktivieren Sie „Als ausführen“ -Unterstützung für Linux und macOS verwaltete Knoten

Standardmäßig authentifiziert Session Manager Verbindungen mit den Anmeldeinformationen des vom System generierten `ssm-user`-Kontos, das auf einem verwalteten Knoten erstellt wird. (Auf Linux- und macOS-Maschinen wird das Konto zu `/etc/sudoers/` hinzugefügt.) Wenn Sie möchten, können Sie stattdessen Sitzungen mit den Anmeldeinformationen eines Benutzerkontos des Betriebssystems (OS) authentifizieren. In diesem Fall überprüft Session Manager, ob das angegebene Betriebssystemkonto auf dem Knoten vorhanden ist, bevor die Sitzung gestartet wird. Wenn Sie versuchen, eine Sitzung mit einem Betriebssystemkonto zu starten, das auf dem Knoten nicht vorhanden ist, schlägt die Verbindung fehl.

> **Note**
>
> Session Manager unterstützt nicht die Verwendung des `root`-Benutzerkontos eines Betriebssystems zur Authentifizierung von Verbindungen. Für Sitzungen, die mit einem Betriebssystem-Benutzerkonto authentifiziert werden, gelten die Betriebssystem- und Verzeichnisrichtlinien des Knotens, wie Anmeldeeinschränkungen oder Nutzungseinschränkungen für Systemressourcen, möglicherweise nicht.

Funktionsweise

Wenn Sie die Run As-Unterstützung für Sitzungen aktivieren, überprüft das System für Zugriffsberechtigungen wie folgt:

1. Wurde die IAM-Entität (Benutzer oder Rolle) des Benutzers, der die Sitzung startet, mit `SSMSessionRunAs = os user account name` gekennzeichnet?

   Falls ja, ist der Betriebssystem-Benutzername auf dem verwalteten Knoten vorhanden? Wenn dies der Fall ist, wird die Sitzung gestartet. Wenn dies nicht der Fall ist, wird das Starten der Sitzung verboten.

   Wenn die IAM-Entität nicht mit `SSMSessionRunAs = os user account name` gekennzeichnet wurde, fahren Sie mit Schritt 2 fort.
2. Wenn die IAM-Entität nicht markiert wurde`SSMSessionRunAs = os user account name`, wurde in den Session Manager Einstellungen von ein Betriebssystem-Benutzername angegeben? AWS-Konto

Falls ja, ist der Betriebssystem-Benutzername auf dem verwalteten Knoten vorhanden? Wenn dies der Fall ist, wird die Sitzung gestartet. Wenn dies nicht der Fall ist, wird das Starten der Sitzung verboten.

> **Note**
>
> Wenn Sie die Unterstützung „Ausführen als“ aktivieren, wird Session Manager daran gehindert, Sitzungen mit dem `ssm-user`-Konto auf einem verwalteten Knoten zu starten. Dies bedeutet, dass, wenn Session Manager die Verbindung nicht mithilfe des angegebenen Betriebssystem-Benutzerkontos herstellen kann, es nicht auf die Standardmethode zurückgreift.
> Wenn Sie „Ausführen als“ aktivieren, ohne ein Betriebssystemkonto anzugeben oder eine IAM-Entität zu markieren, und Sie in den Session Manager-Einstellungen kein Betriebssystemkonto angegeben haben, schlagen Sitzungsverbindungsversuche fehl.

So aktivieren Sie den Run-As-Support für Linux- und macOS-verwaltete Knoten

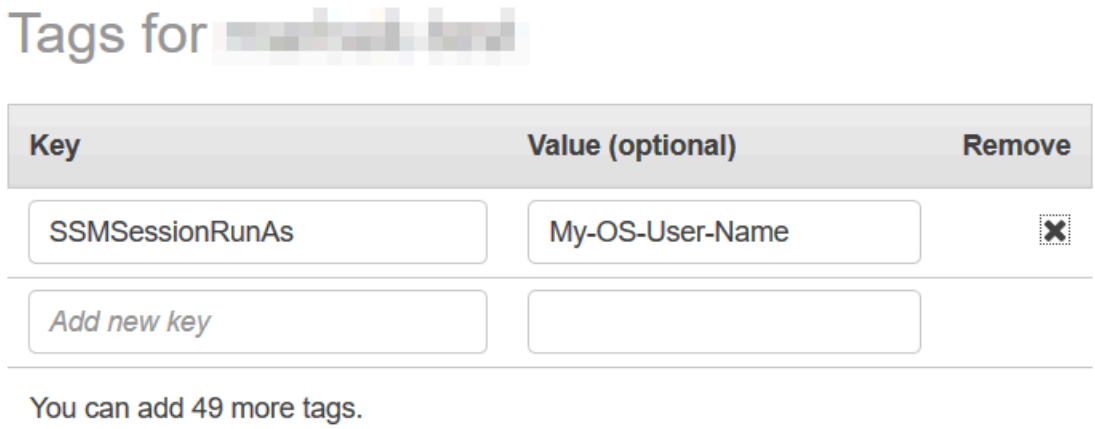
1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Aktivieren Sie das Kontrollkästchen neben Run As-Unterstützung für Linux-Instances aktivieren.
5. Führen Sie eine der folgenden Aktionen aus:

   - Option 1: Geben Sie im Feld Benutzername des Betriebssystems den Namen des Benutzerkontos des Betriebssystems auf dem verwalteten Zielknoten ein, den Sie zum Starten von Sitzungen verwenden möchten. Wenn Sie diese Option verwenden, werden alle Sitzungen von demselben Betriebssystembenutzer für alle Benutzer in Ihrem System ausgeführt AWS-Konto , die eine Verbindung herstellenSession Manager.
   - Option 2: (Empfohlen) Wählen Sie den Link zur IAM-Konsole aus. Wählen Sie im Navigationsbereich eine der Optionen Users (Benutzer) oder Roles (Rollen). Wählen Sie die Entität (Benutzer oder Rolle) aus, der Sie Tags hinzufügen möchten, und wählen Sie dann die Registerkarte Tags. Geben Sie `SSMSessionRunAs` als Schlüsselname ein. Geben Sie

den Namen eines Betriebssystem-Benutzerkontos als den Schlüsselwert ein. Wählen Sie Änderungen speichern aus.

Mit dieser Option können Sie bei Bedarf eindeutige Betriebssystembenutzer für verschiedene IAM-Entitäten angeben. Weitere Informationen zum Markieren von IAM-Ressourcen finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch

Im Folgenden wird ein Beispiel gezeigt.



6. Klicken Sie auf Speichern.

So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole)

Verwenden Sie AWS Key Management Service (AWS KMS), um Verschlüsselungsschlüssel zu erstellen und zu verwalten. Mit AWS KMS können Sie die Verwendung der Verschlüsselung in einer breiten Palette an AWS-Services und in Ihren Anwendungen steuern. Sie können festlegen, dass die Sitzungsdaten, die zwischen Ihren verwalteten Knoten und den lokalen Maschinen der Benutzer in Ihrem AWS-Konto übertragen werden, mithilfe der KMS-Schlüsselverschlüsselung verschlüsselt werden. (Dies wird zusätzlich zur TLS 1.2-Verschlüsselung angewendet, die AWS bereits standardmäßig zur Verfügung stellt.) Um Session Manager Sitzungsdaten zu verschlüsseln, erstellen Sie einen symmetrischen KMS-Schlüssel mit AWS KMS.

AWS KMS Die -Verschlüsselung ist für die `NonInteractiveCommands` Sitzungstypen `Standard_StreamInteractiveCommands`, und verfügbar. Damit Sie die Option zum Verschlüsseln von Sitzungsdaten mit einem in AWS KMS erstellten Schlüssel verwenden können, muss Version 2.3.539.0 oder höher des AWS Systems Manager SSM Agent auf dem verwalteten Knoten installiert sein.

> **Note**
>
> Sie müssen die AWS KMS-Verschlüsselung erlauben, um Passwörter auf Ihren verwalteten Knoten über die AWS Systems Manager-Konsole zurücksetzen. Weitere Informationen finden Sie unter Zurücksetzen eines Passworts auf einem verwalteten Knoten.

Sie können einen Schlüssel verwenden, den Sie in Ihrem AWS-Konto erstellt haben. Sie können jedoch auch einen Schlüssel verwenden, der in einem anderen AWS-Konto erstellt wurde. Der Ersteller des Schlüssel in einem anderen AWS-Konto muss Ihnen die erforderlichen Berechtigungen zur Nutzung des Schlüssels erteilen.

Nachdem Sie die KMS-Schlüsselverschlüsselung für Ihre Sitzungsdaten aktiviert haben, müssen sowohl die Benutzer, die Sitzungen starten, als auch die verwalteten Knoten, mit denen sie verbunden sind, über die Berechtigung zur Verwendung des Schlüssels verfügen. Sie erteilen die Berechtigung zur Verwendung des KMS-Schlüssels mit Session Manager anhand von AWS Identity and Access Management (IAM)-Richtlinien. Weitere Informationen finden Sie unter den folgenden Themen:

- Hinzufügen von AWS KMS-Berechtigungen für Benutzer in Ihrem Konto: Muster-IAM-Richtlinien für Session Manager.
- Hinzufügen von AWS KMS-Berechtigungen für verwalteten Knoten in Ihrem Konto: Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager.

Weitere Informationen zum Erstellen und Verwalten von KMS-Schlüsseln finden Sie im AWS Key Management Service-Entwicklerhandbuch.

Weitere Informationen zur Verwendung der AWS CLI zur Aktivierung der KMS-Schlüsselverschlüsselung von Sitzungsdaten in Ihrem Konto finden Sie unter Erstellen eines Dokuments mit Session Manager-Einstellungen (Befehlszeile) oder Aktualisieren von Session Manager-Einstellungen (Befehlszeile).

> **Note**
>
> Es entstehen Kosten für die Verwendung von KMS-Schlüsseln. Weitere Informationen finden Sie unter AWS Key Management Service-Preise.

So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Aktivieren Sie das Kontrollkästchen neben Enable KMS encryption (Aktivieren der KMS-Verschlüsselung).
5. Führen Sie eine der folgenden Aktionen aus:

   - Klicken Sie auf die Schaltfläche neben Select a KMS key in my current account (Einen KMS-Schlüssel in meinem aktuellen Konto auswählen) und wählen Sie anschließend einen Schlüssel aus der Liste aus.

     –oder–

     Wählen Sie die Schaltfläche neben Enter a KMS key alias or KMS key ARN (Einen KMS-Schlüssel-Alias oder KMS-Schlüssel-ARN eingeben) aus. Geben Sie manuell einen KMS-Schlüssel-Alias für einen Schlüssel ein, der in Ihrem aktuellen Konto erstellt wurde. Für einen Schlüssel in einem anderen Konto geben Sie den Amazon-Ressourcennamen (ARN) des Schlüssels ein. Im Folgenden sind einige Beispiele aufgeführt:

     - Schlüssel-Alias: `alias/my-kms-key-alias`
     - Schüssel-ARN: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-12345EXAMPLE`

     –oder–

     Wählen Sie Create new key (Neuen Schlüssel erstellen), um einen neuen KMS-Schlüssel in Ihrem Konto zu erstellen. Nachdem Sie den neuen Schlüssel erstellt haben, kehren Sie zur Registerkarte Preferences (Einstellungen) zurück und wählen Sie den Schlüssel zum Verschlüsseln der Sitzungsdaten in Ihrem Konto aus.

   Weitere Informationen zum Freigeben von Schlüsseln finden Sie unter Externen AWS-Konten den Zugriff auf einen Schlüssel erlauben im AWS Key Management Service-Entwicklerhandbuch.
6. Klicken Sie auf Speichern.

Erstellen eines Dokuments mit Session Manager-Einstellungen (Befehlszeile)

Gehen Sie wie folgt vor, um SSM-Dokumente zu erstellen, die Ihre Einstellungen für AWS Systems Manager Session Manager Sitzungen definieren. Sie können das Dokument verwenden, um Sitzungsoptionen wie Datenverschlüsselung, Sitzungsdauer und Protokollierung zu konfigurieren. Sie können beispielsweise angeben, ob Sitzungsprotokolldaten in einem Amazon Simple Storage Service (Amazon S3) -Bucket oder einer Amazon CloudWatch Logs-Protokollgruppe gespeichert werden sollen. Sie können Dokumente erstellen, die allgemeine Einstellungen für alle Sitzungen für ein AWS-Konto und AWS-Region oder Einstellungen für einzelne Sitzungen definieren.

> **Note**
>
> Sie können die allgemeinen Sitzungseinstellungen auch über die Session-Manager-Konsole konfigurieren.

Dokumente, die zum Einstellen von Session-Manager-Einstellungen verwendet werden, müssen einen `sessionType` von `Standard_Stream` haben. Weitere Informationen zu Sitzungs-Dokumenten finden Sie unter the section called “Schema des Sitzungsdokuments”.

Weitere Informationen zur Verwendung der Befehlszeile zum Aktualisieren vorhandener Session Manager-Einstellungen finden Sie unter Aktualisieren von Session Manager-Einstellungen (Befehlszeile).

Ein Beispiel für die Erstellung von Sitzungseinstellungen mit AWS CloudFormation finden Sie unter Erstellen eines Systems Manager Manager-Dokuments für Session Manager Einstellungen im AWS CloudFormation Benutzerhandbuch.

> **Note**
>
> In diesem Verfahren wird beschrieben, wie Dokumente für die Festlegung von Session Manager Einstellungen auf der AWS-Konto Ebene erstellt werden. Um Dokumente zu erstellen, die für die Festlegung von Einstellungen auf Sitzungsebene verwendet werden, geben Sie einen anderen Wert als `SSM-SessionManagerRunShell` für die dateibezogenen Befehlseingaben an.
>
> Wenn Sie Ihr Dokument verwenden möchten, um Einstellungen für Sitzungen festzulegen, die mit der AWS Command Line Interface (AWS CLI) gestartet wurden, geben Sie den Dokumentnamen als `--document-name`-Parameterwert an. Um Einstellungen für Sitzungen

vorzunehmen, die von der Session-Manager-Konsole aus gestartet werden, können Sie den Namen Ihres Dokuments eingeben oder aus einer Liste auswählen.

So erstellen Sie Session Manager-Einstellungen (Befehlszeile)

1. Erstellen Sie eine JSON-Datei auf Ihrem lokalen Computer und geben Sie Ihr beispielsweise einen Namen wie `SessionManagerRunShell.json`. Fügen Sie der Datei anschließend den folgenden Inhalt ein.

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

Sie können Werte auch mithilfe von Parametern an Ihre Sitzungseinstellungen übergeben, anstatt die Werte fest zu kodieren, wie im folgenden Beispiel gezeigt.

```
{
   "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. Legen Sie fest, wohin Sie die Sitzungsdaten senden möchten. Sie können einen S3-Bucket-Namen (mit optionalem Präfix) oder einen CloudWatch Logs-Log-Gruppennamen angeben. Wenn Sie die Daten zwischen dem lokalen Client und den verwalteten Knoten weiter verschlüsseln möchten, geben Sie den KMS-Schlüssel ein, der für die Verschlüsselung verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
    "s3BucketName": "DOC-EXAMPLE-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

Wenn Sie die Protokolldaten der Sitzung nicht verschlüsseln möchten, ändern Sie für `s3EncryptionEnabled` `true` in `false`.

Wenn Sie keine Protokolle an einen Amazon S3 S3-Bucket oder eine CloudWatch Logs-Protokollgruppe senden, aktive Sitzungsdaten nicht verschlüsseln oder die Unterstützung „Als ausführen“ für die Sitzungen in Ihrem Konto nicht aktivieren möchten, können Sie die Zeilen für diese Optionen löschen. Überprüfen Sie, dass die letzte Zeile im Abschnitt `inputs` nicht mit einem Komma endet.

Wenn Sie eine KMS-Schlüssel-ID zum Verschlüsseln Ihrer Sitzungsdaten hinzufügen, müssen sowohl die Benutzer, die die Sitzungen starten, als auch die verwalteten Knoten, mit denen sie sich verbinden, über die Berechtigung zur Verwendung des Schlüssels verfügen. Sie erteilen die Berechtigung zur Verwendung des KMS-Schlüssels mit Session Manager anhand von IAM-Richtlinien. Weitere Informationen finden Sie unter den folgenden Themen:

- Fügen Sie AWS KMS Berechtigungen für Benutzer in Ihrem Konto hinzu: Muster-IAM-Richtlinien für Session Manager
- Fügen Sie AWS KMS Berechtigungen für verwaltete Knoten in Ihrem Konto hinzu: Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager

3. Speichern Sie die Datei.
4. Führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, den folgenden Befehl aus.

   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
   ```

   Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

   ```
   {
       "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"
    }
}
```

Aktualisieren von Session Manager-Einstellungen (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie Ihr bevorzugtes Befehlszeilentool verwenden, um Änderungen an den AWS Systems Manager Session Manager Einstellungen für Ihr AWS-Konto ausgewähltes Konto vorzunehmen AWS-Region. Verwenden Sie Session Manager Einstellungen, um Optionen für die Protokollierung von Sitzungsdaten in einem Amazon Simple Storage Service (Amazon S3) -Bucket oder einer Amazon CloudWatch Logs-Protokollgruppe festzulegen. Mithilfe der Session Manager-Einstellungen können Sie zudem Ihre Sitzungsdaten verschlüsseln.

So aktualisieren Sie Session Manager-Einstellungen (Befehlszeile)

1. Erstellen Sie eine JSON-Datei auf Ihrem lokalen Computer und geben Sie Ihr beispielsweise einen Namen wie `SessionManagerRunShell.json`. Fügen Sie der Datei anschließend den folgenden Inhalt ein.

```
{
    "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. Legen Sie fest, wohin Sie die Sitzungsdaten senden möchten. Sie können einen S3-Bucket-Namen (mit einem optionalen Präfix) oder einen CloudWatch Logs-Protokollgruppennamen angeben. Wenn Sie Daten zwischen dem lokalen Client und den verwalteten Knoten weiter verschlüsseln möchten, geben Sie den für die Verschlüsselung AWS KMS key zu verwendenden Knoten an. Im Folgenden wird ein Beispiel gezeigt.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
    "s3BucketName": "DOC-EXAMPLE-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**
>
> Wenn Sie die Protokolldaten der Sitzung nicht verschlüsseln möchten, ändern Sie für `s3EncryptionEnabled` `true` in `false`.

Wenn Sie keine Protokolle an einen Amazon S3 S3-Bucket oder eine CloudWatch Logs-Protokollgruppe senden, aktive Sitzungsdaten nicht verschlüsseln oder die Unterstützung „Als ausführen“ für die Sitzungen in Ihrem Konto nicht aktivieren möchten, können Sie die Zeilen für diese Optionen löschen. Überprüfen Sie, dass die letzte Zeile im Abschnitt `inputs` nicht mit einem Komma endet.
Wenn Sie eine KMS-Schlüssel-ID zum Verschlüsseln Ihrer Sitzungsdaten hinzufügen, müssen sowohl die Benutzer, die die Sitzungen starten, als auch die verwalteten Knoten, mit denen sie sich verbinden, über die Berechtigung zur Verwendung des Schlüssels verfügen. Sie erteilen die Erlaubnis, den KMS-Schlüssel Session Manager mithilfe von AWS Identity and Access Management (IAM) -Richtlinien zu verwenden. Weitere Informationen finden Sie unter den folgenden Themen:

- Fügen Sie AWS KMS Berechtigungen für Benutzer in Ihrem Konto hinzu:Muster-IAM-Richtlinien für Session Manager.
- Fügen Sie AWS KMS Berechtigungen für verwaltete Knoten in Ihrem Konto hinzu:Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager.

3. Speichern Sie die Datei.
4. Führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, den folgenden Befehl aus.

   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'
```

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
    "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"
    }
}
```

Schritt 5: (Optional) Beschränken des Zugriffs auf Befehle in einer Sitzung

Sie können die Befehle einschränken, die ein Benutzer in einer AWS Systems Manager Session Manager Sitzung ausführen kann, indem Sie ein Dokument mit benutzerdefiniertem `Session` Typ AWS Systems Manager (SSM) verwenden. In dem Dokument definieren Sie den Befehl, der ausgeführt wird, wenn der Benutzer eine Sitzung startet, und die Parameter, die der Benutzer dem Befehl übergeben kann. Die `Session` des `schemaVersion`-Dokuments muss 1.0 und der `sessionType` des Dokuments muss `InteractiveCommands` lauten. Anschließend können

Sie AWS Identity and Access Management (IAM)-Richtlinien erstellen, die es den Benutzern ermöglichen, nur auf die von Ihnen definierten `Session`-Dokumente zuzugreifen. Weitere Informationen zur Verwendung von IAM-Richtlinien zum Beschränken des Zugriffs auf Befehle in einer Sitzung finden Sie unter IAM-Richtlinienbeispiele für interaktive Befehle.

Dokumente mit dem Zeichen `sessionType` von `InteractiveCommands` werden nur für Sitzungen unterstützt, die mit AWS Command Line Interface (AWS CLI) gestartet wurden. Der Benutzer gibt den Namen des benutzerdefinierten Dokuments als `--document-name`-Parameterwert an und gibt alle Befehlsparameterwerte über die Option `--parameters` an. Weitere Informationen zur Ausführung interaktiver Befehle finden Sie unter Starten einer Sitzung (interaktive und nicht interaktive Befehle).

Gehen Sie wie folgt vor, um ein SSM-Dokument vom benutzerdefinierten Typ `Session` zu erstellen, das den Befehl definiert, den ein Benutzer ausführen darf.

Beschränken des Zugriffs auf Befehle in einer Sitzung (Konsole)

So beschränken Sie die Befehle, die ein Benutzer in einer Session Manager-Sitzung ausführen kann (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie Create command or session (Befehl oder Sitzung erstellen) aus.
4. Geben Sie unter Name einen aussagekräftigen Namen für das Dokument ein.
5. Wählen Sie für Document type (Dokumenttyp) die Option Session document (Sitzungsdokument) aus.
6. Geben Sie mithilfe von JSON oder YAML Ihren Dokumentinhalt ein, der den Befehl definiert, den ein Benutzer in einer Session Manager-Sitzung ausführen kann, wie im folgenden Beispiel gezeigt.

   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. Wählen Sie Create document (Dokument erstellen) aus.

Beschränken des Zugriffs auf Befehle in einer Sitzung (Befehlszeile)

Bevor Sie beginnen

Falls Sie es noch nicht getan haben, installieren und konfigurieren Sie die AWS Command Line Interface (AWS CLI) oder die AWS Tools for PowerShell. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

So beschränken Sie die Befehle, die ein Benutzer in einer Session Manager-Sitzung ausführen kann (Befehlszeile)

1. Erstellen Sie eine JSON- oder YAML-Datei für Ihren Dokumentinhalt, der den Befehl definiert, den ein Benutzer in einer Session Manager-Sitzung ausführen kann, wie im folgenden Beispiel gezeigt.

   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. Führen Sie die folgenden Befehle aus, um ein SSM-Dokument unter Verwendung Ihres Inhalts zu erstellen, der den Befehl definiert, den ein Benutzer in einer Session Manager-Sitzung ausführen kann.

   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"
   ```

Interaktive Befehlsparameter und AWS CLI

Sie können interaktive Befehlsparameter bereitstellen, wenn Sie die AWS CLI verwenden. Je nach Betriebssystem (OS) Ihres Client-Computers, mit dem Sie eine Verbindung zu verwalteten Knoten herstellen, kann die Syntax AWS CLI, die Sie für Befehle angeben, die Sonder- oder Escape-Zeichen enthalten, unterschiedlich sein. Die folgenden Beispiele zeigen einige der verschiedenen Möglichkeiten, wie Sie Befehlsparameter angeben können, wenn Sie die verwenden AWS CLI, und wie mit Sonder- oder Escape-Zeichen umgegangen wird.

Auf Parameter, die in gespeichert sind, Parameter Store kann in den AWS CLI Befehlsparametern verwiesen werden, wie im folgenden Beispiel gezeigt.

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}}"]}'
```

Das folgende Beispiel zeigt, wie Sie mit der AWS CLI eine Kurzschriftsyntax verwenden, um Parameter zu übergeben.

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"
```

Sie können auch optionale Parameter in JSON angeben, wie im folgenden Beispiel dargestellt.

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"]}'
```

Parameter können auch in einer JSON-Datei gespeichert und für die bereitgestellt werden, AWS CLI wie im folgenden Beispiel gezeigt. Weitere Informationen zur Verwendung von AWS CLI -Parametern aus einer Datei finden Sie unter Laden von AWS CLI -Parametern aus einer Datei im AWS Command Line Interface -Benutzerhandbuch.

```
{
    "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
```

Sie können auch ein AWS CLI Skelett aus einer JSON-Eingabedatei generieren, wie im folgenden Beispiel gezeigt. Weitere Informationen zum Generieren von AWS CLI Skeletten aus JSON-

Eingabedateien finden Sie im Benutzerhandbuch unter Generieren von AWS CLI Skeletten und Eingabeparametern aus einer JSON- oder YAML-Eingabedatei.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
```

Um Zeichen in Anführungszeichen zu maskieren, müssen Sie den Escapezeichen zusätzliche umgekehrte Schrägstriche hinzufügen, wie im folgenden Beispiel gezeigt.

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\""]}'
```

Informationen zum Verwenden von Anführungszeichen bei Befehlsparametern in AWS CLI finden Sie unter Verwenden von Anführungszeichen mit Zeichenfolgen in der AWS CLI im AWS Command Line Interface -Benutzerhandbuch.

IAM-Richtlinienbeispiele für interaktive Befehle

Sie können IAM-Richtlinien erstellen, mit denen Benutzer nur auf die von Ihnen definierten `Session`-Dokumente zugreifen können. Dadurch werden die Befehle, die ein Benutzer in einer Session Manager-Sitzung ausführen kann, nur auf die Befehle beschränkt, die in Ihren benutzerdefinierten SSM-Dokumenten vom Typ `Session` definiert sind.

Einem Benutzer erlauben, einen interaktiven Befehl auf einem einzelnen verwalteten Knoten auszuführen

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ssm:StartSession",
         "Resource":[
            "arn:aws:ec2:region:987654321098:instance/i-02573cafcfEXAMPLE",
            "arn:aws:ssm:region:987654321098:document/exampleAllowedSessionDocument"
         ],
         "Condition":{
            "BoolIfExists":{
               "ssm:SessionDocumentAccessCheck":"true"
            }
         }
      }
   ]
}
```

Einem Benutzer erlauben, einen interaktiven Befehl auf allen verwalteten Knoten auszuführen

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ssm:StartSession",
         "Resource":[
```

```
            "arn:aws:ec2:us-west-2:987654321098:instance/*",
            "arn:aws:ssm:us-
west-2:987654321098:document/exampleAllowedSessionDocument"
         ],
         "Condition":{
            "BoolIfExists":{
               "ssm:SessionDocumentAccessCheck":"true"
            }
         }
      }
   ]
}
```

Einem Benutzer erlauben, mehrere interaktive Befehle auf allen verwalteten Knoten auszuführen

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ssm:StartSession",
         "Resource":[
            "arn:aws:ec2:us-west-2:987654321098:instance/*",
            "arn:aws:ssm:us-
west-2:987654321098:document/exampleAllowedSessionDocument",
            "arn:aws:ssm:us-
west-2:987654321098:document/exampleAllowedSessionDocument2"
         ],
         "Condition":{
            "BoolIfExists":{
               "ssm:SessionDocumentAccessCheck":"true"
            }
         }
      }
   ]
}
```

Schritt 6: (Optional) Verwenden von AWS PrivateLink zum Einrichten eines VPC-Endpunkts für Session Manager

Sie können den Sicherheitsstatus Ihrer verwalteten Knoten weiter verbessern, indem Sie AWS Systems Manager zum Verwenden eines Virtual Private Cloud (VPC)-Schnittstellenendpunkts

konfigurieren. Schnittstellenendpunkte werden von einer Technologie unterstützt AWS PrivateLink, mit der Sie über private IP-Adressen privat auf Amazon Elastic Compute Cloud (Amazon EC2) und Systems Manager APIs zugreifen können.

AWS PrivateLink schränkt den gesamten Netzwerkverkehr zwischen Ihren verwalteten Knoten, Systems Manager und Amazon EC2 auf das Amazon-Netzwerk ein. (Verwaltete Knoten haben keinen Zugriff auf das Internet.) Zudem benötigen Sie kein Internet-Gateway, kein NAT-Gerät und kein Virtual Private Gateway.

Informationen zum Erstellen eines VPC-Endpunkts finden Sie unter Verbessern der Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

Die Alternative zur Verwendung eines VPC-Endpunkts ist das Erlauben von ausgehendem Internetzugriff auf Ihre verwalteten Knoten. In diesem Fall müssen die verwalteten Knoten auch ausgehenden HTTPS-Datenverkehr (Port 443) zu den folgenden Endpunkten erlauben:

- `ec2messages.region.amazonaws.com`
- `ssm.region.amazonaws.com`
- `ssmmessages.region.amazonaws.com`

Systems Manager verwendet `ssmmessages.region.amazonaws.com`, den letzten dieser Endpunkte, über den Sie Anrufe von SSM Agent auf den Session Manager-Service in der Cloud tätigen können.

Um optionale Funktionen wie AWS Key Management Service (AWS KMS) -Verschlüsselung, das Streamen von Protokollen an Amazon CloudWatch Logs (CloudWatch Logs) und das Senden von Protokollen an Amazon Simple Storage Service (Amazon S3) zu nutzen, müssen Sie ausgehenden HTTPS-Verkehr (Port 443) zu den folgenden Endpunkten zulassen:

- `kms.region.amazonaws.com`
- `logs.region.amazonaws.com`
- `s3.region.amazonaws.com`

Weitere Informationen zu erforderlichen Endpunkten für Systems Manager finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.

Schritt 7: (Optional) Deaktivieren oder Aktivieren der Administratorberechtigungen für das SSM-Benutzerkonto

Ab Version 2.3.50.0 von AWS Systems Manager SSM Agent, erstellt der Agent ein lokales Benutzerkonto namens `ssm-user` und fügt es `/etc/sudoers` (Linux und macOS) bzw. der Administratorengruppe (Windows) hinzu. Auf Agenten-Versionen vor 2.3.612.0 wird das Konto beim ersten Start bzw. Neustart des SSM Agent nach der Installation erstellt. Auf Version 2.3.612.0 und höher wird das `ssm-user`-Konto beim ersten Start einer Sitzung auf einem Knoten erstellt. Dieser `ssm-user` ist der Standardbenutzer des Betriebssystems (OS), wenn eine AWS Systems Manager Session Manager-Sitzung gestartet wird. SSM Agent Version 2.3.612.0 wurde am 8. Mai 2019 veröffentlicht.

Wenn Sie verhindern möchten, dass Session Manager-Benutzer administrative Befehle auf einem Knoten ausführen, können Sie ihre `ssm-user`-Kontoberechtigungen aktualisieren. Sie können diese Berechtigungen wiederherstellen, nachdem sie entfernt wurden.

Themen

- Verwalten von sudo-Berechtigungen für ssm-user-Konten in Linux und macOS
- Verwalten von Administratorberechtigungen für das Konto ssm-user in Windows Server

Verwalten von sudo-Berechtigungen für ssm-user-Konten in Linux und macOS

Mit den folgenden Verfahren können Sie die sudo-Berechtigungen von ssm-Benutzerkonten auf Linux- und macOS-verwalteten Knoten aktivieren oder deaktivieren.

Verwenden von Run Command zum Ändern von sudo-Berechtigungen für ssm-user (Konsole)

- Verwenden Sie die Prozedur in Ausführen von Befehlen über die Konsole mit den folgenden Werten:
  - Wählen Sie unter Command document (Befehlsdokument) die Option `AWS-RunShellScript` aus.
  - Um den sudo-Zugriff zu entfernen, fügen Sie im Bereich Command parameters (Befehlsparameter) Folgendes in das Feld Commands (Befehle) ein.

```
cd /etc/sudoers.d
echo "#User rules for ssm-user" > ssm-agent-users
```

–oder–

Um den sudo-Zugriff wiederherzustellen, fügen Sie im Bereich Command parameters (Befehlsparameter) Folgendes in das Feld Commands (Befehle) ein.

```
cd /etc/sudoers.d
echo "ssm-user ALL=(ALL) NOPASSWD:ALL" > ssm-agent-users
```

Verwenden der Befehlszeile zum Ändern von sudo-Berechtigungen für ssm-user (AWS CLI)

1. Stellen Sie eine Verbindung zum verwalteten Knoten her und führen Sie den folgenden Befehl aus.

   ```
   sudo -s
   ```

2. Ändern Sie den Arbeitsordner mit dem folgenden Befehl.

   ```
   cd /etc/sudoers.d
   ```

3. Öffnen Sie die Datei mit dem Namen `ssm-agent-users`, um sie zu bearbeiten.
4. Um den sudo-Zugriff zu entfernen, löschen Sie die folgende Zeile.

   ```
   ssm-user ALL=(ALL) NOPASSWD:ALL
   ```

   –oder–

   Um den sudo-Zugriff wiederherzustellen, fügen Sie die folgende Zeile hinzu.

   ```
   ssm-user ALL=(ALL) NOPASSWD:ALL
   ```

5. Speichern Sie die Datei.

Verwalten von Administratorberechtigungen für das Konto ssm-user in Windows Server

Mit den folgenden Verfahren können Sie die Administratorberechtigungen von ssm-user-Konten auf von Windows Server verwalteten Knoten aktivieren oder deaktivieren.

Verwenden von Run Command zum Ändern von Administratorberechtigungen (Konsole)

- Verwenden Sie die Prozedur in Ausführen von Befehlen über die Konsole mit den folgenden Werten:

  Wählen Sie unter Command document (Befehlsdokument) die Option `AWS-RunPowerShellScript` aus.

  Um den administrativen Zugriff zu entfernen, fügen Sie im Bereich Command parameters (Befehlsparameter) Folgendes in das Feld Commands (Befehle) ein.

  ```
  net localgroup "Administrators" "ssm-user" /delete
  ```

  –oder–

  Um den administrativen Zugriff wiederherzustellen, fügen Sie im Bereich Command parameters (Befehlsparameter) Folgendes in das Feld Commands (Befehle) ein.

  ```
  net localgroup "Administrators" "ssm-user" /add
  ```

Verwenden des PowerShell- oder Eingabeaufforderungs-Fensters zum Ändern von Administratorberechtigungen

1. Stellen Sie eine Verbindung mit dem verwalteten Knoten her und öffnen Sie das PowerShell- oder Eingabeaufforderungsfenster.
2. Um den administrativen Zugriff zu entfernen, führen Sie den folgenden Befehl aus.

   ```
   net localgroup "Administrators" "ssm-user" /delete
   ```

   –oder–

   Um den administrativen Zugriff wiederherzustellen, führen Sie den folgenden Befehl aus.

   ```
   net localgroup "Administrators" "ssm-user" /add
   ```

Verwenden Sie die Windows-Konsole, um die Berechtigungen für Administratoren zu ändern

1. Stellen Sie eine Verbindung mit dem verwalteten Knoten her und öffnen Sie das PowerShell- oder Eingabeaufforderungsfenster.
2. Führen Sie in der Befehlszeile `lusrmgr.msc` aus, um die Konsole Local Users and Groups (Lokale Benutzer und Gruppen) zu öffnen.
3. Öffnen Sie das Verzeichnis Benutzer und dann ssm-user.
4. Führen Sie auf der Registerkarte Member Of (Mitglied von) einen der folgenden Schritte aus:

   - Um den administrativen Zugriff zu entfernen, wählen Sie Administrators (Administratoren) und dann Remove (Entfernen) aus.

     –oder–

     Um den administrativen Zugriff wiederherzustellen, geben Sie **`Administrators`** in das Textfeld ein und klicken dann auf Add (Hinzufügen).
5. Wählen Sie OK.

Schritt 8: (Optional) Erlauben und Steuern von Berechtigungen für SSH-Verbindungen über Session Manager

Sie können Benutzern in Ihrem Konto erlauben AWS-Konto , mithilfe von AWS Command Line Interface (AWS CLI) Secure Shell (SSH) -Verbindungen zu verwalteten Knoten herzustellen. AWS Systems Manager Session Manager Benutzer, die eine Verbindung über SSH herstellen, können auch mit dem Secure Copy Protocol (SCP) Dateien zwischen ihren lokalen Maschinen und verwalteten Knoten kopieren. Sie können diese Funktionalität verwenden, um eine Verbindung zu verwalteten Knoten herzustellen, ohne eingehende Ports öffnen oder Bastion-Hosts pflegen zu müssen.

Nachdem Sie SSH-Verbindungen zugelassen haben, können Sie AWS Identity and Access Management (IAM) -Richtlinien verwenden, um Benutzern, Gruppen oder Rollen das Herstellen von SSH-Verbindungen explizit zu gestatten oder zu verweigern. Session Manager

> **Note**
>
> Protokollieren ist für Session Manager-Sitzungen, die eine Verbindung über Port-Weiterleitung oder SSH herstellen, nicht verfügbar. Dies liegt daran, dass SSH alle

Sitzungsdaten verschlüsselt und Session Managernur als Tunnel für SSH-Verbindungen dient.

Themen

- Zulassen von SSH-Verbindungen für Session Manager
- Steuern von Benutzerberechtigungen für SSH-Verbindungen über Session Manager

Zulassen von SSH-Verbindungen für Session Manager

Gehen Sie wie folgt vor, um über Session Manager SSH-Verbindungen für einen verwalteten Knoten zuzulassen.

Um SSH-Verbindungen für Session Manager zuzulassen

1. Gehen Sie auf dem verwalteten Knoten, zu dem Sie SSH-Verbindungen erlauben möchten, wie folgt vor:

   - Stellen Sie sicher, dass SSH auf dem verwalteten Knoten ausgeführt wird. (Sie können eingehende Ports für den Knoten schließen.)
   - Stellen Sie sicher, dass SSM Agent Version 2.3.672.0 oder höher auf dem verwalteten Knoten installiert ist.

     Weitere Informationen zum Installieren oder Aktualisieren von SSM Agent auf einem verwalteten Knoten finden Sie in den folgenden Themen:

     - Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Windows Server.
     - Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux
     - Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für macOS
     - Wie installiert man das SSM Agent auf hybriden Windows-Knoten
     - Wie installiert man den SSM Agent auf Hybrid-Linux-Knoten

     > **Note**
     >
     > Um Session Manager mit On-Premises-Servern, Edge-Geräten und virtuellen Maschinen (VMs) zu verwenden, die Sie als verwaltete Knoten aktiviert haben,

müssen Sie das Advanced-Instances-Kontingent nutzen. Weitere Informationen über erweiterte Instances finden Sie unter Konfigurieren von Instance-Kontingenten.

2. Gehen Sie auf der lokalen Maschine, mit der Sie mit SSH eine Verbindung zu einem verwalteten Knoten herstellen möchten, wie folgt vor:

   - Stellen Sie sicher, dass Version 1.1.23.0 oder höher des Session Manager-Plug-in installiert ist.

     Weitere Informationen zur Installation des Session Manager-Plug-ins finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.

   - Aktualisieren Sie die SSH-Konfigurationsdatei so, dass ein Proxy-Befehl ausgeführt wird, der eine Session Manager-Sitzung startet und alle Daten über die Verbindung überträgt.

     Linux und macOS

     > **Tip**
     >
     > Die SSH-Konfigurationsdatei befindet sich in der Regel unter `~/.ssh/config`.

     Fügen Sie der Konfigurationsdatei auf dem lokalen Computer den folgenden Code hinzu.

     ```
     # SSH over Session Manager
     host i-* mi-*
         ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-StartSSHSession --parameters 'portNumber=%p'"
     ```

     Windows

     > **Tip**
     >
     > Die SSH-Konfigurationsdatei befindet sich in der Regel unter `C:\Users\<username>\.ssh\config`.

     Fügen Sie der Konfigurationsdatei auf dem lokalen Computer den folgenden Code hinzu.

     ```
     # 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"
```

- Erstellen ein Privacy-Enhanced-Mail-Zertifikat (eine PEM-Datei) oder mindestens einen öffentlichen Schlüssel, bzw. überprüfen Sie, ob sie darüber verfügen, die beim Herstellen von Verbindungen zu verwalteten Knoten verwendet werden sollen. Dies muss ein Schlüssel sein, der dem verwalteten Knoten bereits zugeordnet ist. Die Berechtigungen für Ihre private Schlüsseldatei müssen so festgelegt sein, dass nur Sie diese lesen können. Mit dem folgenden Befehl können Sie die Berechtigungen für Ihre private Schlüsseldatei so festlegen, dass nur Sie diese lesen können.

  ```
  chmod 400 <my-key-pair>.pem
  ```

  Für eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance kann dies beispielsweise die Schlüsselpaardatei sein, die Sie beim Erstellen der Instance erstellt oder ausgewählt haben. (Sie geben den Pfad zum Zertifikat oder Schlüssel als Teil des Befehls zum Starten einer Sitzung an. Informationen zum Starten einer Sitzung mithilfe von SSH finden Sie unter Starten einer Sitzung (SSH).)

Steuern von Benutzerberechtigungen für SSH-Verbindungen über Session Manager

Nachdem Sie SSH-Verbindungen über Session Manager auf einem verwalteten Knoten aktiviert haben, können Sie IAM-Richtlinien verwenden, um Benutzern, Gruppen oder Rollen zu erlauben oder zu verweigern, SSH-Verbindungen über Session Manager herzustellen.

So verwenden Sie eine IAM-Richtlinie, um SSH-Verbindungen über Session Manager zu erlauben

- Wählen Sie eine der folgenden Optionen aus:
  - Option 1: Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

    Wählen Sie im Navigationsbereich Policies (Richtlinien) aus und aktualisieren Sie dann die Berechtigungsrichtlinie für den Benutzer oder die Rolle, dem/der Sie die Berechtigung erteilen möchten, SSH-Verbindungen über Session Manager zu starten.

Fügen Sie beispielsweise das folgende Element der Schnellstart-Richtlinie hinzu, die Sie in Kurzeinführung in Endbenutzerrichtlinien für Session Manager erstellt haben. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:StartSession",
            "Resource": [
                "arn:aws:ec2:region:account-id:instance/instance-id",
                "arn:aws:ssm:*:*:document/AWS-StartSSHSession"
            ],
            "Condition": {
                "BoolIfExists": {
                    "ssm:SessionDocumentAccessCheck": "true"
                }
             }
        }
    ]
}
```

- Option 2: Hängen Sie mithilfe der, der oder der AWS Management Console AWS API eine Inline-Richtlinie an AWS CLI eine Benutzerrichtlinie an.

  Verwenden Sie die Methode Ihrer Wahl und fügen Sie die Richtlinienerklärung in Option 1 der Richtlinie für einen AWS Benutzer, eine Gruppe oder eine Rolle bei.

  Informationen finden Sie im Abschnitt Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch.

So verwenden Sie eine IAM-Richtlinie, um SSH-Verbindungen über Session Manager zu verweigern

- Wählen Sie eine der folgenden Optionen aus:

  - Option 1: Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/. Wählen Sie im Navigationsbereich Policies (Richtlinien) aus und aktualisieren Sie dann die Berechtigungsrichtlinie für den Benutzer oder die Rolle, die am Starten von Session Manager-Sitzungen gehindert werden soll.

Fügen Sie beispielsweise das folgende Element der Schnellstart-Richtlinie hinzu, die Sie in Kurzeinführung in Endbenutzerrichtlinien für Session Manager erstellt haben.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Deny",
            "Action": "ssm:StartSession",
            "Resource": "arn:aws:ssm:*:*:document/AWS-StartSSHSession"
        }
    ],
            "Condition": {
                "BoolIfExists": {
                    "ssm:SessionDocumentAccessCheck": "true"
                }
             }
}
```

- Option 2: Hängen Sie eine Inline-Richtlinie an eine Benutzerrichtlinie an AWS Management Console, indem Sie die AWS CLI, oder die AWS API verwenden.

  Verwenden Sie die Methode Ihrer Wahl und fügen Sie die Richtlinienerklärung in Option 1 der Richtlinie für einen AWS Benutzer, eine Gruppe oder eine Rolle bei.

  Informationen finden Sie im Abschnitt Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch.

Arbeiten mit Session Manager

Sie können die AWS Systems Manager-Konsole, die Amazon Elastic Compute Cloud (Amazon EC2)-Konsole oder die AWS Command Line Interface (AWS CLI) verwenden, um Sitzungen zu starten, die eine Verbindung zu den verwalteten Knoten herstellen, auf die Ihr Systemadministrator Ihnen mit AWS Identity and Access Management (IAM)-Richtlinien Zugriff gewährt hat. Abhängig von Ihren Berechtigungen können Sie auch Informationen zu Sitzungen anzeigen, noch nicht abgelaufene inaktive Sitzungen fortsetzen und Sitzungen beenden. Nachdem eine Sitzung eingerichtet wurde, ist sie nicht von der Dauer der IAM-Rollensitzung betroffen. Informationen zur Begrenzung der

Sitzungsdauer mit Session Manager, finden Sie unter Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen und Angeben der maximalen Sitzungsdauer.

Weitere Informationen zu Sitzungen finden Sie unter Was ist eine Sitzung?.

Themen

- Installieren des Session Manager-Plugins für die AWS CLI
- Starten einer Sitzung
- Beenden einer Sitzung
- Anzeigen des Sitzungsverlaufs

Installieren des Session Manager-Plugins für die AWS CLI

Um Session Manager-Sitzungen mit Ihren verwalteten Knoten mithilfe von AWS Command Line Interface (AWS CLI) zu initiieren, müssen Sie das Session Manager-Plug-in auf Ihrer lokalen Maschine installieren. Sie können das Plugin auf unterstützten Versionen von Microsoft Windows Server, macOS, Linux und installierenUbuntu Server.

> **Note**
>
> Um das Session Manager Plugin verwenden zu können, muss AWS CLI Version 1.16.12 oder höher auf Ihrem lokalen Computer installiert sein. Weitere Informationen finden Sie unter Installieren oder Aktualisierung auf die neueste Version von AWS Command Line Interface.

Themen

- Aktuelle Version und Versionsverlauf des Session Manager-Plugins
- Installieren Sie das Session Manager-Plug-in auf Windows
- Installieren Sie das Session Manager-Plug-in auf macOS
- Installieren Sie das Session Manager Plugin auf Amazon Linux 2 und Red Hat Enterprise Linux Distributionen
- Das Session Manager-Plugin in Debian Server und Ubuntu Server installieren
- Verifizieren der Session Manager-Plug-In-Installation
- Session Manager -Plugin auf GitHub
- (Optional) Aktivieren Sie die Session Manager-Plug-In-Protokollierung

Aktuelle Version und Versionsverlauf des Session Manager-Plugins

Auf Ihrem lokalen Computer muss eine unterstützte Version des Session Manager-Plugins ausgeführt werden. Die aktuelle unterstützte Mindestversion ist 1.1.17.0. Wenn Sie eine ältere Version ausführen, werden Ihre Session Manager-Operationen möglicherweise nicht erfolgreich abgeschlossen.

Um zu prüfen, ob Sie die neueste Version ausführen, führen Sie den folgenden Befehl in der AWS CLI aus.

> **Note**
>
> Der Befehl gibt nur dann Ergebnisse zurück, wenn sich das Plug-In im Standardinstallationsverzeichnis für Ihren Betriebssystemtypen befindet. Sie können die Version auch in der Datei `VERSION` überprüfen. Diese Datei befindet sich in dem Verzeichnis, in dem Sie das Plug-In installiert haben.

```
session-manager-plugin --version
```

In der folgenden Tabelle finden Sie alle Versionen des Session Manager-Plugins sowie die in den einzelnen Versionen enthaltenen Features und Erweiterungen.

Version	Datum der Veröffentlichung	Details
1,2,633,0	30. Mai 2024	Verbesserung: Das Dockerfile wurde aktualisiert, um ein Amazon Elastic Container Registry (Amazon ECR) -Image zu verwenden.
1.2.553.0	10. Januar 2024	Verbesserung: Aktualisierte aws-sdk-go und abhängige Golang-Pakete.
1.2.536.0	4. Dezember 2023	Verbesserung: Unterstützung für die Übergabe einer StartSessionAPI-Antwort als Umgebungsvariable an hinzugefügt. session-manager-plugin
1.2.497.0	1. August 2023	Verbesserung: Go SDK wurde auf v1.44.302 aktualisiert.

Version	Datum der Veröffentlichung	Details
1,2,463,0	15. März 2023	Verbesserung: Mac with Apple silicon Unterstützung für Apple Mac (M1) im macOS-Bundle-Installer und im signierten Installer hinzugefügt.
1.2.398.0	14. Oktober 2022	Verbesserung: Unterstützung für Golang-Version 1.17. Aktualisieren Sie den session-manager-plugin Standard-Runner für macOS, um Python3 zu verwenden. Aktualisieren Sie den Importpfad von SSMCLI auf. session-manager-plugin
1.2.339.0	16. Juni 2022	Fehlerbehebung: Behebt die Zeitbeschränkung der Leerlaufs itzung für Portsitzungen.
1.2.331,0	27. Mai 2022	Fehlerbehebung: Behebt Portsitzungen, die vorzeitig geschlossen werden, wenn der lokale Server vor der Zeitbeschränkung keine Verbindung herstellt.
1.2.323,0	19. Mai 2022	Fehlerbehebung: Deaktivieren Sie Smux Keep Alive, um das Timeout-Feature im Leerlauf zu verwenden.
1.2.312,0	31. März 2022	Verbesserung: Unterstützt mehr Payload-Typen für Ausgabenachrichten.
1.2.295,0	12. Januar 2022	Fehlerbehebung: Aufgehängte Sitzungen durch das erneute Senden von Stream-Daten des Clients, wenn der Agent inaktiv wird, und falsche Protokolle für die Nachrichten `start_publication` und `pause_publication` .
1.2.279,0	27. Oktober 2021	Verbesserung: ZIP-Paketierung für Windows-Plattform.
1.2.245,0	19. August 2021	Verbesserung: Aktualisieren von `aws-sdk-go` auf die neueste Version (v1.40.17), um AWS IAM Identity Center zu unterstützen.
1,2,234,0	26. Juli 2021	Fehlerbehebung: Abrupt abgebrochenes Szenario im interakti ven Sitzungstyp behandeln.

Version	Datum der Veröffentlichung	Details
1.2.205,0	10. Juni 2021	Verbesserung: Unterstützung für signiertes macOS-Installationsprogramm.
1.2.54,0	29. Januar 2021	Verbesserung: Unterstützung für das Ausführen von Sitzungen im NonInteractiveCommands Ausführungsmodus hinzugefügt.
1.2.30.0	24. November 2020	Verbesserung: (Nur Port-Weiterleitungssitzungen) Verbesserte Gesamtleistung.
1.2.7.0	15. Oktober 2020	Verbesserung: (Nur Port-Weiterleitungssitzungen) Verringerte Latenz und verbesserte Gesamtleistung.
1.1.61.0	17. April 2020	Verbesserung: ARM-Unterstützung für Linux und Ubuntu hinzugefügt.
1.1.54.0	6. Januar 2020	Fehlerbehebung: Verarbeitung des Race-Bedingungsszenarios von Paketen, die gelöscht werden, wenn das Session Manager-Plug-In nicht bereit ist.
1.1.50.0	19. November 2019	Erweiterung: Unterstützung für die Weiterleitung eines Ports an einen lokalen Unix-Socket hinzugefügt.
1.1.35.0	7. November 2019	Verbesserung: (Nur Portweiterleitungssitzungen) Sendet einen TerminateSession Befehl an, SSM Agent wenn der lokale Benutzer drückt. `Ctrl+C`
1.1.33.0	26. September 2019	Erweiterung: (Nur Port-Weiterleitungssitzungen) Senden Sie ein Trennsignal an den Server, wenn der Client die TCP-Verbindung trennt.
1.1.31.0	6. September 2019	Erweiterung: Aktualisieren Sie, um die Port-Weiterleitungssitzung offen zu halten, bis der Remote-Server die Verbindung schließt.

Version	Datum der Veröffentlichung	Details
1.1.26.0	30. Juli 2019	Erweiterung: Begrenzung der Datenübertragungsrate während einer Sitzung aktualisiert.
1.1.23.0	9. Juli 2019	Verbesserung: Unterstützung für die Ausführung von SSH-Sitzungen mit Session Manager hinzugefügt.
1.1.17.0	4. April 2019	Erweiterung: Unterstützung für die zusätzliche Verschlüs selung von Sitzungsdaten mit AWS Key Management Service (AWS KMS) hinzugefügt.
1.0.37.0	20. September 2018	Verbesserung: Fehlerbehebung für Windows-Version.
1.0.0.0	11. September 2018	Erstveröffentlichung des Session Manager-Plugins.

Installieren Sie das Session Manager-Plug-in auf Windows

Sie können das Session Manager-Plug-In auf Windows Vista oder höher mit dem eigenständigen Installationsprogramm installieren.

Wenn Aktualisierungen veröffentlicht werden, müssen Sie die Installation wiederholen, um die aktuelle Version des Session Manager-Plugins zu erhalten.

> **Note**
>
> Um optimale Ergebnisse zu erzielen, sollten Sie Sitzungen auf Windows-Clients mittels Windows PowerShell-Version 5 oder höher starten. Alternativ hierzu können Sie auch die Befehls-Shell in Windows 10 verwenden. Das Session Manager-Plug-In unterstützt nur PowerShell und die Befehls-Shell. Die Befehlszeilentools von Drittanbietern sind möglicherweise nicht mit dem Plug-In kompatibel.

So installieren Sie das Session Manager-Plug-In mithilfe des EXE-Installationsprogramms

1. Laden Sie das Installationsprogramm über die folgende URL herunter.

```
https://s3.amazonaws.com/session-manager-downloads/plugin/latest/windows/
SessionManagerPluginSetup.exe
```

Alternativ können Sie eine gezippte Version des Installationsprogramms mit der folgenden URL herunterladen.

```
https://s3.amazonaws.com/session-manager-downloads/plugin/latest/windows/
SessionManagerPlugin.zip
```

2. Führen Sie das heruntergeladene Installationsprogramm aus und folgen Sie den Anweisungen auf dem Bildschirm. Wenn Sie die gezippte Version des Installationsprogramms heruntergeladen haben, müssen Sie zuerst das Installationsprogramm entpacken.

   Lassen Sie das Feld „Installationsspeicherort“ leer, um das Plug-In im Standardverzeichnis zu installieren.

   - `%PROGRAMFILES%\Amazon\SessionManagerPlugin\bin\`

3. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plug-In-Installation.

> **Note**
>
> Wenn Windows die ausführbare Datei nicht finden kann, müssen Sie die Eingabeaufforderung möglicherweise erneut öffnen oder das Installationsverzeichnis der Umgebungsvariablen PATH manuell hinzufügen. Informationen finden Sie im Fehlerbehebungsthema Session Manager-Plugin wurde nicht automatisch zumBefehlszeilenpfad hinzugefügt (Windows).

Installieren Sie das Session Manager-Plug-in auf macOS

Wählen Sie eines der folgenden Themen aus, unter dem das Session Manager-Plugin auf macOS installiert werden soll. Das mitgelieferte Installationsprogramm verwendet eine ZIP-Datei. Nach dem Entpacken können Sie das Plugin mithilfe der Binärdatei installieren. Das signierte Installationsprogramm ist eine signierte .pkg-Datei.

Themen

- Installieren Sie das Session Manager-Plug-in auf macOS

- Installieren des Session Manager-Plugins in macOS mittels des signierten Installationsprogramms

Installieren Sie das Session Manager-Plug-in auf macOS

In diesem Abschnitt wird beschrieben, wie Sie das Session Manager-Plugin mithilfe des mitgelieferten Installers auf macOS installieren.

> **Important**
>
> Das gebündelte Installationsprogramm unterstützt keine Installation in Pfaden, die Leerzeichen enthalten.

Sie installieren Sie das Session Manager-Plug-In mittels des Paketinstallationsprogramms (macOS)

1. Laden Sie das Paketinstallationsprogramm herunter.

   x86_64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/mac/
   sessionmanager-bundle.zip" -o "sessionmanager-bundle.zip"
   ```

   Mac mit Apple Silicon

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/
   mac_arm64/sessionmanager-bundle.zip" -o "sessionmanager-bundle.zip"
   ```

2. Entpacken Sie das Paket.

   ```
   unzip sessionmanager-bundle.zip
   ```

3. Führen Sie den Installationsbefehl aus.

   ```
   sudo ./sessionmanager-bundle/install -i /usr/local/sessionmanagerplugin -b /usr/
   local/bin/session-manager-plugin
   ```

   > **Note**
   >
   > Das Plug-In erfordert Python 2.6.5 oder höher oder Python 3.3 oder höher. Standardmäßig wird das Installationsskript unter der Standard-Systemversion von

Python ausgeführt. Wenn Sie eine alternative Version von Python installiert haben und diese zum Installieren des Session Manager-Plugins verwenden möchten, führen Sie das Installationsskript mit dieser Version aus, indem Sie den absoluten Pfad der ausführbaren Python-Datei verwenden. Im Folgenden wird ein Beispiel gezeigt.

```
sudo /usr/local/bin/python3.8 sessionmanager-bundle/install -i /usr/local/
sessionmanagerplugin -b /usr/local/bin/session-manager-plugin
```

Das Installationsprogramm installiert das Session Manager-Plug-In in `/usr/local/sessionmanagerplugin` und erstellt den Symlink `session-manager-plugin` im Verzeichnis `/usr/local/bin`. Dies beseitigt die Notwendigkeit, das Installationsverzeichnis in der $PATH-Variablen des Benutzers anzugeben.

Eine Erklärung der Optionen `-i` und `-b` sehen Sie, indem Sie die Option `-h` verwenden.

```
./sessionmanager-bundle/install -h
```

4. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plug-In-Installation.

> **Note**
>
> Um das Plugin zu deinstallieren, führen Sie die folgenden beiden Befehle in der angegebenen Reihenfolge aus.
>
> ```
> sudo rm -rf /usr/local/sessionmanagerplugin
> ```
>
> ```
> sudo rm /usr/local/bin/session-manager-plugin
> ```

Installieren des Session Manager-Plugins in macOS mittels des signierten Installationsprogramms

In diesem Abschnitt wird beschrieben, wie Sie das Session Manager-Plugin mithilfe des signierten Installers auf macOS installieren.

So installieren Sie das Session Manager-Plug-In mittels des signierten Installationsprogramms (macOS)

1. Laden Sie das signierte Installationsprogramm herunter.

   x86_64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/mac/
   session-manager-plugin.pkg" -o "session-manager-plugin.pkg"
   ```

   Mac mit Apple Silicon

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/
   mac_arm64/session-manager-plugin.pkg" -o "session-manager-plugin.pkg"
   ```

2. Führen Sie die Installationsbefehle aus.

   ```
   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. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plug-In-Installation.

Installieren Sie das Session Manager Plugin auf Amazon Linux 2 und Red Hat Enterprise Linux Distributionen

Verwenden Sie das folgende Verfahren, um das Session Manager-Plugin auf RHEL-Distributionen zu installieren.

> **Note**
>
> Das Session Manager Plugin wird auf Amazon Linux 1 nicht unterstützt. Es wird unterstützt unter Amazon Linux 2 und höher.

1. Laden Sie das Session Manager-Plug-In-RPM-Paket herunter und installieren Sie es.

x86_64

Führen Sie auf RHEL 7 den folgenden Befehl aus:

```
sudo yum install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_64bit/session-manager-plugin.rpm
```

Führen Sie auf RHEL 8 und 9 den folgenden Befehl aus:

```
sudo dnf install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_64bit/session-manager-plugin.rpm
```

86 x

Führen Sie auf RHEL 7 den folgenden Befehl aus:

```
sudo yum install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_32bit/session-manager-plugin.rpm
```

Führen Sie auf RHEL 8 und 9 den folgenden Befehl aus:

```
sudo dnf install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_32bit/session-manager-plugin.rpm
```

ARM64

Führen Sie auf RHEL 7 den folgenden Befehl aus:

```
sudo yum install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_arm64/session-manager-plugin.rpm
```

Führen Sie auf RHEL 8 und 9 den folgenden Befehl aus:

```
sudo dnf install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
latest/linux_arm64/session-manager-plugin.rpm
```

2. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plug-In-Installation.

> **Note**
>
> Wenn Sie das Plug-In jemals deinstallieren möchten, führen Sie `sudo yum erase session-manager-plugin -y` aus.

Das Session Manager-Plugin in Debian Server und Ubuntu Server installieren

1. Laden Sie das Session Manager-Plug-In-deb-Paket herunter.

   x86_64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/
   ubuntu_64bit/session-manager-plugin.deb" -o "session-manager-plugin.deb"
   ```

   86 x

   ```
   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. Führen Sie den Installationsbefehl aus.

   ```
   sudo dpkg -i session-manager-plugin.deb
   ```

3. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plug-In-Installation.

> **Note**
>
> Wenn Sie das Plug-In jemals deinstallieren möchten, führen Sie `sudo dpkg -r session-manager-plugin` aus.

Verifizieren der Session Manager-Plug-In-Installation

Führen Sie die folgenden Befehle aus, um zu überprüfen, ob das Session Manager-Plug-In erfolgreich installiert wurde.

```
session-manager-plugin
```

Wenn die Installation erfolgreich war, wird die folgende Meldung zurückgegeben.

```
The Session Manager plugin is installed successfully. Use the AWS CLI to start a
 session.
```

Sie können die Installation auch testen, indem Sie den start-sessionBefehl in der AWS Command Line Interface(AWS CLI) ausführen. Ersetzen Sie im folgenden Befehl *`instance-id`* mit Ihren eigenen Informationen.

```
aws ssm start-session --target instance-id
```

Dieser Befehl funktioniert nur AWS CLI, wenn Sie den installiert und konfiguriert haben und wenn Ihr Session Manager Administrator Ihnen die erforderlichen IAM-Berechtigungen für den Zugriff auf den verwalteten Zielknoten erteilt hat. Session Manager

Session Manager -Plugin auf GitHub

Der Quellcode für das Session Manager Plugin ist auf verfügbar, GitHub sodass Sie das Plugin an Ihre Anforderungen anpassen können. Wir möchten Sie bitten, uns eventuelle Änderungswünsche mitzuteilen. Amazon Web Services bietet jedoch keine Unterstützung für die Ausführung modifizierter Kopien dieser Software.

(Optional) Aktivieren Sie die Session Manager-Plug-In-Protokollierung

Das Session Manager-Plug-In enthält eine Option zum Aktivieren der Protokollierung für von Ihnen ausgeführte Sitzungen. Standardmäßig ist die Protokollierung deaktiviert.

Wenn Sie die Protokollierung aktivieren, erstellt das Session Manager-Plug-In Protokolldateien sowohl für Anwendungsaktivitäten (`session-manager-plugin.log`) als auch für Fehler (`errors.log`) auf Ihrem lokalen Computer.

Themen

- Aktivieren der Protokollierung für das Session Manager-Plug-In (Windows)

- Aktivieren der Protokollierung für das Session Manager-Plug-In (Linux und macOS)

Aktivieren der Protokollierung für das Session Manager-Plug-In (Windows)

1. Suchen Sie die Datei `seelog.xml.template` für das Plug-In.

   Der Standardspeicherort ist `C:\Program Files\Amazon\SessionManagerPlugin\seelog.xml.template`.
2. Ändern Sie den Namen der Datei in `seelog.xml`.
3. Öffnen Sie die Datei und ändern Sie `minlevel="off"` in `minlevel="info"` oder `minlevel="debug"`.

   > **Note**
   >
   > Standardmäßig werden Protokolleinträge zum Öffnen eines Datenkanals und Neuverbinden von Sitzungen auf INFO-Ebene aufgezeichnet. Datenflusseinträge (Pakete und Bestätigung) werden auf DEBUG-Ebene aufgezeichnet.
4. Ändern Sie andere Konfigurationsoptionen, die Sie ändern möchten. Optionen, die Sie ändern können, sind:

   - Debug-Ebene: Sie können die Debug-Ebene von `formatid="fmtinfo"` in `formatid="fmtdebug"` ändern.
   - DieProtokolldateioptionen: Sie können die Protokolldateioptionen einschließlich des Speicherorts der Protokolle ändern, jedoch nicht die Namen von Protokolldateien.

     > **Important**
     >
     > Sie dürfen die Dateinamen nicht ändern. Wenn Sie dies tun, funktioniert die Protokollierung nicht korrekt.

     ```
     <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. Speichern Sie die Datei.

Aktivieren der Protokollierung für das Session Manager-Plug-In (Linux und macOS)

1. Suchen Sie die Datei `seelog.xml.template` für das Plug-In.

   Der Standardspeicherort ist `/usr/local/sessionmanagerplugin/seelog.xml.template`.

2. Ändern Sie den Namen der Datei in `seelog.xml`.
3. Öffnen Sie die Datei und ändern Sie `minlevel="off"` in `minlevel="info"` oder `minlevel="debug"`.

   > **Note**
   >
   > Standardmäßig werden Protokolleinträge zum Öffnen von Datenkanälen und Neuverbinden von Sitzungen auf INFO-Ebene aufgezeichnet. Datenflusseinträge (Pakete und Bestätigung) werden auf DEBUG-Ebene aufgezeichnet.

4. Ändern Sie andere Konfigurationsoptionen, die Sie ändern möchten. Optionen, die Sie ändern können, sind:

   - Debug-Ebene: Sie können die Debug-Ebene von `formatid="fmtinfo"` in `outputs formatid="fmtdebug"` ändern.
   - DieProtokolldateioptionen: Sie können die Protokolldateioptionen einschließlich des Speicherorts der Protokolle ändern, jedoch nicht die Namen von Protokolldateien.

     > **Important**
     >
     > Sie dürfen die Dateinamen nicht ändern. Wenn Sie dies tun, funktioniert die Protokollierung nicht korrekt.

```
<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**
>
> Wenn Sie das angegebene Standardverzeichnis für das Speichern von Protokollen verwenden, müssen Sie Sitzungsbefehle entweder über sudo ausführen oder dem Verzeichnis, in dem das Plug-In installiert ist, vollständige Lese- und Schreibberechtigungen erteilen. Um diese Einschränkungen zu umgehen, ändern Sie den Speicherort, an dem die Protokolle gespeichert werden.

5. Speichern Sie die Datei.

Starten einer Sitzung

Sie können die AWS Systems Manager Konsole, die Amazon Elastic Compute Cloud (Amazon EC2) -Konsole, die AWS Command Line Interface (AWS CLI) oder SSH verwenden, um eine Sitzung zu starten.

Themen

- Starten einer Sitzung (Systems Manager-Konsole)
- Starten einer Sitzung (Amazon EC2-Konsole)
- Starten einer Sitzung (AWS CLI)
- Starten einer Sitzung (SSH)
- Starten einer Sitzung (Port-Weiterleitung)
- Starten einer Sitzung (Port-Weiterleitung an entfernten Host)
- Starten einer Sitzung (interaktive und nicht interaktive Befehle)

Starten einer Sitzung (Systems Manager-Konsole)

Sie können die AWS Systems Manager Konsole verwenden, um eine Sitzung mit einem verwalteten Knoten in Ihrem Konto zu starten.

> **Note**
>
> Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.

Starten einer Sitzung (Systems-Manager-Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie Start session (Sitzung starten) aus.
4. (Optional) Geben Sie eine Beschreibung für die Sitzung im Feld Grund für die Sitzung ein.
5. Aktivieren Sie unter Ziel-Instances das Optionsfeld links neben dem verwalteten Knoten aus, mit dem Sie eine Verbindung herstellen möchten.

   Wenn der gewünschte Knoten nicht in der Liste enthalten ist oder wenn Sie einen Knoten auswählen und einen Konfigurationsfehler erhalten, lesen Sie die Schritte zur Fehlerbehebung unter Verwalteter Knoten ist nicht verfügbar oder nicht für Session Manager konfiguriert.
6. Wählen Sie Sitzung starten, um die Sitzung sofort zu starten.

   –oder–

   Wählen Sie Weiter für die Sitzungsoptionen.
7. (Optional) Wählen Sie unter Sitzungsdokument das Dokument aus, das Sie zu Beginn der Sitzung ausführen möchten. Wenn Ihr Dokument Laufzeitparameter unterstützt, können Sie in jedes Parameterfeld einen oder mehrere durch Komma getrennte Werte eingeben.
8. Wählen Sie Weiter aus.
9. Wählen Sie Start session (Sitzung starten) aus.

Nach der Herstellung der Verbindung können Sie Bash-Befehle (Linux und macOS) oder PowerShell-Befehle (Windows) wie für jeden anderen Verbindungstyp ausführen.

> **Important**
>
> Wenn Sie Benutzern die Möglichkeit geben möchten, beim Starten von Sitzungen in der Session-Manager-Konsole ein Dokument anzugeben, beachten Sie Folgendes:
>
> - Sie müssen Benutzern die in ihrer IAM-Richtlinie festgelegten Berechtigungen `ssm:GetDocument` und `ssm:ListDocuments` gewähren. Weitere Informationen finden Sie unter Zugriff auf benutzerdefinierte Sitzungsdokumente in der Konsole gewähren.

- Die Konsole unterstützt nur Sitzungsdokumente, für die der `sessionType` als `Standard_Stream` definiert ist. Weitere Informationen finden Sie unter Schema des Sitzungsdokuments.

Starten einer Sitzung (Amazon EC2-Konsole)

Sie können die Amazon Elastic Compute Cloud (Amazon EC2)-Konsole verwenden, um eine Sitzung mit einer Instance in Ihrem Konto zu starten.

> **Note**
>
> Wenn Sie den Fehler erhalten, dass Sie nicht berechtigt sind, eine oder mehrere Systems Manager (`ssm:command-name`)-Aktionen auszuführen, müssen Sie sich an Ihren Administrator wenden. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt. Bitten Sie diese Person, Ihre Richtlinien zu aktualisieren, damit Sie Sitzungen von der Amazon EC2-Konsole aus starten können. Wenn Sie ein Administrator sind, finden Sie weitere Informationen unter Muster-IAM-Richtlinien für Session Manager.

Starten einer Sitzung (Amazon EC2-Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die Instance und Connect (Verbinden) aus.
4. Für Connection method (Verbindungsmethode) wählen Sie Session Manager.
5. Wählen Sie Connect aus.

Nach der Herstellung der Verbindung können Sie Bash-Befehle (Linux und macOS) oder PowerShell-Befehle (Windows) wie für jeden anderen Verbindungstyp ausführen.

Starten einer Sitzung (AWS CLI)

Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.

Um die Befehle AWS CLI to run session verwenden zu können, muss das Session Manager Plugin auch auf Ihrem lokalen Computer installiert sein. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.

Um eine Sitzung mit dem zu starten AWS CLI, führen Sie den folgenden Befehl aus und ersetzen Sie *instance-id* durch Ihre eigenen Informationen.

```
aws ssm start-session \
    --target instance-id
```

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Starten einer Sitzung (SSH)

Um eine Session Manager-SSH-Sitzung zu starten, muss Version 2.3.672.0 oder höher von SSM Agent auf dem verwalteten Knoten installiert sein.

Anforderungen für SSH-Verbindungen

Beachten Sie die folgenden Anforderungen und Einschränkungen für Sitzungsverbindungen mit SSH:

- Ihr anvisierter verwalteter Knoten muss so konfiguriert sein, dass er SSH-Verbindungen unterstützt. Weitere Informationen finden Sie unter (Optional) Zulassen und Steuern von Berechtigungen für SSH-Verbindungen über Session Manager.
- Sie müssen mithilfe des Kontos des verwalteten Knotens verbinden, der dem Privacy Enhanced Mail (PEM)-Zertifikat zugeordnet ist, und nicht dem `ssm-user`-Konto, das für andere Arten von Sitzungsverbindungen verwendet wird. Auf EC2-Instances für Linux und macOS, ist beispielsweise der Standardbenutzer `ec2-user`. Informationen zur Identifizierung des Standardbenutzers für jeden Instance-Typ finden Sie unter Get Information About Your Instance im Amazon EC2 EC2-Benutzerhandbuch.
- Protokollieren ist für Session Manager-Sitzungen, die eine Verbindung über Port-Weiterleitung oder SSH herstellen, nicht verfügbar. Dies liegt daran, dass SSH alle Sitzungsdaten verschlüsselt und Session Managernur als Tunnel für SSH-Verbindungen dient.

> **Note**
>
> Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.

Um eine Sitzung über SSH zu starten, führen Sie folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
ssh -i /path/my-key-pair.pem username@instance-id
```

> **Tip**
>
> Wenn Sie eine Sitzung mit SSH starten, können Sie mit dem folgenden Befehl lokale Dateien auf den anvisierten verwalteten Knoten kopieren.
>
> ```
> scp -i /path/my-key-pair.pem /path/ExampleFile.txt username@instance-id:~
> ```

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Starten einer Sitzung (Port-Weiterleitung)

Um eine Session Manager-Port-Weiterleitungs-Sitzung zu starten, muss Version 2.3.672.0 oder höher des SSM Agent auf dem verwalteten Knoten installiert sein.

> **Note**
>
> Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.
> Um die Befehle AWS CLI to run session verwenden zu können, müssen Sie das Session Manager Plug-in auf Ihrem lokalen Computer installieren. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.

Je nach Betriebssystem und Befehlszeilentool kann die Platzierung von Anführungszeichen unterschiedlich sein, und möglicherweise sind Escapezeichen erforderlich.

Um eine Port-Weiterleitungssitzung zu starten, führen Sie den folgenden Befehl in der CLI aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

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`ist der Remote-Port auf dem verwalteten Knoten, an den der Sitzungsverkehr umgeleitet werden soll. Sie können beispielsweise Port `3389` für die Verbindung zu einem Windows-Knoten mithilfe des Remote Desktop Protocol (RDP) angeben. Wenn Sie den `portNumber`-Parameter nicht angeben, verwendet Session Manager die `80`-Standardschulungsparameter.

`localPortNumber`ist der Port auf Ihrem lokalen Computer, an dem der Verkehr beginnt, z. `56789` B. Dieser Wert ist, was Sie eingeben, wenn Sie eine Verbindung mit einem verwalteten Knoten über einen Client herstellen. z. B. **`localhost:56789`**.

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Weitere Informationen zu Port-Weiterleitungssitzungen finden Sie unter Port-Weiterleitung unter Verwendung von AWS Systems ManagerSession Manager im AWS News Blog.

Starten einer Sitzung (Port-Weiterleitung an entfernten Host)

Um eine Session Manager-Port-Weiterleitungs-Sitzung zu einem entfernten Host zu starten, muss Version 3.1.1374.0 oder höher des SSM Agent auf dem verwalteten Knoten installiert sein. Der Remote-Host muss nicht von Systems Manager verwaltet werden.

> **Note**
>
> Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.
> Um die Befehle AWS CLI to run session verwenden zu können, müssen Sie das Session Manager Plug-in auf Ihrem lokalen Computer installieren. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.
> Je nach Betriebssystem und Befehlszeilentool kann die Platzierung von Anführungszeichen unterschiedlich sein, und möglicherweise sind Escapezeichen erforderlich.

Um eine Portweiterleitungssitzung zu starten, führen Sie den folgenden Befehl vom aus aus AWS CLI. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

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"
```

Der `host`-Wert stellt den Hostnamen oder die IP-Adresse des Remote-Hosts dar, zu dem Sie eine Verbindung herstellen möchten. Es gelten weiterhin allgemeine Anforderungen an Konnektivität und Namensauflösung zwischen dem verwalteten Knoten und dem Remote-Host.

`portNumber`ist der Remote-Port auf dem verwalteten Knoten, an den der Sitzungsverkehr umgeleitet werden soll. Sie können beispielsweise Port `3389` für die Verbindung zu einem Windows-Knoten mithilfe des Remote Desktop Protocol (RDP) angeben. Wenn Sie den `portNumber`-Parameter nicht angeben, verwendet Session Manager die `80`-Standardschulungsparameter.

`localPortNumber`ist der Port auf Ihrem lokalen Computer, an dem der Verkehr beginnt, z. `56789` B. Dieser Wert ist, was Sie eingeben, wenn Sie eine Verbindung mit einem verwalteten Knoten über einen Client herstellen. z. B. **`localhost:56789`**.

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Eine Sitzung mit einer Amazon ECS-Aufgabe starten

Session Managerunterstützt das Starten einer Portweiterleitungssitzung mit einer Aufgabe innerhalb eines Amazon Elastic Container Service (Amazon ECS) -Clusters. Dazu müssen Sie die Aufgabenrolle in IAM so aktualisieren, dass sie die folgenden Berechtigungen enthält:

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
       "Effect": "Allow",
       "Action": [
            "ssmmessages:CreateControlChannel",
            "ssmmessages:CreateDataChannel",
            "ssmmessages:OpenControlChannel",
            "ssmmessages:OpenDataChannel"
       ],
      "Resource": "*"
      }
   ]
}
```

Um eine Portweiterleitungssitzung mit einer Amazon ECS-Aufgabe zu starten, führen Sie den folgenden Befehl von der aus AWS CLI. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

> **Note**
>
> Entfernen Sie die < and > Symbole aus dem `target` Parameter. Diese Symbole dienen nur zur Verdeutlichung des Lesers.

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"
```

Starten einer Sitzung (interaktive und nicht interaktive Befehle)

Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.

Um die Befehle AWS CLI to run session verwenden zu können, muss das Session Manager Plugin auch auf Ihrem lokalen Computer installiert sein. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.

Um eine Interactive Befehls-Sitzung zu starten, führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

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"
```

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Weitere Informationen

- Verwenden Sie Port-Forwarding in AWS Systems ManagerSession Manager, um eine Verbindung zu Remote-Hosts herzustellen
- Amazon EC2 EC2-Instance-Portweiterleitung mit AWS Systems Manager
- AWS Verwaltete Microsoft AD-Ressourcen mit Session Manager Portweiterleitung verwalten
- Port-Weiterleitung mit AWS Systems ManagerSession Manager im AWS News Blog.

Beenden einer Sitzung

Sie können mithilfe der AWS Systems Manager-Konsole oder der AWS Command Line Interface (AWS CLI) eine Sitzung beenden, die Sie in Ihrem Konto gestartet haben. Wenn es nach 20 Minuten keine Benutzeraktivitäten gegeben hat, wird eine Sitzung beendet. Einmal beendete Sitzungen können nicht mehr fortgesetzt werden.

Themen

- So beenden Sie eine Sitzung (Konsole)
- Beenden einer Sitzung (AWS CLI)

So beenden Sie eine Sitzung (Konsole)

Sie können mithilfe der AWS Systems Manager-Konsole eine Sitzung in Ihrem Konto beenden.

So beenden Sie eine Sitzung (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie unter Sessions (Sitzungen) die Optionsschaltfläche links neben der Sitzung aus, die Sie beenden möchten.
4. Wähen Sie Beenden.

Beenden einer Sitzung (AWS CLI)

Um eine Sitzung über die AWS CLI zu beenden, führen Sie folgenden Befehl aus. Ersetzen Sie *session-id* mit Ihren eigenen Informationen.

```
aws ssm terminate-session \
    --session-id session-id
```

Weitere Informationen zum Befehl terminate-session finden Sie unter terminate-session im Abschnitt AWS Systems Manager der Befehlsreferenz AWS CLI.

Anzeigen des Sitzungsverlaufs

Sie können die AWS Systems Manager-Konsole oder die AWS Command Line Interface (AWS CLI) verwenden, um Informationen zu Sitzungen in Ihrem Konto anzuzeigen. In der Konsole können Sie Sitzungsdetails wie die folgenden anzeigen:

- Die ID der Sitzung
- Welche Benutzer über eine Sitzung eine Verbindung mit einem verwalteten Knoten hergestellt haben
- Die ID des verwalteten Knotens
- Wann die Sitzung gestartet und beendet wurde
- Den Status der Sitzung
- Den für das Speichern von Sitzungsprotokollen angegebenen Speicherort (wenn aktiviert)

Über die AWS CLI können Sie eine Liste der Sessions in Ihrem Konto anzeigen, jedoch nicht die zusätzlichen Details, die in der Konsole verfügbar sind.

Informationen zur Protokollierung des Sitzungsverlaufs finden Sie unter Protokollierung von Sitzungsaktivitäten aktivieren und deaktivieren.

Themen

- Anzeigen des Sitzungsverlaufs (Konsole)
- Anzeigen des Sitzungsverlaufs (AWS CLI)

Anzeigen des Sitzungsverlaufs (Konsole)

Sie können die AWS Systems Manager-Konsole verwenden, um Details zu den Sitzungen in Ihrem Konto anzuzeigen.

So zeigen Sie den Sitzungsverlauf an (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Session history (Sitzungsverlauf) aus.

   –oder–

   Wenn die Session Manager-Startseite zum ersten Mal geöffnet wird, wählen Sie Einstellungen konfigurieren und dann die Registerkarte Sitzungsverlauf aus.

Anzeigen des Sitzungsverlaufs (AWS CLI)

Um eine Liste der Sitzungen in Ihrem Konto über die AWS CLI anzuzeigen, führen Sie folgenden Befehl aus.

```
aws ssm describe-sessions \
    --state History
```

> **Note**
>
> Dieser Befehl gibt nur Ergebnisse für Verbindungen zu Zielen zurück, die mit Session Manager initiiert wurden. Es werden keine Verbindungen aufgeführt, die auf andere Weise hergestellt wurden, z. B. Remote Desktop Protocol (RDP) oder Secure Shell Protocol (SSH).

Informationen zu anderen Optionen, die Sie mit dem Befehl describe-sessions verwenden können, finden Sie unter describe-sessions im Abschnitt AWS Systems Manager der AWS CLI-Command Reference.

Prüfen von Sitzungsaktivitäten

Zusätzlich zur Bereitstellung von Informationen zu den aktuellen und abgeschlossenen Sitzungen in der Systems Manager-Konsole stellt Session Manager Ihnen Optionen für die Prüfung von Sitzungsaktivitäten in Ihrem AWS-Konto mit AWS CloudTrail bereit.

CloudTrail erfasst Session-API-Aufrufe über die Systems Manager-Konsole, das AWS Command Line Interface (AWS CLI) und das Systems Manager SDK. Sie können die Informationen auf der CloudTrail Konsole anzeigen oder sie in einem bestimmten Amazon Simple Storage Service (Amazon S3) -Bucket speichern. Ein Amazon S3 S3-Bucket wird für alle CloudTrail Protokolle Ihres Kontos verwendet. Weitere Informationen finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.

> **Note**
>
> Für wiederkehrende, historische, analytische Analysen Ihrer Protokolldateien sollten Sie erwägen, CloudTrail Protokolle mithilfe von CloudTrail Lake oder einer von Ihnen verwalteten Tabelle abzufragen. Weitere Informationen finden Sie unter AWS CloudTrail Logs abfragen im AWS CloudTrail Benutzerhandbuch.

Überwachung der Sitzungsaktivität mit Amazon EventBridge (Konsole)

Mit können Sie Regeln einrichten EventBridge, um zu erkennen, wann Änderungen an AWS Ressourcen vorgenommen werden. Sie können eine Regel erstellen, um zu erkennen, wenn ein Benutzer in Ihrer Organisation eine Sitzung startet oder beendet, und dann z. B. über Amazon SNS eine Benachrichtigung bezüglich des Ereignisses erhalten.

EventBridge Die Unterstützung für Session Manager stützt sich auf Aufzeichnungen von API-Vorgängen, die von aufgezeichnet wurden CloudTrail. (Sie können die CloudTrail Integration mit verwenden EventBridge , um auf die meisten AWS Systems Manager Ereignisse zu reagieren.) Aktionen, die innerhalb einer Sitzung stattfinden, z. B. ein `exit` Befehl, der keinen API-Aufruf durchführt, werden von nicht erkannt EventBridge.

Die folgenden Schritte zeigen, wie Sie Benachrichtigungen über Amazon Simple Notification Service (Amazon SNS) initiieren, wenn ein Session ManagerAPI-Ereignis eintritt, z. B. StartSession.

Um die Sitzungsaktivität mit Amazon EventBridge (Konsole) zu überwachen

1. Erstellen Sie ein Amazon SNS-Thema zum Senden von Benachrichtigungen, wenn das Session Manager-Ereignis eintritt, die Sie überwachen möchten.

   Weitere Informationen finden Sie unter Erstellen eines Themas im Amazon Simple Notification Service-Entwicklerhandbuch.
2. Erstellen Sie eine EventBridge Regel, um das Amazon SNS SNS-Ziel für den Session Manager Ereignistyp aufzurufen, den Sie verfolgen möchten.

   Informationen zur Erstellung der Regel finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erstellen von EventBridge Amazon-Regeln, die auf Ereignisse reagieren.

   Wählen Sie während der Erstellung der Regel die folgenden Optionen aus:

   - Wählen Sie für AWS service (-Service), die Option Systems Manager aus.
   - Wählen Sie als Ereignistyp die Option AWS API Call through aus CloudTrail.
   - Wählen Sie Specific operation (s) (Spezifische Operation(en)) aus und geben Sie dann nacheinander die Session Manager-Befehle ein, für die Sie Benachrichtigungen erhalten möchten. Sie können StartSession, ResumeSession und TerminateSession auswählen. (unterstützt EventBridge keine `Describe*` Befehle `Get* List*`, und.)
   - Für Select a target (Wählen Sie ein Ziel aus), wählen Sie SNS-Thema aus. Wählen Sie unter Topic (Thema) den Namen des von Ihnen in Schritt 1 erstellten Amazon SNS-Themas aus.

Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch und im Amazon Simple Notification Service Getting Started Guide.

Protokollierung von Sitzungsaktivitäten aktivieren und deaktivieren

Zusätzlich zur Bereitstellung von Informationen zu den aktuellen und abgeschlossenen Sitzungen in der Systems Manager-Konsole stellt Session Manager Ihnen Optionen für die Protokollierung von Sitzungsaktivitäten in Ihrem AWS-Konto bereit. Damit können Sie Folgendes tun:

- Erstellen und Speichern von Sitzungsprotokollen zu Archivierungszwecken.
- Generieren eines Berichts mit Details zu jeder Verbindung, die mit Ihren verwalteten Knoten über Session Manager in den letzten 30 Tagen hergestellt wurde.
- Generieren Sie Benachrichtigungen über Sitzungsaktivitäten in Ihren AWS-Konto, z. B. Amazon Simple Notification Service (Amazon SNS) -Benachrichtigungen.

- Initiieren Sie automatisch eine weitere Aktion für eine AWS Ressource als Ergebnis einer Sitzungsaktivität, z. B. das Ausführen einer AWS Lambda Funktion, das Starten einer AWS CodePipeline Pipeline oder das Ausführen eines AWS Systems Manager Run Command Dokuments.

Important

Beachten Sie die folgenden Anforderungen und Einschränkungen für Session Manager:

- Session Manager protokolliert die von Ihnen eingegebenen Befehle und deren Ausgabe während einer Sitzung abhängig von Ihren Sitzungseinstellungen. Um zu verhindern, dass vertrauliche Daten wie Passwörter in Ihren Sitzungsprotokollen angezeigt werden, empfehlen wir die folgenden Befehle, wenn Sie während einer Sitzung vertrauliche Daten eingeben.

  Linux & macOS

  ```
  stty -echo; read passwd; stty echo;
  ```

  Windows

  ```
  $Passwd = Read-Host -AsSecureString
  ```

- Wenn Sie Windows Server 2012 oder früher verwenden, werden die Daten in Ihren Protokollen möglicherweise nicht optimal formatiert. Wir empfehlen die Verwendung von Windows Server 2012 R2 und höher, um optimal formatierte Protokolle zu erhalten.
- Wenn Sie Linux- oder macOS-verwaltete Knoten verwenden, muss das Screen-Serviceprogramm installiert sein. Wenn dies nicht der Fall ist, werden die Protokolldaten möglicherweise abgeschnitten. Auf Amazon Linux 1, Amazon Linux 2, AL2023 und Ubuntu Server ist das Screen Utility standardmäßig installiert. Um Screen manuell zu installieren, müssen Sie abhängig von Ihrer Linux-Version entweder `sudo yum install screen` oder `sudo apt-get install screen` ausführen.
- Protokollieren ist für Session Manager-Sitzungen, die eine Verbindung über Port-Weiterleitung oder SSH herstellen, nicht verfügbar. Dies liegt daran, dass SSH alle Sitzungsdaten verschlüsselt und Session Managernur als Tunnel für SSH-Verbindungen dient.

Weitere Informationen zu den Berechtigungen, die für die Verwendung von Amazon S3 oder Amazon CloudWatch Logs für die Protokollierung von Sitzungsdaten erforderlich sind, finden Sie unterErstellen einer IAM-Rolle mit Berechtigungen für Amazon S3 Session Manager und CloudWatch Logs (Konsole).

Weitere Informationen zu Protokollierungsoptionen für Session Manager finden Sie in den folgenden Themen.

Themen

- Streaming-Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)
- Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)
- Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)
- Deaktivieren der Session Manager Aktivitätsprotokollierung in CloudWatch Logs und Amazon S3

Streaming-Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

Sie können einen kontinuierlichen Stream von Sitzungsdatenprotokollen an Amazon CloudWatch Logs senden. Wichtige Details, wie die Befehle, die ein Benutzer in einer Sitzung ausgeführt hat, die ID des Benutzers, der die Befehle ausgeführt hat, und Zeitstempel, wann die Sitzungsdaten in CloudWatch Logs gestreamt werden, sind beim Streamen von Sitzungsdaten enthalten. Beim Streamen von Sitzungsdaten werden die Protokolle JSON-formatiert, um Ihnen bei der Integration in Ihre vorhandenen Protokollierungslösungen zu helfen. Streaming-Sitzungsdaten werden für interaktive Befehle nicht unterstützt.

> **Note**
>
> Um Sitzungsdaten von Windows Server-verwalteten Knoten zu streamen, müssen Sie PowerShell 5.1 oder höher installiert haben. Standardmäßig ist bei Windows Server 2016 und höher die erforderliche PowerShell-Version installiert. Bei Windows Server 2012 und 2012 R2 ist die erforderliche PowerShell-Version jedoch nicht standardmäßig installiert. Wenn Sie PowerShell noch nicht auf Ihren von Windows Server 2012 oder 2012 R2 verwalteten Knoten aktualisiert haben, können Sie dies mit Run Command tun. Informationen zur Aktualisierung von PowerShell mithilfe von Run Command finden Sie unter Aktualisierung PowerShell mit Run Command.

> **Important**
>
> Wenn Sie die Einstellung der PowerShell Transkriptionsrichtlinie auf Ihren Windows Server verwalteten Knoten konfiguriert haben, können Sie keine Sitzungsdaten streamen.

Um Sitzungsdaten mit Amazon CloudWatch Logs zu streamen (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Aktivieren Sie unter CloudWatch Protokollierung das Kontrollkästchen neben Aktivieren.
5. Wählen Sie die Option Sitzungsungsprotokolle streamen aus.
6. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Nur verschlüsselte CloudWatch Protokollgruppen zulassen. Wenn diese Funktion aktiviert ist, werden die Protokolldaten mithilfe des serverseitigen Verschlüsselungsschlüssels, der für die Protokollgruppe angegeben wurde, verschlüsselt. Wenn Sie die Protokolldaten, die an CloudWatch Logs gesendet werden, nicht verschlüsseln möchten, deaktivieren Sie das Kontrollkästchen. Außerdem müssen Sie das Kontrollkästchen deaktivieren, wenn die Verschlüsselung für die Protokollgruppe nicht aktiviert ist.
7. Wählen Sie für CloudWatch Protokolle eine der folgenden Optionen aus, AWS-Konto um die bestehende CloudWatch Protokollgruppe Logs in die Sie die Sitzungsprotokolle hochladen möchten, anzugeben:

   - Geben Sie den Namen einer bereits in Ihrem Konto erstellten Protokollgruppe in das Textfeld ein, um die Sitzungsprotokolldaten zu speichern.
   - Protokollgruppen durchsuchen: Wählen Sie eine bereits in Ihrem Konto erstellte Protokollgruppe aus, um die Sitzungsprotokolldaten zu speichern.
8. Wählen Sie Speichern.

Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)

Sie können Sitzungsprotokolldaten zu Debugging- und Fehlerbehebungszwecken in einem angegebenen Amazon Simple Storage Service (Amazon S3)-Bucket speichern. Standardmäßig

werden Protokolle an einen verschlüsselten Amazon S3-Bucket gesendet. Die Verschlüsselung erfolgt mit dem für den Bucket angegebenen Schlüssel, entweder einem AWS KMS key oder einem Amazon S3 S3-Schlüssel für serverseitige Verschlüsselung (SSE) (AES-256).

> **Important**
>
> Bei Verwendung von Buckets im Stil eines virtuellen Hostings mit SSL (Secure Sockets Layer) stimmt das SSL-Wildcard-Zertifikat nur mit Buckets überein, die keine Punkte enthalten. Um dies zu umgehen, verwenden Sie HTTP oder schreiben Sie Ihre eigene Logik zur Verifizierung von Zertifikaten. Wir empfehlen, bei der Verwendung von Buckets im Stil des virtuellen Hostings keine Punkte („.“) in Bucket-Namen zu verwenden.

Verschlüsselung von Amazon S3-Bucket

Um Protokolle mit Verschlüsselung an Ihren Amazon S3-Bucket senden zu können, muss die Verschlüsselungsfunktion für den Bucket aktiviert sein. Weitere Informationen zur Amazon S3 Bucket-Verschlüsselung finden Sie unter Amazon S3-Standardverschlüsselung für S3-Buckets.

Kundenverwalteter Schlüssel

Wenn Sie zum Verschlüsseln Ihres Buckets einen KMS-Schlüssel verwenden, den Sie selbst verwalten, muss das Ihren Instances angefügte IAM-Instance-Profil explizite Berechtigungen zum Lesen des Schlüssels besitzen. Wenn Sie einen verwenden Von AWS verwalteter Schlüssel, benötigt die Instance diese ausdrückliche Genehmigung nicht. Weitere Informationen zum Bereitstellen des Instance-Profils mit Zugriff auf die Verwendung des Schlüssels finden Sie unter Gestattet Schlüsselbenutzern die Verwendung des Schlüssels im AWS Key Management Service - Entwicklerhandbuch.

Gehen Sie wie folgt vor, um Session Manager zum Speichern von Sitzungsprotokollen im Amazon S3-Bucket zu konfigurieren.

> **Note**
>
> Sie können den auch verwenden AWS CLI , um den Amazon S3 S3-Bucket anzugeben oder zu ändern, an den Sitzungsdaten gesendet werden. Weitere Informationen finden Sie unter Aktualisieren von Session Manager-Einstellungen (Befehlszeile).

Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Wählen Sie bei S3-Protokollierung neben Aktivieren das Kontrollkästchen aus.
5. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Nur verschlüsselte S3-Buckets zulassen. Wenn diese Funktion aktiviert ist, werden die Protokolldaten mithilfe des serverseitigen Verschlüsselungsschlüssels, der für den Bucket angegeben wurde, verschlüsselt. Wenn Sie die an Amazon S3 gesendeten Protokolldaten nicht verschlüsseln möchten, aktivieren Sie das Kontrollkästchen. Außerdem müssen Sie das Kontrollkästchen deaktivieren, wenn die Verschlüsselung für den S3-Bucket nicht aktiviert ist.
6. Wählen Sie in S3 bucket name (Name des S3-Buckets) eine der folgenden Optionen aus:

   > **Note**
   >
   > Wir empfehlen, bei der Verwendung von Buckets im Stil des virtuellen Hostings keine Punkte („.“) in Bucket-Namen zu verwenden. Weitere Informationen zu Namenskonventionen für Amazon-S3-Buckets finden Sie unter Bucket-Einschränkungen und -Limits im Benutzerhandbuch zu Amazon Simple Storage Service.

   - Choose a bucket name from the list (Bucket-Namen aus der Liste auswählen): Wählen Sie einen bereits in Ihrem Konto erstellten Amazon S3-Bucket aus, um Sitzungsprotokolldaten zu speichern.
   - Enter a bucket name in the text box (Geben Sie einen Namen für den Bucket in das Textfeld ein): Geben Sie den Namen eines bereits in Ihrem Konto erstellten Amazon S3-Buckets ein, um Sitzungsprotokolldaten zu speichern.
7. (Optional) Geben Sie in S3 key prefix (S3-Schlüsselpräfix) den Namen eines vorhandenen oder neuen Ordners ein, in dem die Protokolle im ausgewählten Bucket gespeichert werden sollen.
8. Wählen Sie Speichern.

Weitere Informationen zum Arbeiten mit Amazon S3 und Amazon-S3-Buckets finden Sie im Benutzerhandbuch zu Amazon Simple Storage Service und im Benutzerhandbuch zu Amazon Simple Storage Service.

Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

Mit Amazon CloudWatch Logs können Sie Protokolldateien aus verschiedenen Quellen überwachen, speichern und darauf zugreifen AWS-Services. Sie können Sitzungsprotokolldaten zu Debugging- und Fehlerbehebungszwecken an eine CloudWatch Logs-Protokollgruppe senden. Standardmäßig werden Protokolldaten nach Verschlüsselung mit Ihrem KMS-Schlüssel gesendet. Sie können die Daten jedoch mit oder ohne Verschlüsselung an ihre Protokollgruppe senden.

Gehen Sie wie folgt vor, um AWS Systems Manager Session Manager zu konfigurieren, dass Sitzungsprotokolldaten am Ende Ihrer Sitzungen an eine CloudWatch Logs-Protokollgruppe gesendet werden.

> **Note**
>
> Sie können auch AWS CLI die CloudWatch Logs-Protokollgruppe angeben oder ändern, an die Sitzungsdaten gesendet werden. Weitere Informationen finden Sie unter Aktualisieren von Session Manager-Einstellungen (Befehlszeile).

So protokollieren Sie Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Aktivieren Sie unter CloudWatch Protokollierung das Kontrollkästchen neben Aktivieren.
5. Wählen Sie die Option Sitzungsungsprotokolle hochladen aus.
6. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Nur verschlüsselte CloudWatch Protokollgruppen zulassen. Wenn diese Funktion aktiviert ist, werden die Protokolldaten mithilfe des serverseitigen Verschlüsselungsschlüssels, der für die Protokollgruppe angegeben wurde, verschlüsselt. Wenn Sie die Protokolldaten, die an CloudWatch Logs gesendet werden, nicht verschlüsseln möchten, deaktivieren Sie das Kontrollkästchen. Außerdem müssen Sie das

Kontrollkästchen deaktivieren, wenn die Verschlüsselung für die Protokollgruppe nicht aktiviert ist.

7. Wählen Sie für CloudWatch Protokolle eine der folgenden Optionen aus, AWS-Konto um die bestehende CloudWatch Protokollgruppe Logs in die Sie die Sitzungsprotokolle hochladen möchten, anzugeben:

   - Choose a log group from the list (Eine Protokollgruppe aus der Liste auswählen): Wählen Sie eine bereits in Ihrem Konto erstellte Protokollgruppe aus, in die die Sitzungsprotokolldaten gespeichert werden sollen.
   - Enter a log group name in the text box (Geben Sie den Namen einer Protokollgruppe in das Textfeld ein): Geben Sie den Namen einer bereits in Ihrem Konto erstellten Protokollgruppe ein, in die die Sitzungsprotokolldaten gespeichert werden sollen.

8. Wählen Sie Speichern.

Weitere Informationen zur Arbeit mit CloudWatch Logs finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

Deaktivieren der Session Manager Aktivitätsprotokollierung in CloudWatch Logs und Amazon S3

Sie können die Systems Manager Manager-Konsole verwenden oder AWS CLI die Protokollierung der Sitzungsaktivitäten in Ihrem Konto deaktivieren.

Um die Protokollierung von Sitzungsaktivitäten zu deaktivieren (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Um die CloudWatch Protokollierung zu deaktivieren, deaktivieren CloudWatch Sie im Abschnitt Protokollierung das Kontrollkästchen Aktivieren.
5. Um die S3-Protokollierung zu deaktivieren, deaktivieren Sie im Abschnitt S3-Protokollierung das Kontrollkästchen Aktivieren.
6. Wählen Sie Speichern.

Um die Protokollierung der Sitzungsaktivitäten zu deaktivieren (AWS CLI)

Um die Protokollierung von Sitzungsaktivitäten mithilfe von zu deaktivieren AWS CLI, folgen Sie den Anweisungen unterAktualisieren von Session Manager-Einstellungen (Befehlszeile).

Stellen Sie in Ihrer JSON-Datei sicher, dass die Eingaben `s3BucketName` und `cloudWatchLogGroupName` keine Werte enthalten. Beispielsweise:

```
"inputs": {
        "s3BucketName": "",
        ...
        "cloudWatchLogGroupName": "",
        ...
    }
```

Alternativ können Sie alle `S3*` und `cloudWatch*` Eingaben aus Ihrer JSON-Datei entfernen, um die Protokollierung zu deaktivieren.

Schema des Sitzungsdokuments

Die folgenden Informationen beschreiben die Schemaelemente eines Session-Dokuments. AWS Systems Manager Session Managerverwendet Sitzungsdokumente, um zu bestimmen, welcher Sitzungstyp gestartet werden soll, z. B. eine Standardsitzung, eine Portweiterleitungssitzung oder eine Sitzung zur Ausführung eines interaktiven Befehls.

schemaVersion

Die Schemaversion des Sitzungsdokuments. Sitzungsdokumente unterstützen nur die Version 1.0.

Typ: Zeichenfolge

Erforderlich: Ja

description

Eine Beschreibung, die Sie für das Sitzungsdokument angeben. Beispiel: „Dokument zum Starten der Port-Weiterleitungssitzung mit Session Manager“.

Typ: Zeichenfolge

Erforderlich: Nein

sessionType

Der Sitzungstyp, mit dem das Sitzungsdokument erstellt wird.

Typ: Zeichenfolge

Erforderlich: Ja

Zulässige Werte: `InteractiveCommands` | `NonInteractiveCommands` | `Port` | `Standard_Stream`

inputs

Die Sitzungseinstellungen, die für Sitzungen verwendet werden, die mit diesem Sitzungsdokument erstellt wurden. Dieses Element ist für Sitzungsdokumente erforderlich, die zum Erstellen von `Standard_Stream`-Sitzungen verwendet werden.

Typ: StringMap

Erforderlich: Nein

s3BucketName

Der Amazon Simple Storage Service (Amazon S3)-Bucket, an den Sie am Ende Ihrer Sitzungen Sitzungsprotokolle senden möchten.

Typ: Zeichenfolge

Erforderlich: Nein

s3KeyPrefix

Das Präfix, das beim Senden von Protokollen an den Amazon S3-Bucket verwendet werden soll, den Sie in der `s3BucketName`-Eingabe angegeben haben. Weitere Hinweise zur Verwendung eines freigegebenen Präfixes für in Amazon S3 gespeicherte Objekte finden Sie unter Wie kann ich Ordner in einem S3-Bucket verwenden? im Benutzerhandbuch zu Amazon Simple Storage Service.

Typ: Zeichenfolge

Erforderlich: Nein

s3EncryptionEnabled

Wenn auf `true` eingestellt ist, muss der Amazon S3-Bucket, den Sie in der `s3BucketName`-Eingabe angegeben haben, verschlüsselt werden.

Typ: Boolesch

Erforderlich: Ja

cloudWatchLogGroupName

Der Name der Amazon CloudWatch Logs-Gruppe (CloudWatch Logs), an die Sie am Ende Ihrer Sitzungen Sitzungsprotokolle senden möchten.

Typ: Zeichenfolge

Erforderlich: Nein

cloudWatchEncryptionEnabled

Wenn auf `true` eingestellt ist, muss die Protokollgruppe, die Sie in der `cloudWatchLogGroupName`-Eingabe angegeben haben, verschlüsselt werden.

Typ: Boolesch

Erforderlich: Ja

cloudWatchStreamingEnabled

Wenn auf `true` eingestellt ist, wird ein kontinuierlicher Stream von Sitzungsdaten an die Protokollgruppe gesendet, die Sie in der `cloudWatchLogGroupName`-Eingabe angegeben haben. Wenn auf `false` eingestellt ist, werden Sitzungsprotokolle am Ende Ihrer Sitzungen an die Protokollgruppe gesendet, die Sie in der `cloudWatchLogGroupName`-Eingabe angegeben haben.

Typ: Boolesch

Erforderlich: Ja

kmsKeyId

Die ID der, die AWS KMS key Sie verwenden möchten, um Daten zwischen Ihren lokalen Client-Computern und den von Amazon Elastic Compute Cloud (Amazon EC2) verwalteten Knoten, mit denen Sie eine Verbindung herstellen, weiter zu verschlüsseln.

Typ: Zeichenfolge

Erforderlich: Nein

runAsEnabled

Wenn auf `true` eingestellt ist, müssen Sie in der `runAsDefaultUser`-Eingabe ein Benutzerkonto angeben, das auf den verwalteten Knoten vorhanden ist, mit denen Sie eine Verbindung herstellen möchten. Andernfalls können Sitzungen nicht gestartet werden. Standardmäßig werden Sitzungen mit dem von AWS Systems Manager SSM Agent erstellten `ssm-user`-Konto gestartet. Das Run-As-Feature wird nur für die Verbindung mit Linux-verwalteten Knoten unterstützt.

Typ: Boolesch

Erforderlich: Ja

runAsDefaultUser

Der Name des Benutzerkontos, mit dem Sitzungen auf Linux-verwalteten Knoten gestartet werden sollen, wenn die `runAsEnabled`-Eingabe auf `true` eingestellt ist. Das Benutzerkonto, das Sie für diese Eingabe angeben, muss auf den verwalteten Knoten vorhanden sein, mit denen Sie eine Verbindung herstellen möchten. Andernfalls können Sitzungen nicht gestartet werden.

Typ: Zeichenfolge

Erforderlich: Nein

idleSessionTimeout

Die Zeit der Inaktivität, die Sie zulassen möchten, bevor eine Sitzung beendet wird. Diese Eingabe wird in Minuten gemessen.

Typ: Zeichenfolge

Zulässige Werte: 1 bis 60

Erforderlich: Nein

maxSessionDuration

Die maximale Zeit, die Sie erlauben möchten, bevor eine Sitzung beendet wird. Diese Eingabe wird in Minuten gemessen.

Typ: Zeichenfolge

Zulässige Werte: 1–1 440

Erforderlich: Nein

shellProfile

Die Einstellungen, die Sie pro Betriebssystem angeben und die in Sitzungen angewendet werden, wie Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnissen und das Ausführen mehrerer Befehle.

Typ: StringMap

Erforderlich: Nein

windows

Die Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnisse und Befehle, die Sie für Sitzungen auf Windows-verwalteten Knoten angeben.

Typ: Zeichenfolge

Erforderlich: Nein

linux

Die Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnisse und Befehle, die Sie für Sitzungen auf Linux-verwalteten Knoten angeben.

Typ: Zeichenfolge

Erforderlich: Nein

parameters

Ein Objekt, das die Parameter definiert, die das Dokument akzeptiert. Weitere Informationen zum Definieren von Dokumentparametern finden Sie unter Parameter im Top-Level-Datenelemente. Wir empfehlen, Parameter, auf die Sie häufig verweisen, im Systems Manager Parameter Store abzuspeichern und dann auf sie zu verweisen. In diesem Abschnitt des Dokuments können Sie die Parameter Store-Parameter `String` und `StringList` referenzieren. In diesem Abschnitt des Dokuments können Sie die Parameter Store-Parameter `SecureString` nicht referenzieren. Sie können über das folgende Format einen Parameter Store-Parameter referenzieren.

```
{{ssm:parameter-name}}
```

Mehr über Parameter Store erfahren Sie unter AWS Systems Manager Parameter Store.

Typ: StringMap

Erforderlich: Nein

properties

Ein Objekt, dessen Werte Sie angeben, die in der `StartSession` API-Operation verwendet werden.

Für Sitzungsdokumente, die für `InteractiveCommands`-Sitzungen verwendet werden,enthält das Eigenschaftenobjekt die Befehle, die auf den von Ihnen angegebenen Betriebssystemen ausgeführt werden sollen. Mit der `runAsElevated` booleschen Eigenschaft können Sie auch festlegen, ob Befehle als `root` ausgeführt werden. Weitere Informationen finden Sie unter Zugriff auf Befehle in einer Sitzung beschränken.

Für Sitzungsdokumente, die für `Port`-Sitzungen verwendet werden, enthält das Eigenschaftenobjekt die Portnummer, an die der Datenverkehr umgeleitet werden soll. Ein Beispiel ist das Beispiel zum Sitzungsdokument des Typs `Port` an späterer Stelle in diesem Thema.

Typ: StringMap

Erforderlich: Nein

Beispiel für Sitzungsdokument vom Typ `Standard_Stream`

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"
        }
    }
}
```

Beispiel für Sitzungsdokument vom Typ `InteractiveCommands`

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
        }
    }
}
```

Beispiel für Sitzungsdokument vom Typ `Port`

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"
    }
}
```

Beispiel für Sitzungsdokument mit Sonderzeichen

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
      }
   }
}
```

Fehlerbehebung für Session Manager

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit AWS Systems Manager Session Manager.

Themen

- Session Manager kann von der Amazon-EC2-Konsole aus keine Verbindung herstellen
- Keine Berechtigung zum Starten einer Sitzung
- Keine Berechtigung zum Ändern von Sitzungspräferenzen
- Verwalteter Knoten ist nicht verfügbar oder nicht für Session Manager konfiguriert
- Session Manager-Plug-In nicht gefunden
- Session Manager-Plugin wurde nicht automatisch zumBefehlszeilenpfad hinzugefügt (Windows)
- Session Manager-Plug-In reagiert nicht mehr
- TargetNotVerbunden
- Nach dem Starten einer Sitzung wird ein leerer Bildschirm angezeigt
- Verwalteter Knoten reagiert während langer Sitzungen nicht mehr

- Beim Aufrufen des Vorgangs ist ein Fehler aufgetreten (InvalidDocument) StartSession

Session Manager kann von der Amazon-EC2-Konsole aus keine Verbindung herstellen

Problem: Nach der Erstellung einer neuen Instance bietet Ihnen die Registerkarte Session Manager in der Konsole von Amazon Elastic Compute Cloud (Amazon EC2) nicht die Möglichkeit, sich zu verbinden.

Lösung A: Erstellen Sie ein Instanzprofil: Falls Sie dies noch nicht getan haben (wie in den Informationen auf der Registerkarte „Session Manager“ in der EC2-Konsole beschrieben), erstellen Sie ein AWS Identity and Access Management (IAM-) Instanzprofil mithilfe von. Quick Setup Quick Setupist eine Fähigkeit von. AWS Systems Manager

Session Manager erfordert ein IAM-Instance-Profil, um eine Verbindung zu Ihrer Instance herzustellen. Sie können ein Instance-Profil erstellen und es Ihrer Instance zuweisen, indem Sie eine Host-Verwaltungs-Konfiguration mit Quick Setup erstellen. Eine Host-Verwaltungs-Konfiguration erstellt ein Instance-Profil mit den erforderlichen Berechtigungen und weist es Ihrer Instance zu. Eine Host-Verwaltungs-Konfiguration ermöglicht auch andere Systems-Manager-Funktionen und erstellt IAM-Rollen für die Ausführung dieser Funktionen. Die Nutzung von Quick Setup oder der Funktionen, die durch die Host-Verwaltungs-Konfiguration aktiviert werden, ist kostenlos. Öffnen Sie Quick Setup und erstellen Sie eine Host-Verwaltungs-Konfiguration.

> **Important**
>
> Nachdem Sie die Host-Verwaltungs-Konfiguration erstellt haben, kann es einige Minuten dauern, bis Amazon EC2 die Änderung registriert und die Registerkarte Session Manager aktualisiert hat. Wenn auf der Registerkarte nach zwei Minuten keine Verbindungsschaltfläche angezeigt wird, starten Sie Ihre Instance neu. Wenn Sie nach dem Neustart immer noch keine Verbindungsoption sehen, öffnen Sie Quick Setup und stellen Sie sicher, dass Sie nur über eine Host-Management-Konfiguration verfügen. Wenn es zwei gibt, löschen Sie die ältere Konfiguration und warten Sie einige Minuten.

Wenn Sie nach dem Erstellen einer Host-Verwaltungs-Konfiguration immer noch keine Verbindung herstellen können oder wenn Sie eine Fehlermeldung erhalten, einschließlich einer Fehlermeldung zu SSM Agent, nutzen Sie eine der folgenden Lösungen:

- Lösung B: Kein Fehler, aber es kann immer noch keine Verbindung hergestellt werden

- Lösung C: Fehler wegen fehlendem SSM Agent

Lösung B: Kein Fehler, aber es kann immer noch keine Verbindung hergestellt werden

Wenn Sie die Host-Verwaltungs-Konfiguration erstellt haben, mehrere Minuten gewartet haben, bevor Sie versucht haben, eine Verbindung herzustellen, und immer noch keine Verbindung herstellen können, müssen Sie die Host-Management-Konfiguratio möglicherweise manuell auf Ihre Instance anwenden. Gehen Sie wie folgt vor, um eine Quick Setup Host-Verwaltungs-Konfiguration zu aktualisieren und Änderungen auf eine Instance anzuwenden.

So aktualisieren Sie eine Host-Verwaltungs-Konfiguration mit Quick Setup

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie in der Konfigurationsliste die Host-Verwaltungs-Konfiguration aus, die Sie erstellt haben.
4. Wählen Sie Aktionen und wählen Sie dann Konfiguration bearbeiten.
5. Wählen Sie im Abschnitt Ziele die Option Manuell.
6. Wählen Sie im Abschnitt Instances die Instance aus, die Sie erstellt haben.
7. Wählen Sie Aktualisieren.

Warten Sie einige Minuten, bis EC2 die Registerkarte Session Manager aktualisiert hat. Wenn Sie immer noch keine Verbindung herstellen können oder eine Fehlermeldung angezeigt wird, überprüfen Sie die verbleibenden Lösungen für dieses Problem.

Lösung C: Fehler wegen fehlendem SSM Agent

Wenn Sie mithilfe von Quick Setup keine Host-Verwaltungs-Konfiguration erstellen konnten oder wenn Sie die Fehlermeldung erhalten haben, dass SSM Agent nicht installiert ist, müssen Sie möglicherweise SSM Agent manuell auf Ihrer Instance installieren. SSM Agent ist eine Amazon-Software, die es Systems Manager ermöglicht, sich über Session Manager mit Ihrer Instance zu verbinden. SSM Agent ist standardmäßig auf den meisten Amazon Machine Images (AMIs) installiert. Wenn Ihre Instance aus einem nicht standardmäßigen AMI oder einem älteren AMI erstellt wurde, müssen Sie den Agenten möglicherweise manuell installieren. Informationen zum Installationsverfahren von SSM Agent finden Sie im folgenden Thema, das Ihrem Instance-Betriebssystem entspricht.

- Windows Server
- macOS
- AlmaLinux
- Amazon Linux 1
- Amazon Linux 2 und AL2023
- CentOS
- CentOS Stream
- Debian Server
- Oracle Linux
- Red Hat Enterprise Linux
- Rocky Linux
- SUSE Linux Enterprise Server
- Ubuntu Server

Bei Problemen mit SSM Agent siehe Fehlerbehebung für SSM Agent.

Keine Berechtigung zum Starten einer Sitzung

Problem: Sie versuchen, eine Sitzung zu starten. Das System teilt Ihnen jedoch mit, dass Sie nicht über die erforderlichen Berechtigungen verfügen.

- Lösung: Ein Systemadministrator hat Ihnen keine AWS Identity and Access Management (IAM-) Richtlinienberechtigungen zum Starten von Session Manager Sitzungen erteilt. Weitere Informationen finden Sie unter Kontrollieren des Sitzungszugriffs von Benutzern auf Instances.

Keine Berechtigung zum Ändern von Sitzungspräferenzen

Problem: Sie versuchen, globale Sitzungspräferenzen für Ihre Organisation zu aktualisieren. Das System teilt Ihnen jedoch mit, dass Sie nicht über die erforderlichen Berechtigungen verfügen.

- Lösung: Ihnen wurden vom Systemadministrator keine IAM-Richtlinienberechtigungen zum Festlegen von Session Manager-Präferenzen erteilt. Weitere Informationen finden Sie unter Gewähren oder Verweigern von Benutzerberechtigungen zum Aktualisieren von Session Manager-Einstellungen.

Verwalteter Knoten ist nicht verfügbar oder nicht für Session Manager konfiguriert

Problem 1: Sie möchten auf der Seite Start a session (Sitzung starten) der Konsole eine Sitzung starten. Es befindet sich jedoch kein verwalteter Knoten in der Liste.

- Lösung A: Der verwaltete Knoten, zu dem Sie eine Verbindung herstellen möchten, wurde möglicherweise nicht konfiguriert. AWS Systems Manager Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

  > **Note**
  >
  > Wenn er bereits auf einem verwalteten Knoten ausgeführt AWS Systems Manager SSM Agent wird, wenn Sie das IAM-Instanzprofil anhängen, müssen Sie den Agenten möglicherweise neu starten, bevor die Instanz auf der Seite Sitzungskonsole starten aufgeführt wird.

- Lösung B: Die Proxy-Konfiguration, die Sie auf den SSM Agent auf Ihrem verwalteten Knoten angewendet haben, ist möglicherweise falsch. Wenn die Proxy-Konfiguration falsch ist, kann der verwaltete Knoten die erforderlichen Service-Endpunkte nicht erreichen, oder der Knoten meldet sich möglicherweise als anderes Betriebssystem beim Systems Manager. Weitere Informationen finden Sie unter Konfiguration SSM Agent für die Verwendung eines Proxys auf Linux-Knoten und Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances.

Problem 2: Ein verwalteter Knoten, mit dem Sie eine Verbindung herstellen möchten, befindet sich in der Liste auf der Seite Start a session (Sitzung starten) der Konsole. Die Seite meldet jedoch, dass „die von Ihnen ausgewählte Instance nicht für die Verwendung mit Session Manager konfiguriert wurde.“

- Lösung A: Der verwaltete Knoten wurde für die Verwendung mit dem Systems-Manager-Service konfiguriert. Das dem Knoten angefügte IAM-Instance-Profil enthält jedoch möglicherweise keine Berechtigungen für die Session Manager-Funktion. Informationen hierzu finden Sie unter Überprüfen oder Erstellen eines IAM-Instance-Profils mit Session Manager-Berechtigungen.
- Lösung B: Der verwaltete Knoten wird nicht auf einer Version von SSM Agent ausgeführt, die den Session Manager unterstützt. Aktualisieren Sie SSM Agent auf dem Knoten auf die Version 2.3.68.0 oder höher.

  Sie können SSM Agent manuell auf einem verwalteten Knoten aktualisieren, indem Sie die in Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Windows Server,

Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux oder Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für macOS beschriebenen Schritte ausführen, abhängig vom Betriebssystem.

Alternativ können Sie das Run Command-Dokument `AWS-UpdateSSMAgent` verwenden, um die Agent-Version auf einem oder mehreren verwalteten Knoten gleichzeitig zu aktualisieren. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

> **Tip**
>
> Damit Ihr Agent stets auf dem aktuellen Stand ist, sollten Sie die Aktualisierung des SSM Agent auf die jeweils aktuelle Version anhand eines automatisierten Zeitplans ausführen, den Sie mittels einer der beiden folgenden Methoden definieren:
>
> - Führen Sie `AWS-UpdateSSMAgent` als Teil einer State Manager-Zuordnung aus. Weitere Informationen finden Sie unter Anleitung: Automatische Aktualisierung von SSM Agent (CLI).
> - Führen Sie `AWS-UpdateSSMAgent` als Teil eines Wartungsfensters aus. Weitere Informationen zum Arbeiten mit Wartungsfenstern finden Sie unter Arbeiten mit Wartungsfenstern (Konsole) und Tutorial: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI).

- Lösung C: Der verwaltete Knoten kann die erforderlichen Service-Endpunkte nicht erreichen. Sie können die Sicherheitslage Ihrer verwalteten Knoten verbessern, indem Sie Schnittstellenendpunkte verwenden, die mit Strom versorgt werden, AWS PrivateLink um eine Verbindung zu Systems Manager Manager-Endpunkten herzustellen. Die Alternative zur Verwendung von Schnittstellenendpunkten ist das Erlauben von ausgehendem Internetzugriff auf Ihre verwalteten Knoten. Weitere Informationen finden Sie unter Verwenden PrivateLink zum Einrichten eines VPC-Endpunkts für Session Manager.
- Lösung D: Der verwaltete Knoten verfügt über begrenzte verfügbare CPU- oder Speicherressourcen. Auch wenn Ihr verwalteter Knoten ansonsten funktionsfähig ist, können Sie keine Sitzung einrichten, wenn der Knoten nicht über genügend verfügbare Ressourcen verfügt. Weitere Informationen finden Sie unter Problembehandlung bei unerreichbaren Instances.

Session Manager-Plug-In nicht gefunden

Um die Befehle AWS CLI to run session verwenden zu können, muss das Session Manager Plugin auch auf Ihrem lokalen Computer installiert sein. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.

Session Manager-Plugin wurde nicht automatisch zumBefehlszeilenpfad hinzugefügt (Windows)

Wenn Sie das Session Manager-Plug-In auf Windows installieren, sollte die ausführbare Datei `session-manager-plugin` der Umgebungsvariablen `PATH` Ihres Betriebssystems automatisch hinzugefügt werden. Wenn die Prüfung auf korrekte Installation des Session Manager-Plugins (`aws ssm start-session --target` *`instance-id`*) ergibt, dass der Befehl fehlgeschlagen ist, müssen Sie das Plug-In möglicherweise mittels des folgenden Verfahrens manuell einrichten.

So ändern Sie die Variable PATH (Windows)

1. Betätigen Sie die Windows-Taste und geben Sie **`environment variables`** ein.
2. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).
3. Wählen Sie PATH und anschließend Edit.
4. Fügen Sie dem Feld Variable value (Variablenwert) Pfade hinzu, die durch Semikolons getrennt sind, wie in diesem Beispiel dargestellt: *`C:\existing\path`*;***`C:\new\path`***

   `C:\existing\path` stellt den Wert dar, der sich bereits im Feld befindet. ***`C:\new\path`*** stellt den Pfad dar, den Sie hinzufügen möchten, wie in diesen Beispielen dargestellt.

   - 64-Bit-Computer: `C:\Program Files\Amazon\SessionManagerPlugin\bin\`
   - 32-Bit-Computer: `C:\Program Files (x86)\Amazon\SessionManagerPlugin\bin\`
5. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
6. Schließen Sie alle etwa ausgeführten Eingabeaufforderungen und öffnen Sie erneut.

Session Manager-Plug-In reagiert nicht mehr

Während einer Port-Weiterleitungssitzung kann der Datenverkehr nicht mehr weitergeleitet werden, wenn auf Ihrem lokalen Computer Antivirus-Software installiert ist. In einigen Fällen stört Antivirus-Software das Session Manager-Plug-In, was zu Prozess-Deadlocks führt. Um dieses Problem zu

beheben, erlauben Sie das Session Manager-Plug-In in der Antivirus-Software oder schließen Sie es aus. Weitere Informationen zum Standardinstallationspfad für das Session Manager-Plug-In finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.

TargetNotVerbunden

Problem: Sie versuchen, eine Sitzung zu starten, aber das System gibt die Fehlermeldung „Beim Aufrufen der StartSession Operation ist ein Fehler aufgetreten (TargetNotConnected): *`InstanceID is not connected`*“ zurück.

- Lösung A: Dieser Fehler wird zurückgegeben, wenn der angegebene verwaltete Knoten für die Sitzung nicht vollständig für die Verwendung mit dem Session Manager konfiguriert wurde. Weitere Informationen finden Sie unter Einrichten von Session Manager.
- Lösung B: Dieser Fehler wird auch zurückgegeben, wenn Sie versuchen, eine Sitzung auf einem verwalteten Knoten zu starten, der sich in einem anderen AWS-Konto oder befindet. AWS-Region

Nach dem Starten einer Sitzung wird ein leerer Bildschirm angezeigt

Problem: Sie starten eine Sitzung und Session Manager zeigt einen leeren Bildschirm an.

- Lösung A: Dieses Problem kann auftreten, wenn das Root-Volume des verwalteten Knotens voll ist. Aufgrund von mangelndem Speicherplatz funktioniert SSM Agent auf dem Knoten nicht mehr. Um dieses Problem zu lösen, verwenden Sie Amazon, CloudWatch um Metriken und Protokolle von den Betriebssystemen zu sammeln. Weitere Informationen finden Sie unter Erfassung von Metriken, Protokollen und Traces mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.
- Lösung B: Möglicherweise wird ein leerer Bildschirm angezeigt, wenn Sie über einen Link auf die Konsole zugegriffen haben, der ein nicht übereinstimmendes Endpunkt- und Regionspaar enthält. Beispielsweise ist in der folgenden Konsolen-URL `us-west-2` der angegebene Endpunkt, aber `us-west-1` die angegebene AWS-Region.

  ```
  https://us-west-2.console.aws.amazon.com/systems-manager/session-manager/sessions?
  region=us-west-1
  ```

- Lösung C: Der verwaltete Knoten stellt über VPC-Endpunkte eine Verbindung zu Systems Manager her, und Ihre Session Manager Einstellungen schreiben die Sitzungsausgabe in einen Amazon S3 S3-Bucket oder eine Amazon CloudWatch Logs-Protokollgruppe, aber ein `s3` Gateway-Endpunkt oder `logs` Schnittstellenendpunkt ist in der VPC nicht vorhanden. Ein `s3`-Endpunkt

im Format **`com.amazonaws.`*`region`*`.s3`** ist erforderlich, wenn Ihre verwalteten Knoten eine Verbindung zu Systems Manager mithilfe von VPC-Endpunkten herstellen und Ihre Session Manager-Einstellungen die Sitzungsausgabe in einen Amazon-S3-Bucket schreiben. Alternativ **`com.amazonaws.`*`region`*`.logs`**ist ein `logs` Endpunkt in diesem Format erforderlich, wenn Ihre verwalteten Knoten über VPC-Endpunkte eine Verbindung zu Systems Manager herstellen und Ihre Session Manager Einstellungen die Sitzungsausgabe in eine CloudWatch Logs-Protokollgruppe schreiben. Weitere Informationen finden Sie unter Erstellen von VPC-Endpunkten für Systems Manager.

- Lösung D: Die Protokollgruppe oder der Amazon S3-Bucket, die/den Sie in Ihren Sitzungseinstellungen angegeben haben, wurde gelöscht. Aktualisieren Sie Ihre Sitzungseinstellungen mit einer gültigen Protokollgruppe oder einem gültigen S3-Bucket, um dieses Problem zu beheben.
- Lösung E: Die Protokollgruppe oder der Amazon S3-Bucket, die/den Sie in Ihren Sitzungseinstellungen angegeben haben, ist nicht verschlüsselt, aber Sie haben die `cloudWatchEncryptionEnabled`- oder `s3EncryptionEnabled`-Eingabe auf `true` eingestellt. Um dieses Problem zu beheben, aktualisieren Sie Ihre Sitzungseinstellungen mit einer Protokollgruppe oder einem Amazon S3-Bucket, die/der verschlüsselt ist, oder stellen Sie die `cloudWatchEncryptionEnabled`- oder `s3EncryptionEnabled`-Eingabe auf `false` ein. Dieses Szenario gilt nur für Kunden, die Sitzungseinstellungen mithilfe von Befehlszeilentools erstellen.

Verwalteter Knoten reagiert während langer Sitzungen nicht mehr

Problem: Ihr verwalteter Knoten reagiert nicht oder stürzt während einer langen Sitzung ab.

Lösung: Verringern Sie die SSM Agent-Protokollaufbewahrungsdauer für Session Manager.

So verringern Sie die SSM Agent-Protokollaufbewahrungsdauer für Sitzungen

1. Suchen Sie `amazon-ssm-agent.json.template` im `/etc/amazon/ssm/`-Verzeichnis für Linux oder `C:\Program Files\Amazon\SSM` für Windows.
2. Kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` in eine neue Datei im selben Verzeichnis namens `amazon-ssm-agent.json`.
3. Verringern Sie den Standardwert des `SessionLogsRetentionDurationHours`-Werts in der `SSM`-Eigenschaft und speichern Sie die Datei.
4. Starten Sie die SSM Agent neu.

Beim Aufrufen des Vorgangs ist ein Fehler aufgetreten (InvalidDocument) StartSession

Problem: Sie erhalten den folgenden Fehler, wenn Sie eine Sitzung mit dem AWS CLI starten.

```
An error occurred (InvalidDocument) when calling the StartSession operation: Document
 type: 'Command' is not supported. Only type: 'Session' is supported for Session
 Manager.
```

Lösung: Das SSM-Dokument, das Sie für den `--document-name`-Parameter angegeben haben, ist kein Sitzungsdokument. Gehen Sie wie folgt vor, um eine Liste von Sitzungsdokumenten in der AWS Management Console anzuzeigen.

So rufen Sie eine Liste von Sitzungsdokumenten auf

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie in der Liste Kategorien die Option Sitzungsdokumente aus.

AWS Systems Manager Run Command

Mit Run Command einer Funktion von AWS Systems Manager können Sie die Konfiguration Ihrer verwalteten Knoten remote und sicher verwalten. Ein verwalteter Knoten ist jede Instance der Amazon Elastic Compute Cloud (Amazon EC2) oder Nicht-EC2-Maschine in Ihrer Hybrid- und Multi-Cloud-Umgebung, die für Systems Manager konfiguriert wurde. Run Command ermöglicht es Ihnen, allgemeine Verwaltungsaufgaben zu automatisieren und einmalige Konfigurationsänderungen in großem Umfang durchzuführen. Sie können die SDKs Run Command aus AWS Management Console, AWS Command Line Interface (AWS CLI) oder aus den AWS SDKs verwenden. AWS Tools for Windows PowerShellRun Commandwird ohne zusätzliche Kosten angeboten. Um mit Run Command zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Run Command aus.

Administratoren verwenden Run Command zum Installieren oder Bootstrapping von Anwendungen, Entwickeln einer Bereitstellungs-Pipeline, Erfassen von Protokolldateien nach Entfernung einer Instance aus einer Auto-Scaling-Gruppe, zum Anschließen von Instances an eine Windows-Domain und noch mehr.

Erste Schritte

Die folgende Tabelle enthält Informationen, die Ihnen bei den ersten Schritten mit Run Command helfen werden.

Thema	Details
Einrichten AWS Systems Manager	Überprüfen Sie, ob Sie die Einrichtungsanforderungen für Ihre Instances der Amazon Elastic Compute Cloud (Amazon EC2) und Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung erfüllt haben.
Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen	(Optional) Registrieren Sie lokale Server und VMs bei, AWS damit Sie sie mithilfe von verwalten können. Run Command
the section called “Verwaltung von Edge-Geräten mit Systems Manager”	(Optional) Konfigurieren Sie Edge-Geräte, damit Sie sie mit Run Command verwalten können.
Ausführen von Befehlen auf verwalteten Knoten	Erfahren Sie, wie Sie mit der AWS Management Console einen Befehl ausführen, der einen oder mehrere verwaltete Knoten anvisiert.
Walkthroughs für Run Command	Erfahren Sie, wie Sie Befehle entweder mit Tools für Windows PowerShell oder mit dem ausführen. AWS CLI

EventBridge Unterstützung

Diese Systems Manager Manager-Funktion wird in EventBridge Amazon-Regeln sowohl als Ereignistyp als auch als Zieltyp unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager.

Weitere Informationen

- Remote-Run Command auf einer EC2-Instance (10-Minuten-Tutorial)
- Systems Manager-Service Quotas im Allgemeine Amazon Web Services-Referenz

- AWS Systems Manager API Reference

Themen

- Einrichten von Run Command
- Ausführen von Befehlen auf verwalteten Knoten
- Verwendung von Beendigungscodes in Befehlen
- Grundlegendes zu Befehlsstatus
- Walkthroughs für Run Command
- Fehlerbehebung von Systems Manager Run Command

Einrichten von Run Command

Bevor Sie Knoten mit Run Command, eine Funktion von AWS Systems Manager, verwalten können, müssen Sie eine AWS Identity and Access Management (IAM)-Richtlinie für jeden Benutzer konfigurieren, der Befehle ausführen wird.

Sie müssen Ihre Knoten auch für Systems Manager konfigurieren. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

Wir empfehlen, die folgenden optionalen Einrichtungs-Aufgaben auszuführen, um den Sicherheitsstatus und die tägliche Verwaltung Ihrer verwalteten Knoten zu minimieren.

Ausführung von Befehlen mit Amazon EventBridge überwachen

Mit EventBridge können Sie Änderungen des Status der Befehlsausführung protokollieren. Sie können eine Regel erstellen, die ausgeführt wird, sobald ein Statusübergang oder ein Übergang zu einem oder mehreren Status stattfindet, die für sie von Interesse sind. Sie können auch Run Command als Zielaktion bei Eintreten eines EventBridge-Ereignisses angeben. Weitere Informationen finden Sie unter Konfigurieren von EventBridge für Systems Manager-Ereignisse.

Ausführung von Befehlen mit Amazon CloudWatch Logs überwachen

Sie können Run Command so konfigurieren, dass alle Befehlsausgaben und Fehlerprotokolle periodisch an eine Amazon CloudWatch-Protokollgruppe gesendet werden. Sie können diese Ausgabeprotokolle nahezu in Echtzeit überwachen, nach bestimmten Phrasen, Werten oder Mustern suchen und auf der Grundlage der Suche Warnungen erstellen. Weitere Informationen finden Sie unter Konfiguration von Amazon CloudWatch Logs für Run Command.

Den Zugriff von Run Command auf bestimmte verwaltete Knoten beschränken

Sie können die Fähigkeit eines Benutzers einschränken, Befehle auf verwalteten Knoten auszuführen, indem Sie AWS Identity and Access Management (IAM) verwenden. Insbesondere können Sie eine IAM-Richtlinie mit der Bedingung erstellen, dass der Benutzer nur Befehle auf verwalteten Knoten ausführen kann, die mit bestimmten Tags gekennzeichnet sind. Weitere Informationen finden Sie unter Den Zugriff von Run Command anhand von Tags beschränken.

Den Zugriff von Run Command anhand von Tags beschränken

In diesem Abschnitt wird beschrieben, wie die Fähigkeit eines Benutzers eingeschränkt werden kann, Befehle für verwaltete Knoten auszuführen, indem eine Tag-Bedingung in einer IAM-Richtlinie angegeben wird. Zu verwalteten Knoten gehören Amazon-EC2-Instances und Nicht-EC2-Knoten in einer Hybrid- und Multi-Cloud-Umgebung, die für Systems Manager konfiguriert sind. Obwohl die Informationen nicht explizit dargestellt werden, können Sie auch den Zugriff auf verwaltete AWS IoT Greengrass-Core-Geräte einschränken. Zuerst müssen Sie Ihre AWS IoT Greengrass-Geräte markieren. Weitere Informationen finden Sie unter Markieren Ihrer AWS IoT Greengrass Version 2-Ressourcen im AWS IoT Greengrass Version 2-Entwicklerhandbuch.

Sie können die Befehlsausführung auf bestimmte verwaltete Knoten beschränken, indem Sie eine IAM-Richtlinie erstellen, die eine Bedingung enthält, dass der Benutzer Befehle nur auf Knoten mit bestimmten Tags ausführen kann. Im folgenden Beispiel ist der Benutzer berechtigt, Run Command (`Effect: Allow, Action: ssm:SendCommand`) zu verwenden, indem er ein beliebiges SSM-Dokument (`Resource: arn:aws:ssm:*:*:document/*`) auf einem beliebigen Knoten (`Resource: arn:aws:ec2:*:*:instance/*`) nutzt, unter der Bedingung, dass der Knoten ein Finance WebServer (`ssm:resourceTag/Finance: WebServer`) ist. Wenn der Benutzer einen Befehl an einen Knoten sendet, der nicht markiert ist oder ein anderes Tag als `Finance: WebServer` hat, zeigen die Ausführungsergebnisse `AccessDenied`.

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

Sie können IAM-Richtlinien erstellen, die es einem Benutzer erlauben, Befehle auf verwalteten Knoten auszuführen, die mit mehreren Tags markiert sind. Mit der folgenden Richtlinie hat der Benutzer die Möglichkeit, Befehle auf verwalteten Knoten auszuführen, die über zwei Tags verfügen. Wenn ein Benutzer einen Befehl an einen verwalteten Knoten sendet, der nicht mit beiden dieser Tags markiert ist, zeigen die Ausführungsergebnisse `AccessDenied`.

```
{
   "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":"*"
      }
   ]
}
```

Sie können auch IAM-Richtlinien erstellen, die es einem Benutzer erlauben, Befehle auf mehreren Gruppen von markierten verwalteten Knoten auszuführen. Die folgende Beispiel-Richtlinie erlaubt dem Benutzer die Ausführung von Befehlen entweder für eine der Gruppen von markierten Knoten oder für beide Gruppen.

```
{
   "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":"*"
       }
    ]
}
```

Weitere Informationen zum Erstellen von IAM-Richtlinien finden Sie unter Verwaltete Richtlinien und eingebundene Richtlinien im IAM-Benutzerhandbuch. Weitere Informationen über das Markieren verwalteter Knoten finden Sie unter Tag-Editor im AWS Resource Groups-Benutzerhandbuch.

Ausführen von Befehlen auf verwalteten Knoten

Dieser Abschnitt enthält Informationen darüber, wie Befehle aus der AWS Systems Manager-Konsole zu verwalteten Knoten gesendet werden. Dieser Abschnitt enthält auch Informationen zum Abbrechen eines Befehls.

Informationen zum Senden von Befehlen unter Verwendung von Windows PowerShell finden Sie unter Exemplarische Vorgehensweise: Verwenden Sie das mit AWS Tools for Windows PowerShellRun Command oder die Beispiele im Abschnitt AWS Systems Manager der AWS Tools for PowerShell-Cmdlet Reference. Weitere Informationen zum Senden von Befehlen unter Verwendung der AWS Command Line Interface (AWS CLI) finden Sie unter Anleitung: Verwenden der AWS CLI mit Run Command oder in der SSM CLI Reference.

> **Important**
>
> Wenn Sie einen Befehl mit Run Command senden, schließen Sie keine vertraulichen Informationen ein, die als Klartext formatiert sind, z. B. Passwörter, Konfigurationsdaten oder andere Geheimnisse. Alle Systems-Manager-API-Aktivitäten in Ihrem Konto werden in einem S3-Bucket für AWS CloudTrail-Protokolle protokolliert. Dies bedeutet, dass jeder Benutzer mit Zugriff auf den S3-Bucket die Klartextwerte dieser Geheimnisse anzeigen kann. Aus diesem Grund empfehlen wir, `SecureString`-Parameter zu erstellen und zu verwenden, um die sensiblen Daten zu verschlüsseln, die Sie in Ihren Systems-Manager-Operationen verwenden.
> Weitere Informationen finden Sie unter Einschränken des Zugriffs auf Systems Manager-Parameter mithilfe von IAM-Richtlinien.

Inhalt

- Ausführen von Befehlen über die Konsole
- Ausführen von Befehlen mit einer bestimmten Dokumentversion
- Ausführen von Befehlen in großem Maßstab
- Stornieren eines Befehls

Ausführen von Befehlen über die Konsole

Sie können eine Funktion vonRun Command, von verwenden AWS Systems Manager, um verwaltete Knoten AWS Management Console zu konfigurieren, ohne sich bei ihnen anmelden zu müssen. Dieses Thema enthält ein Beispiel für die Ausführung eines SSM Agent-Updates auf einem verwalteten Knoten mithilfe von Run Command.

Bevor Sie beginnen

Bevor Sie einen Befehl mit Run Command senden, müssen Sie überprüfen, ob Ihre verwalteten Knoten die Systems-Manager-Einrichtungs-Anforderungen erfüllen.

So senden Sie einen Befehl mittels Run Command

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document ein Systems Manager-Dokument.
5. Geben Sie im Abschnitt Command parameters Werte für erforderliche Parameter an.
6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Other parameters (Weitere Parameter):
   - Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.
8. Für Rate control (Ratenregelung):

- Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wählen Sie einen CloudWatch Alarm aus, der auf Ihren Überwachungsbefehl angewendet werden soll. Um Ihrem Befehl einen CloudWatch Alarm hinzuzufügen, muss der IAM-Principal, der den Befehl ausführt, über die entsprechende Berechtigung verfügen`iam:createServiceLinkedRole`. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden. Beachten Sie, dass ausstehende Befehlsaufrufe nicht ausgeführt werden, wenn Ihr Alarm aktiviert wird.
10. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem

anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

11. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.
12. Wählen Sie Ausführen aus.

Weitere Informationen zum Abbrechen eines Befehls finden Sie unter the section called “Stornieren eines Befehls”.

Erneutes Ausführen von Befehlen

Systems Manager enthält zwei Optionen, mit denen Sie einen Befehl auf der Seite Run Command (Befehl ausführen) in der Systems Manager-Konsole erneut ausführen können.

- Rerun (Erneut ausführen): Über diese Schaltfläche können Sie denselben Befehl ausführen, ohne Änderungen daran vorzunehmen.
- Copy to new (In neu kopieren): Über diese Schaltfläche kopieren Sie die Einstellungen eines Befehls in einen neuen Befehl und erhalten die Möglichkeit, diese Einstellungen zu bearbeiten, bevor Sie den Befehl ausführen.

So führen Sie einen Befehl erneut aus

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie einen Befehl aus, der erneut ausgeführt werden soll. Sie können einen Befehl unmittelbar nach der Ausführung von der Befehlsdetailseite aus erneut ausführen. Sie können auch einen Befehl auswählen, den Sie zuvor auf der Registerkarte Command history (Befehlsverlauf) ausgeführt haben.

4. Wählen Sie entweder Rerun (Erneut ausführen) aus, um denselben Befehl ohne Änderungen auszuführen, oder wählen Sie Copy to new (In neuen kopieren) aus, um die Befehlseinstellungen zu bearbeiten, bevor Sie den Befehl ausführen.

Ausführen von Befehlen mit einer bestimmten Dokumentversion

Sie können den Dokumentversionsparameter verwenden, um anzugeben, welche Version eines AWS Systems Manager-Dokuments verwendet werden soll, wenn der Befehl ausgeführt wird. Sie können eine der folgenden Optionen für diesen Parameter festlegen:

- $DEFAULT
- $LATEST
- Versionsnummer:

Gehen Sie wie folgt vor, um einen Befehl unter Verwendung des Dokumentversionsparameters auszuführen.

Linux

So führen Sie Befehle über die AWS CLI auf lokalen Linux-Computern aus

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Listen Sie alle verfügbaren Dokumente auf

   Mit diesem Befehl werden alle verfügbaren Dokumente für Ihr Konto basierend auf AWS Identity and Access Management (IAM)-Berechtigungen ausgeführt.

   ```
   aws ssm list-documents
   ```

3. Verwenden Sie den folgenden Befehl, um die verschiedenen Versionen eines Dokuments anzuzeigen. Ersetzen Sie *Dokumentname* durch Ihre eigenen Informationen.

   ```
   aws ssm list-document-versions \
       --name "document name"
   ```

4. Führen Sie mit dem folgenden Befehl einen Befehl aus, der eine SSM-Dokumentversion verwendet. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --parameters commands="echo Hello" \
    --instance-ids instance-ID \
    --document-version '$LATEST'
```

Windows

So führen Sie Befehle über die AWS CLI auf lokalen Windows-Computern aus

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Listen Sie alle verfügbaren Dokumente auf

   Mit diesem Befehl werden alle verfügbaren Dokumente für Ihr Konto basierend auf AWS Identity and Access Management (IAM)-Berechtigungen ausgeführt.

```
aws ssm list-documents
```

3. Verwenden Sie den folgenden Befehl, um die verschiedenen Versionen eines Dokuments anzuzeigen. Ersetzen Sie *Dokumentname* durch Ihre eigenen Informationen.

```
aws ssm list-document-versions ^
    --name "document name"
```

4. Führen Sie mit dem folgenden Befehl einen Befehl aus, der eine SSM-Dokumentversion verwendet. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws ssm send-command ^
    --document-name "AWS-RunShellScript" ^
    --parameters commands="echo Hello" ^
```

```
    --instance-ids instance-ID ^
    --document-version "$LATEST"
```

PowerShell

Ausführen von Befehlen mit den Tools for PowerShell

1. Installieren und konfigurieren Sie die AWS Tools for PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS Tools for PowerShell.
2. Listen Sie alle verfügbaren Dokumente auf

   Mit diesem Befehl werden alle verfügbaren Dokumente für Ihr Konto basierend auf AWS Identity and Access Management (IAM)-Berechtigungen ausgeführt.

   ```
   Get-SSMDocumentList
   ```
3. Verwenden Sie den folgenden Befehl, um die verschiedenen Versionen eines Dokuments anzuzeigen. Ersetzen Sie *Dokumentname* durch Ihre eigenen Informationen.

   ```
   Get-SSMDocumentVersionList `
       -Name "document name"
   ```
4. Führen Sie mit dem folgenden Befehl einen Befehl aus, der eine SSM-Dokumentversion verwendet. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   ```
   Send-SSMCommand `
       -DocumentName "AWS-RunShellScript" `
       -Parameter @{commands = "echo helloWorld"} `
       -InstanceIds "instance-ID" `
       -DocumentVersion $LATEST
   ```

Ausführen von Befehlen in großem Maßstab

Sie können Run Command, eine Funktion von AWS Systems Manager, verwenden, um Befehle auf einer Flotte verwalteter Knoten auszuführen, indem Sie `targets` verwenden. Der `targets`-Parameter nimmt eine `Key,Value`-Kombination basierend auf Tags an, die Sie für Ihre verwalteten

Knoten angegeben haben. Wenn Sie den Befehl ausführen, versucht das System, den Befehl auf allen verwalteten Knoten auszuführen, die den angegebenen Tags entsprechen. Weitere Informationen zum Markieren von verwalteten Instances finden Sie unter Markieren Ihrer AWS-Ressourcen im Benutzerhandbuch zum Markieren von AWS-Ressourcen. Weitere Informationen über das Markieren Ihrer verwalteten IoT-Geräte finden Sie unter Markieren Sie Ihre AWS IoT Greengrass Version 2-Ressourcen im AWS IoT Greengrass Version 2-Entwicklerhandbuch.

Sie können auch den `targets`-Parameter verwenden, um eine Liste spezifischer IDs von verwalteten Knoten als Ziel zu wählen, wie im nächsten Abschnitt beschrieben.

Zur Steuerung der Befehlsausführung auf Hunderten und Tausenden von verwalteten Knoten umfasst Run Command auch Parameter für die Einschränkung, wie viele Knoten gleichzeitig eine Anforderung verarbeitet können, und wie viele Fehler durch einen Befehl ausgelöst werden können, bevor der Befehl beendet wird.

Inhalt

- Mehrere verwaltete Knoten anvisieren
- Verwenden von Ratensteuerungen

Mehrere verwaltete Knoten anvisieren

Sie können einen Befehl ausführen und verwaltete Knoten anvisieren, indem Sie Tags, Namen von AWS-Ressourcengruppen oder IDs von verwalteten Knoten angeben.

Die folgenden Beispiele zeigen das Befehlsformat bei Verwendung von Run Command aus der AWS Command Line Interface (AWS CLI). Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen. Die Beispielbefehle werden in diesem Abschnitt sind mit `[...]` abgeschnitten.

Beispiel 1: Angabe von Tags als Ziel

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 ^
    [...]
```

Beispiel 2: Angabe einer AWS-Ressourcengruppe nach Namen

Sie können maximal einen Ressourcengruppenamen pro Befehlsaufruf angeben. Wenn Sie eine Ressourcengruppe erstellen, empfehlen wir, `AWS::SSM:ManagedInstance` und `AWS::EC2::Instance` als Ressourcentypen in dem Gruppierungskriterium aufzunehmen.

> **Note**
>
> Zum Senden von Befehlen mit einer Ressourcengruppe als Ziel benötigen Sie AWS Identity and Access Management (IAM)-Berechtigungen zum Auflisten oder Anzeigen der Ressourcen, die zu der Gruppe gehören. Weitere Informationen finden Sie unter Einrichten von Berechtigungen im AWS Resource Groups-Benutzerhandbuch.

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 ^
    [...]
```

Beispiel 3: Angabe einer AWS-Ressource Groups nach Ressourcentyp

Sie können maximal fünf Ressourcengruppentypen pro Befehlsaufruf angeben. Wenn Sie eine Ressourcengruppe erstellen, empfehlen wir, `AWS::SSM:ManagedInstance` und `AWS::EC2::Instance` als Ressourcentypen in dem Gruppierungskriterium aufzunehmen.

> **Note**
>
> Zum Senden von Befehlen mit einer Ressourcengruppe als Ziel benötigen Sie IAM-Berechtigungen zum Auflisten oder Anzeigen der Ressourcen, die zu der Gruppe gehören. Weitere Informationen finden Sie unter Einrichten von Berechtigungen im AWS Resource Groups-Benutzerhandbuch.

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 ^
    [...]
```

Beispiel 4: Angabe von Instance-IDs als Ziel

Die folgenden Beispiele veranschaulichen, wie verwaltete Knoten mithilfe des `instanceids`-Schlüssels mit dem `targets`-Parameter anvisiert werden können. Sie können diesen Schlüssel verwenden, um verwaltete AWS IoT Greengrass-Core-Geräte anzuvisieren, da jedem Gerät ein mi-*ID_Nummer* zugewiesen ist. Sie können Geräte-IDs in Fleet Manager, eine Funktion von AWS Systems Manager, anzeigen.

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 ^
    [...]
```

Wenn Sie verwaltete Knoten für unterschiedliche Umgebungen mit einem `Key` namens `Environment` und `Values` von `Development`, `Test`, `Pre-production` und`Production` markiert haben, könnten Sie einen Befehl an alle verwalteten Knoten in einer dieser Umgebungen mit dem `targets`-Parameter mit der folgenden Syntax senden.

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 ^
    [...]
```

Sie könnten weitere verwaltete Knoten in anderen Umgebungen auswählen, indem Sie sie zur `Values`-Liste hinzufügen. Trennen Sie die Elemente durch Kommas.

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 ^
    [...]
```

Variation: Anpassung Ihrer Ziele mit mehreren `Key`-Kriterien

Sie können die Anzahl der Ziele für Ihren Befehl verfeinern, indem Sie mehrere `Key` Kriterien berücksichtigen. Wenn Sie mehr als ein `Key`-Kriterium einschließen, wird das System verwaltete Knoten anvisieren, die alle Kriterien erfüllen. Mit dem folgenden Befehl werden alle verwaltete Knoten anvisiert, die für die Finanzabteilung und für die Datenbankserver-Rolle markiert sind.

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 ^
    [...]
```

Variation: Verwenden mehrerer `Key`- und `Value`-Kriterien

Aufbauend auf dem vorherigen Beispiel können Sie mehrere Abteilungen und mehrere Server-Rollen auswählen, indem zusätzliche Elemente in die `Values` Kriterien einfügen.

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 ^
    [...]
```

Variation: Anvisieren markierter verwalteter Knoten mithilfe von mehreren `Values`-Kriterien

Wenn Sie verwaltete Knoten für unterschiedliche Umgebungen mit einem `Key` namens `Department` und `Values` von `Sales` und `Finance` markiert haben, könnten Sie einen Befehl an alle Knoten in diesen Umgebungen mit dem `targets`-Parameter mit der folgenden Syntax senden.

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 ^
    [...]
```

Sie können maximal fünf Schlüssel und fünf Werte für jeden Schlüssel angeben.

Wenn entweder ein Tag-Schlüssel (der Variablenname) oder eine Tag-Wert Leerzeichen enthält, setzen Sie den Tagschlüssel oder den Wert in Anführungszeichen, wie in den folgenden Beispielen gezeigt.

Beispiel: Leerzeichen in `Value`-Tag

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:OS,Values="Windows Server 2016 Nano" \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:OS,Values="Windows Server 2016 Nano" ^
    [...]
```

Beispiel: Leerzeichen in `tag`-Schlüssel und `Value`

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key="tag:Operating System",Values="Windows Server 2016 Nano" \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key="tag:Operating System",Values="Windows Server 2016 Nano" ^
    [...]
```

Beispiel: Leerzeichen in einem Element in einer Liste von `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" ^
    [...]
```

Verwenden von Ratensteuerungen

Sie können das Tempo steuern, mit dem Befehle an verwaltete Knoten in einer Gruppe gesendet werden, indem Sie Nebenläufigkeits-Kontrollen und Fehlerkontrollen verwenden.

Themen

- Verwenden von Gleichzeitigkeitssteuerungen
- Verwenden von Fehlersteuerungen

Verwenden von Gleichzeitigkeitssteuerungen

Sie können die Anzahl der verwalteten Knoten steuern, die einen Befehl gleichzeitig ausführen, indem Sie den `max-concurrency`-Parameter verwenden (die Concurrency (Nebenläufigkeit)-Optionen auf der Seite Run a command (Befehl ausführen). Sie können entweder eine absolute Anzahl an verwalteten Knoten, z. B. **`10`**, oder einen Prozentsatz des festgelegten Ziels, beispielsweise **`10%`**, angeben. Das Warteschlangensystem liefert den Befehl an einen einzelnen Knoten und wartet, bis das System den ersten Aufruf bestätigt hat, bevor der Befehl an zwei weitere Knoten gesendet wird. Das System sendet exponentiell Befehle an mehrere Knoten, bis das System den Wert `max-concurrency` erreicht hat. Der Standardwert `max-concurrency` beträgt 50. Die folgenden Beispiele zeigen, wie Sie Werte für den Parameter `max-concurrency` angeben.

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 ^
    [...]
```

Verwenden von Fehlersteuerungen

Sie können auch die Ausführung eines Befehls auf Hunderten oder Tausenden von verwalteten Knoten steuern, indem Sie eine Fehlerbegrenzung mit den `max-errors`-Parametern einstellen (das Feld Error threshold (Fehlerschwelle) auf der Seite Run a command (Befehl ausführen)). Der Parameter gibt an, wie viele Fehler zulässig sind, bevor das System keinen Befehl mehr an zusätzliche verwaltete Knoten sendet. Sie können entweder eine absolute Anzahl an Fehlern, z. B. **`10`**, oder einen Prozentsatz des festgelegten Ziels, beispielsweise **`10%`**, festlegen. Wenn Sie z. B. **`3`** angeben, sendet das System keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Wenn Sie **`0`** angeben, sendet das System keinen weiteren Befehl an zusätzliche verwaltete Knoten, nachdem das erste Fehlerergebnis zurückgegeben wird. Wenn Sie einen Befehl an 50 verwaltete Knoten senden und `max-errors` auf **`10%`** einstellen, sendet das System keinen Befehl mehr an weitere Knoten, wenn der sechste Fehler empfangen wird.

Aufrufe, die bereits einen Befehl ausführen, wenn `max-errors` erreicht ist, dürfen abgeschlossen werden, jedoch können einige dieser Aufrufe ebenso fehlschlagen. Wenn Sie sicherstellen müssen, dass es nicht mehr als `max-errors` fehlgeschlagene Aufrufe geben wird, setzen Sie `max-concurrency` auf **`1`**, sodass die Aufrufe jeweils um eins fortfahren. Die Standardwert für max-errors ist 0. Die folgenden Beispiele zeigen, wie Sie Werte für den Parameter `max-errors` angeben.

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 ^
```

```
    [...]
```

Stornieren eines Befehls

Sie können versuchen, einen Befehl abzubrechen, solange der Service entweder im Ausstehenden oder Ausführenden Status angezeigt wird. Wenn jedoch ein Befehl sich nach wie vor in einem dieser Zustände befindet, können wir nicht garantieren, dass der Befehl beendet wird und der zugrunde liegenden Prozess gestoppt wurde.

So stornieren Sie einen Befehl mithilfe der Konsole

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie den Befehlsaufruf, den Sie stornieren möchten.
4. Wählen Sie Cancel Command.

Um einen Befehl mit dem abzubrechen AWS CLI

Führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

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"
```

Weitere Informationen über den Status eines stornierten Befehls finden Sie unter Grundlegendes zu Befehlsstatus.

Verwendung von Beendigungscodes in Befehlen

In einigen Fällen müssen Sie möglicherweise mithilfe von Beendigungscodes verwalten, wie Ihre Befehle verarbeitet werden.

Angabe von Beendigungscodes in Befehlen

Mit Run Command, einer Funktion von AWS Systems Manager, können Sie Beendigungscodes angeben, um festzulegen, wie Befehle verarbeitet werden. Standardmäßig wird der Beendigungscode des letzten in einem Skript ausgeführten Befehls als Beendigungscode für das gesamte Skript gemeldet. Sie haben beispielsweise ein Skript, das drei Befehle enthält. Der erste schlägt fehl, die folgenden werden jedoch erfolgreich ausgeführt. Da der letzte Befehl erfolgreich war, wird der Status der Ausführung als `succeeded` gemeldet.

Shell-Skripts

Damit das gesamte Skript beim ersten Befehlsfehler fehlschlägt, können Sie eine bedingte Shell-Anweisung einschließen, sodass das Skript beendet wird, wenn ein Befehl vor dem letzten Befehl fehlschlägt. Gehen Sie wie folgt vor.

```
<command 1>
    if [ $? != 0 ]
    then
        exit <N>
    fi
    <command 2>
    <command 3>
```

Im folgenden Beispiel schlägt das gesamte Skript fehl, wenn der erste Befehl fehlschlägt.

```
cd /test
    if [ $? != 0 ]
    then
        echo "Failed"
        exit 1
    fi
    date
```

PowerShell-Skripts

PowerShell erfordert, dass Sie `exit` explizit in Ihren Skripts aufrufen, damit Run Command den Beendigungscode erfolgreich erfassen kann.

```
<command 1>
    if ($?) {<do something>}
    else {exit <N>}
    <command 2>
    <command 3>
    exit <N>
```

Ein Beispiel:

```
cd C:\
    if ($?) {echo "Success"}
    else {exit 1}
    date
```

Umgang mit Neustarts beim Ausführen von Befehlen

Wenn Sie Run Command, eine Funktion von AWS Systems Manager, verwenden, um Skripts auszuführen, die verwaltete Knoten neu starten, empfehlen wir Ihnen, einen Beendigungscode in Ihrem Skript anzugeben. Wenn Sie versuchen, einen Knoten von einem Skript aus mit einem anderen Verfahren neu zu starten, wird der Ausführungsstatus des Skripts möglicherweise nicht korrekt aktualisiert. Dies passiert auch dann, wenn der Neustart der letzte Schritt in Ihrem Skript ist. Für Windows-verwaltete Knoten geben Sie `exit 3010` in Ihrem Skript an. Für Linux- und macOS-verwaltete Knoten geben Sie `exit 194` an. Der Beendigungscode weist AWS Systems Manager den Agent (SSM Agent) an, den verwalteten Knoten neu zu starten und das Skript nach Abschluss des Neustarts neu zu starten. Vor dem Neustart informiert SSM Agent den Systems Manager-Service in der Cloud, dass die Kommunikation während des Serverneustarts unterbrochen werden wird.

> **Note**
>
> Das Neustartskript kann nicht Teil eines `aws:runDocument`-Plugins sein. Wenn ein Dokument das Neustartskript enthält und ein anderes Dokument versucht, dieses Dokument über das `aws:runDocument`-Plugin auszuführen, gibt SSM Agent einen Fehler zurück.

Idempotente Skripts erstellen

Bei der Entwicklung von Skripts, die verwaltete Knoten neu starten, machen Sie die Skripts idempotent, damit die Skriptausführung nach dem Neustart an der Stelle fortgesetzt wird, wo sie unterbrochen wurde. Idempotente Skripts verwalten den Status und prüfen, ob die Aktion ausgeführt wurde oder nicht. Dadurch wird verhindert, dass ein Schritt mehrmals ausgeführt wird, wenn er nur einmal ausgeführt werden soll.

Hier finden Sie ein Beispiel für ein idempotentes Skript, das einen verwalteten Knoten mehrfach neu startet.

```
$name = Get current computer name
If ($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
    }
```

Beispiele

Die folgenden Skript-Beispiele verwenden Beendigungscodes für den Neustart von verwalteten Knoten. Das Linux-Beispiel installiert Paket-Updates auf Amazon Linux und startet den Knoten dann neu. Das Windows Server Beispiel installiert den Telnet-Client auf dem Knoten und startet ihn dann neu.

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
    }
```

Grundlegendes zu Befehlsstatus

Run Command, eine Funktion von AWS Systems Manager, meldet detaillierte Statusinformationen über die verschiedenen Zustände, die ein Befehl während der Verarbeitung erlebt, und für jeden verwalteten Knoten, der den Befehl verarbeitet hat. Sie können Befehlsstatus mithilfe der folgenden Methoden überwachen:

- Wählen Sie das Symbol Refresh (Aktualisieren) auf der Registerkarte Commands (Befehle) in der Run Command-Konsolenschnittstelle.
- Rufen Sie mit der () Listenbefehle oder Listenbefehl-Aufrufe auf. AWS Command Line Interface AWS CLIOder rufen Sie Get-SSMCommand oder AWS Tools for Windows PowerShell Get-SSM mit CommandInvocation.
- Konfigurieren Sie Amazon so EventBridge , dass es auf Status- oder Statusänderungen reagiert.
- Konfigurieren Sie Amazon Simple Notification Service (Amazon SNS), um Benachrichtigungen für alle Statusänderungen oder bestimmte Status wie `Failed` oder `TimedOut` zu senden.

Run Command-Status

Run Command berichtet Statusdetails für drei Bereiche: Plugins, Aufrufe und eine allgemeinen Compliance-Befehl. Ein Plugin ist ein Code-Ausführungsblock, der im SSM-Dokument des Befehls definiert ist. Weitere Informationen zu den Plug-ins finden Sie unter Referenz für Befehlsdokument-Plug-ins.

Wenn Sie einen Befehl an mehrere verwaltete Knoten gleichzeitig senden, ist jede Kopie des Befehls, die jeden Knoten anvisiert, ein Befehlsaufruf. Wenn Sie z. B. das `AWS-RunShellScript`-Dokument verwenden und einen `ifconfig`-Befehl an 20 Linux-Instances senden, hat dieser Befehl 20 Aufrufe. Jeder Befehlsaufruf berichtet einzeln einen Status. Die Plug-ins für einen bestimmten Befehlsaufruf berichten ebenfalls einzeln einen Berichtstatus.

Schließlich umfasst Run Command einen zusammenfassenden Befehlsstatus für alle Plugins und Aufrufe. Der aggregierte Befehlsstatus kann von dem Status, der von Plug-ins oder Aufrufen gemeldet wird, wie in den folgenden Tabellen gezeigt, abweichen.

> **Note**
>
> Wenn Sie Befehle mit den Parametern `max-concurrency` oder `max-errors` für eine großen Anzahl von verwalteten Knoten ausführen, spiegelt der Befehlsstatus die durch diese Parameter auferlegten Grenzen wider, wie in den folgenden Tabellen beschrieben. Weitere Informationen zu diesen Parametern finden Sie unter Ausführen von Befehlen in großem Maßstab.

Detaillierter Status für Befehls-Plugins und Aufrufe

Status	Details
Ausstehend	Der Befehl wurde noch nicht an den verwaltet en Knoten gesendet oder wurde nicht vom SSM Agent empfangen. Wird der Befehl nicht vor Ablauf der Zeitspanne, die der Summe aus dem Parameter Timeout (seconds) und dem Parameter Execution timeout entspricht, vom Agenten empfangen, ändert sich der Status in `Delivery Timed Out`.
InProgress	Systems Manager versucht, den Befehl an den verwalteten Knoten zu senden, oder der Befehl wurde vom SSM Agent empfangen und wird auf der Instance ausgeführt. Je nach Ergebnis aller Befehls-Plugins ändert sich der Status in `Success`, `Failed`, `Delivery Timed Out`,

Status	Details
	oder `Execution Timed Out`. Ausnahme: Wenn der Agent nicht ausgeführt oder auf dem Knoten nicht verfügbar ist, bleibt der Befehlsstatus bei `In Progress`, bis der Agent wieder verfügbar ist oder bis das Ausführungs-Timeout-Limit erreicht ist. Der Status wechselt dann in einen Terminal-Status.
Verzögert	Das System versuchte, den Befehl an den verwalteten Knoten zu senden, war jedoch nicht erfolgreich. Das System startet einen erneuten Versuch.

Status	Details
Herzlichen Glückwunsch	Dieser Status wird unter verschiedenen Bedingungen zurückgegeben. Dieser Status bedeutet nicht, dass der Befehl auf dem Knoten verarbeitet wurde. Beispielsweise kann der Befehl SSM Agent auf dem verwalteten Knoten empfangen werden und den Exit-Code Null zurückgeben, wenn Sie die Ausführung des Befehls PowerShell `ExecutionPolicy` verhindern. Diese ist ein Terminalstatus. Bedingungen, die dazu führen, dass ein Befehl einen `Success` Status zurückgibt, sind: • Bei der Ausrichtung auf eine einzelne Instanz wurde der Befehl SSM Agent auf dem verwalteten Knoten empfangen und es wurde der Exit-Code Null zurückgegeben. • Beim Targeting auf mehrere Instanzen hat die Anzahl der fehlgeschlagenen Aufrufe den im Befehl angegebenen Fehlerschwellenwert nicht überschritten. • Beim Targeting auf mehrere Instanzen war mindestens ein Aufruf erfolgreich, während bei anderen das Timeout abgelaufen ist. Der angegebene Fehlerschwellenwert gilt weiterhin. • Beim Targeting auf ein Tag werden keine Instanzen gefunden, die dem Tag zugeordne t sind. • Beim Targeting auf ein Tag hat die Anzahl der fehlgeschlagenen Aufrufe den im Befehl angegebenen Fehlerschwellenwert nicht überschritten. • Beim Targeting auf ein Tag war mindestens ein Aufruf erfolgreich, während bei anderen

Status	Details
	das Timeout abgelaufen ist. Der angegebene Fehlerschwellenwert gilt weiterhin. • Sie haben Anwendungen oder Richtlinien, die auf Betriebssystemebene durchgesetzt werden und die Ausführung eines Befehls verhindern oder überschreiben, was dazu führt, dass der Exit-Code Null zurückgegeben wird. **Note** Dieselben Bedingungen gelten für die Ausrichtung auf Ressourcengruppen. Um Fehler zu beheben oder weitere Informationen über die Befehlsau sführung zu erhalten, senden Sie einen Befehl, der Fehler oder Ausnahmen handhabt, indem er entsprechende Beendigungscodes (Ausgangscodes für fehlgeschlagenen Befehl (nicht null)) zurückgibt.
DeliveryTimedRaus	Der Befehl wurde nicht an den verwaltet en Knoten übermittelt, bevor die gesamte Zeitbeschränkung abgelaufen ist. Gesamtaus fälle werden nicht auf die übergeordnete Befehlsbegrenzung angerechnet `max-error s` , aber sie tragen dazu bei, ob der übergeord nete Befehlsstatus `Success`, `Incomplete` oder `Delivery Timed Out` ist. Diese ist ein Terminalstatus.

Status	Details
ExecutionTimedRaus	Die Befehlsautomatisierung begann auf dem verwalteten Knoten, aber der Befehl wurde vor Ablauf des Ausführungs-Timeouts nicht abgeschlossen. Ausführungs-Timeouts zählen als Fehler, wodurch eine Antwort ungleich Null gesendet wird, und Systems Manager beendet den Versuch, die Befehlsautomatisierung auszuführen, und meldet einen Fehlerstatus.
Fehlgeschlagen	Der Befehl war auf dem verwalteten Knoten nicht erfolgreich. Für ein Plug-in bedeutet dies, dass der Ergebniscode nicht null war. Für einen Befehlsaufruf bedeutet dies, dass der Ergebniscode für ein oder mehrere Plug-ins nicht null war. Zeitüberschreitungen beim Aufrufen werden auf das `max-errors` Limit des übergeordneten Befehls angerechnet. Diese ist ein Terminalstatus.
Abgebrochen	Der Befehl wurde beendet, bevor er abgeschlossen wurde. Diese ist ein Terminalstatus.

Status	Details
Unzustellbar	Der Befehl kann nicht an den verwalteten Knoten übermittelt werden. Der Knoten existiert möglicherweise nicht oder antwortet nicht. Unzustellbare Aufrufe werden nicht auf die übergeordnete Befehlsbegrenzung angerechnet `max-errors`, aber sie tragen dazu bei, ob der übergeordnete Befehlsstatus `Success` oder `Incomplete` ist. Wenn beispielsweise alle Aufrufe in einem Befehl den Status `Undeliverable` haben, lautet der zurückgegebene Befehlsstatus `Failed`. Wenn ein Befehl jedoch fünf Aufrufe hat, von denen vier den Status `Undeliverable` zurückgeben und einer den Status `Success` zurückgibt, lautet der Status des übergeordneten Befehls `Success`. Diese ist ein Terminalstatus.
Beendet	Der übergeordnete Befehl hat sein Limit `max-errors` überschritten, und nachfolgende Befehlsaufrufe wurden vom System abgebrochen. Diese ist ein Terminalstatus.

Status	Details
InvalidPlatform	Der Befehl wurde an einen verwalteten Knoten gesendet, der nicht den erforderlichen Plattformen entspricht, wie sie im ausgewähl ten Dokument festgelegt wurden. `Invalid Platform` wird nicht auf die maximale Fehlerbegrenzung des übergeordneten Befehls angerechnet, aber es trägt dazu bei, ob der übergeordnete Befehlsstatus Success oder Failed lautet. Wenn beispielsweise alle Aufrufe in einem Befehl den Status `Invalid Platform` haben, lautet der zurückgegebene Befehlsstatus `Failed`. Wenn ein Befehl jedoch fünf Aufrufe hat, von denen vier den Status `Invalid Platform` zurückgeben und einer den Status `Success` zurückgibt, lautet der Status des übergeordneten Befehls `Success`. Diese ist ein Terminalstatus.
AccessDenied	Der AWS Identity and Access Management (IAM-) Benutzer oder die Rolle, die den Befehl initiiert hat, hat keinen Zugriff auf den verwaltet en Zielknoten. `Access Denied`wird nicht auf das `max-errors` Limit des übergeordneten Befehls angerechnet, trägt aber dazu bei, ob der Status des übergeordneten Befehls oder lautet`Success`. `Failed` Wenn beispielsweise alle Aufrufe in einem Befehl den Status `Access Denied` haben, lautet der zurückgegebene Befehlsstatus `Failed`. Wenn ein Befehl jedoch fünf Aufrufe hat, von denen vier den Status `Access Denied` zurückgeben und einer den Status `Success` zurückgibt, lautet der Status des übergeordneten Befehls `Success`. Diese ist ein Terminalstatus.

Detaillierter Status für einen Befehl

Status	Details
Ausstehend	Der Befehl wurde noch von keinem Agenten auf einem verwalteten Knoten empfangen.
InProgress	Der Befehl wurde an mindestens einen verwalteten Knoten gesendet, hat aber keinen endgültigen Status auf allen Knoten erreicht.
Verzögert	Das System versuchte, den Befehl an den Knoten zu senden, war jedoch nicht erfolgreich. Das System startet einen erneuten Versuch.
Herzlichen Glückwunsch	Der Befehl wurde vom SSM Agent auf allen angegebenen oder anvisierten verwalteten Knoten empfangen und ein Ausgangscode von null wurde zurückgegeben. Alle Befehlsau frufe haben einen endgültigen Status erreicht, und der Wert von `max-errors` wurde nicht erreicht. Dieser Status bedeutet nicht, dass der Befehl auf allen angegebenen oder anvisiert en verwalteten Knoten erfolgreich verarbeitet wurde. Diese ist ein Terminalstatus. **Note** Um Fehler zu beheben oder weitere Informationen über die Befehlsau sführung zu erhalten, senden Sie einen Befehl, der Fehler oder Ausnahmen handhabt, indem er entsprechende Beendigungscodes (Ausgangscodes für fehlgeschlagenen Befehl (nicht null)) zurückgibt.

Status	Details
DeliveryTimedAus	Der Befehl wurde nicht an den verwaltet en Knoten übermittelt, bevor die gesamte Zeitbeschränkung abgelaufen ist. Der Wert `max-errors` oder weitere Befehlsaufrufe zeigen den Status `Delivery Timed Out`. Diese ist ein Terminalstatus.
Fehlgeschlagen	Der Befehl war auf dem verwalteten Knoten nicht erfolgreich. Der Wert `max-errors` oder weitere Befehlsaufrufe zeigen den Status `Failed`. Diese ist ein Terminalstatus.
Unvollständig	Der Befehl wurde auf allen verwalteten Knoten versucht und einer oder mehrere der Aufrufe haben nicht den Wert `Success`. Jedoch sind nicht genügend Aufrufe fehlgeschlagen für den Status `Failed`. Diese ist ein Terminalstatus.
Abgebrochen	Der Befehl wurde beendet, bevor er abgeschlo ssen wurde. Diese ist ein Terminalstatus.
RateExceeded	Die Anzahl der verwalteten Knoten, die durch den Befehl anvisiert wurden, überschritt das Kontingent Ihres Kontos für ausstehende Aufrufe. Das System hat den Befehl vor der Ausführung auf einem Knoten abgebrochen. Diese ist ein Terminalstatus.

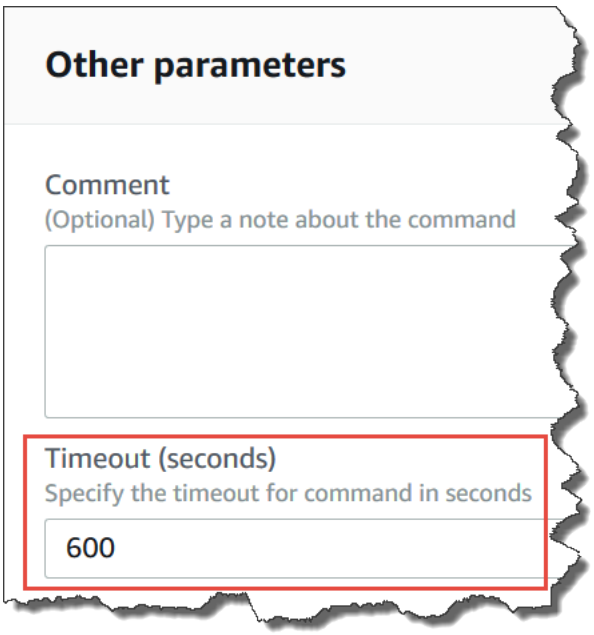
Status	Details
AccessDenied	Der Benutzer oder die Rolle, der oder die den Befehl initiiert, hat keinen Zugriff auf die Zielressourcengruppe. `AccessDenied` zählt nicht zum `max-errors` -Limit des übergeordneten Befehls, trägt aber dazu bei, ob der Status des übergeordneten Befehls `Success` oder `Failed` ist. (Wenn beispielsweise alle Aufrufe in einem Befehl den Status `AccessDenied` haben, dann lautet der zurückgegebene Befehlsstatus `Failed`. Wenn ein Befehl jedoch 5 Aufrufe hat, von denen 4 den Status `AccessDenied` anzeigen und 1 davon den Status `Success` anzeigt, dann lautet der Status des übergeordneten Befehls `Success`.) Diese ist ein Terminalstatus.
Keine Instances im Tag	Der Tag-Schlüsselpaar-Wert oder die Ressourcengruppe, auf die der Befehl ausgerichtet ist, stimmt mit keinem verwalteten Knoten überein. Diese ist ein Terminalstatus.

Informationen zu Timeout-Werten von Befehlen

Systems Manager erzwingt die folgenden Timeout-Werte bei der Ausführung von Befehlen.

Gesamt-Timeout

Geben Sie in der Systems-Manager-Konsole den Zeitbeschränkungs-Wert im Feld Timeout (seconds) (Zeitbeschränkung (Sekunden)) ein. Nachdem ein Befehl gesendet wurde, prüft Run Command, ob der Befehl abgelaufen ist oder nicht. Wenn ein Befehl das Ablauflimit des Befehls (Gesamtzeitlimit) erreicht, ändert er den Status in `DeliveryTimedOut` für alle Aufrufe, die den Status `InProgress`, `Pending` oder `Delayed` haben.



Technisch gesehen ist die gesamte Zeitbeschränkung (Timeout (Sekunden)) eine Kombination aus zwei Timeout-Werten, wie hier gezeigt:

```
Total timeout = "Timeout(seconds)" from the console + "timeoutSeconds": "{{ executionTimeout }}" from your SSM document
```

Beispielsweise beträgt der Standardwert von Timeout (seconds) (Timeout (Sekunden)) 600 Sekunden in der Systems Manager-Konsole. Wenn Sie einen Befehl mit dem `AWS-RunShellScript`-SSM-Dokument ausführen, beträgt der Standardwert von „timeoutSeconds“: „{{executionTimeout}}“ 3600 Sekunden, wie im folgenden Dokumentbeispiel gezeigt:

```
"executionTimeout": {
    "type": "String",
    "default": "3600",

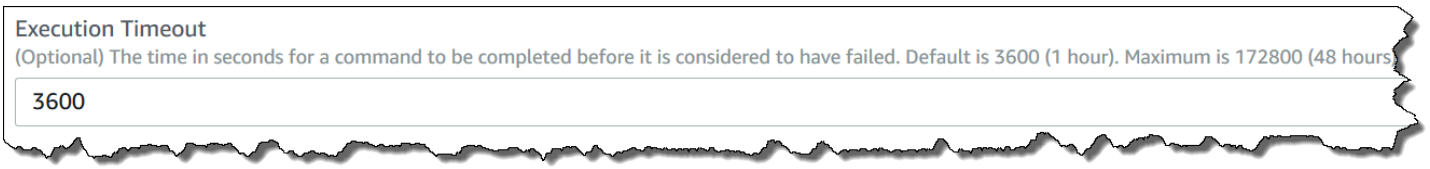
"runtimeConfig": {
  "aws:runShellScript": {
    "properties": [
      {
        "timeoutSeconds": "{{ executionTimeout }}"
```

Das bedeutet, dass der Befehl 4 200 Sekunden (70 Minuten) lang ausgeführt wird, bevor das System den Befehlsstatus auf `DeliveryTimedOut` setzt.

Execution Timeout

In der Systems Manager-Konsole geben Sie den Wert für die Ausführungszeitüberschreitung im Feld Execution Timeout an, sofern verfügbar. Nicht alle SSM-Dokumente erfordern die Angabe eines Ausführungs-Timeout. Das Feld Execution Timeout (Ausführungszeitlimit) wird nur angezeigt, wenn ein entsprechender Eingabeparameter im SSM-Dokument definiert wurde. Falls angegeben, muss der Befehl innerhalb dieser Zeitspanne abgeschlossen werden.

> **Note**
>
> Run Command stützt sich auf die SSM Agent-Dokument-Terminalantwort, um zu bestimmen, ob der Befehl an den Agenten übermittelt wurde oder nicht. SSM Agent muss ein `ExecutionTimedOut`-Signal senden, damit ein Aufruf oder Befehl als `ExecutionTimedOut` markiert wird.



Standard-Ausführungs-Timeout

Wenn ein SSM-Dokument nicht erfordert, dass Sie explizit einen Ausführungs-Timeout-Wert angeben, erzwingt Systems Manager den fest programmierten Standard-Ausführungs-Timeout.

Wie Systems Manager Timeouts meldet

Empfängt Systems Manager eine `execution timeout`-Antwort von SSM Agent auf einem Ziel, dann markiert Systems Manager den Befehlsaufruf als `executionTimeout`.

Erhält Run Command keine Dokumentterminalantwort von SSM Agent, wird der Befehlsaufruf als `deliveryTimeout`gekennzeichnet.

Um den Timeout-Status für ein Ziel zu bestimmen, kombiniert SSM Agent alle Parameter und den Inhalt des SSM-Dokuments, um `executionTimeout` zu berechnen. Wenn SSM Agent feststellt, dass ein Befehl einen Timeout hat, sendet es `executionTimeout` an den Service.

Der Standardwert für Timeout (seconds) (Timeout (Sekunden)) beträgt 3600 Sekunden. Der Standardwert für Execution Timeout beträgt ebenfalls 3600 Sekunden. Daher beträgt die gesamte Standard-Timeout für einen Befehl 7200 Sekunden.

> **Note**
>
> SSM Agent verarbeitet `executionTimeout` unterschiedlich, je nach Art des SSM-Dokuments und der Dokumentversion.

Walkthroughs für Run Command

Die Anleitungen in diesem Abschnitt zeigen die Ausführung von Befehlen mit Run Command, eine Funktion von AWS Systems Manager, AWS Command Line Interface (AWS CLI) oder AWS Tools for Windows PowerShell.

Inhalt

- Aktualisierung von Software mithilfe von Run Command
- Anleitung: Verwenden der AWS CLI mit Run Command
- Exemplarische Vorgehensweise: Verwenden Sie das mit AWS Tools for Windows PowerShellRun Command

Sie können auch Beispiele für Befehle in den folgenden Referenzen anzeigen.

- Systems ManagerAWS CLI-Referenz
- AWS Tools for Windows PowerShell - AWS Systems Manager

Aktualisierung von Software mithilfe von Run Command

In den folgenden Verfahren wird beschrieben, wie Sie Software auf Ihren verwalteten Knoten aktualisieren.

Aktualisierung von SSM Agent mithilfe von Run Command

Im folgenden Verfahren wird beschrieben, wie Sie den SSM Agent auf Ihren verwalteten Knoten aktualisieren können. Sie können entweder auf die neueste Version des SSM Agent aktualisieren oder ein Downgrade auf eine ältere Version durchführen. Wenn Sie den Befehl ausführen, lädt das System die Version von herunter AWS, installiert sie und deinstalliert dann die Version, die vor der Ausführung des Befehls vorhanden war. Wenn während dieses Prozesses ein Fehler auftritt, wird das System auf die Version auf dem Server zurückgesetzt, bevor der Befehl ausgeführt wurde, und der Befehlsstatus zeigt, dass der Befehl fehlgeschlagen ist.

> **Note**
>
> Wenn auf einer Instance macOS-Version 11.0 (Big Sur) oder höher ausgeführt wird, muss die Instance über die SSM Agent-Version 3.1.941.0 oder höher verfügen, um das AWS-UpdateSSMAgent-Dokument auszuführen. Wenn auf der Instance eine Version von SSM Agent ausgeführt wird, die vor 3.1.941.0 veröffentlicht wurde, können Sie Ihr SSM Agent aktualisieren, um das AWS-UpdateSSMAgent-Dokument auszuführen, indem Sie die `brew update`- und `brew upgrade amazon-ssm-agent`-Befehle ausführen.

Um über SSM Agent Updates informiert zu werden, abonnieren Sie die Seite mit den SSM AgentVersionshinweisen unterGitHub.

So aktualisieren Sie SSM Agent mittels Run Command

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option **`AWS-UpdateSSMAgent`** aus.
5. Geben Sie im Abschnitt Command parameters ggf. Werte für die folgenden Parameter an:
   a. (Optional) Geben Sie in Version (Version) die zu installierende SSM Agent-Version ein. Sie können ältere Versionen des Agenten installieren. Wenn Sie keine Version angeben, installiert der Service die neueste Version.
   b. (Optional) Wählen Sie in Allow Downgrade (Downgrade erlauben) die Option true (wahr) aus, um eine frühere SSM Agent-Version zu installieren. Wenn Sie diese Option auswählen, geben Sie die frühere Versionsnummer an. Wählen Sie false, um nur die neueste Version des Dienstes zu installieren.
6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Other parameters (Weitere Parameter):

   - Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

10. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Wählen Sie Ausführen aus.

Aktualisierung PowerShell mit Run Command

Im folgenden Verfahren wird beschrieben, wie Sie auf Ihren verwalteten R2-Knoten Windows Server 2012 und 2012 auf Version 5.1 aktualisieren PowerShell . Das in diesem Verfahren bereitgestellte Skript lädt das Update für Windows Management Framework (WMF) Version 5.1 herunter und startet die Installation des Updates. Der Knoten wird während dieses Prozesses neu gestartet, da dies bei der Installation von WMF 5.1 erforderlich ist. Download und Installation des Updates dauern insgesamt etwa fünf Minuten.

Um zu aktualisieren PowerShell mit Run Command

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.

4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option **AWS-RunPowerShellScript** aus.
5. Fügen Sie die folgenden Befehle für Ihr Betriebssystem im Abschnitt Befehle ein.

   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. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Other parameters (Weitere Parameter):

   - Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Rate control (Ratenregelung):

- Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

  > **Note**
  >
  > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

10. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Wählen Sie Ausführen aus.

Nachdem der verwaltete Knoten neu gestartet wurde und die Installation des Updates abgeschlossen ist, stellen Sie eine Verbindung zu Ihrem Knoten her, um zu überprüfen, ob das Upgrade auf Version PowerShell 5.1 erfolgreich abgeschlossen wurde. Um die Version von PowerShell auf Ihrem Node zu überprüfen, öffnen Sie die Datei PowerShell und geben Sie die Eingabetaste ein`$PSVersionTable`. Der `PSVersion`-Wert in der Ausgabetabelle zeigt 5.1, wenn das Upgrade erfolgreich war.

Wenn der `PSVersion`-Wert nicht 5.1 ist, zum Beispiel 3.0 oder 4.0, überprüfen Sie die Setup-Protokolle im Event Viewer unter Windows-Protokolle. Diese Protokolle geben an, warum die Update-Installation fehlgeschlagen ist.

Anleitung: Verwenden der AWS CLI mit Run Command

Der folgende Beispiel-Walkthrough zeigt, wie Sie mithilfe der AWS Command Line Interface (AWS CLI) Informationen über Befehle und Befehlsparameter anzeigen, wie Sie Befehle ausführen und wie Sie den Status dieser Befehle anzeigen.

> **Important**
>
> Nur vertrauenswürdige Administratoren sollten AWS Systems Manager-vorkonfigurierte Dokumente in diesem Thema verwenden dürfen. Die in Systems-Manager-Dokumenten festgelegten Befehle oder Skripts werden mit Administratorberechtigungen auf Ihren verwalteten Knoten ausgeführt. Wenn ein Benutzer über die Berechtigung zum Ausführen der vordefinierten Systems-Manager-Dokumente (alle Dokumente, die mit `AWS-` beginnen) verfügt, hat dieser Benutzer auch Administratorzugriff auf den Knoten. Für alle anderen Benutzer sollten Sie restriktive Dokumente erstellen und sie mit bestimmten Benutzern teilen.

Themen

- Schritt 1: Erste Schritte
- Schritt 2: Ausführen von Shell-Skripten zum Anzeigen von Ressourcendetails
- Schritt 3: Senden einfacher Befehle mit dem AWS-RunShellScript-Dokument

- Schritt 4: Ausführen eines einfachen Python-Skripts mit Run Command
- Schritt 5: Führen Sie ein Bash-Skript mit Run Command aus

Schritt 1: Erste Schritte

Sie müssen entweder über Administratorberechtigungen auf dem verwalteten Knoten verfügen, den Sie konfigurieren möchten, oder Sie müssen über die geeignete Berechtigung in AWS Identity and Access Management (IAM) verfügen. Beachten Sie auch: das Beispiel verwendet die Region USA Ost (Ohio) (us-east-2). Run Command ist in den AWS-Regionen verfügbar, die unter Systems Manager Service-Endpunkten im Allgemeine Amazon Web Services-Referenz aufgelistet sind. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

So führen Sie Befehle mithilfe der AWS CLI aus

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Listen Sie alle verfügbaren Dokumente auf.

   Mit diesem Befehl werden alle verfügbaren Dokumente für Ihr Konto basierend auf IAM-Berechtigungen ausgeführt.

   ```
   aws ssm list-documents
   ```
3. Überprüfen Sie, ob ein verwalteter Knoten zum Empfangen von Befehlen bereit ist.

   Die Ausgabe des folgenden Befehls zeigt, ob verwaltete Knoten online sind.

   Linux & macOS

   ```
   aws ssm describe-instance-information \
       --output text --query "InstanceInformationList[*]"
   ```

   Windows

   ```
   aws ssm describe-instance-information ^
       --output text --query "InstanceInformationList[*]"
   ```

4. Verwenden Sie den folgenden Befehl, um weitere Details zu einem bestimmten verwalteten Knoten anzuzeigen.

> **Note**
>
> Ersetzen Sie die Instance- und Befehls-IDs zur Ausführung der Befehle in dieser Anleitung Für verwaltete AWS IoT Greengrass-Core-Geräte, benutzen Sie die mi-*ID_Nummer* als Instance-ID. Die Befehls-ID wird als Antwort an send-command zurückgegeben. Instance-IDs sind verfügbar unter Fleet Manager, eine Funktion von 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
```

Schritt 2: Ausführen von Shell-Skripten zum Anzeigen von Ressourcendetails

Mit Run Command und dem `AWS-RunShellScript`-Dokument können Sie Befehle oder Skripts auf einem verwalteten Knoten ausführen, als ob Sie lokal angemeldet wären.

Anzeigen der Beschreibung und verfügbaren Parameter

Führen Sie den folgenden Befehl aus, um eine Beschreibung des Systems Manager JSON-Dokuments anzuzeigen.

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]"
```

Führen Sie den folgenden Befehl aus, um die verfügbaren Parameter und Details zu diesen Parametern anzuzeigen.

Linux & macOS

```
aws ssm describe-document \
    --name "AWS-RunShellScript" \
    --query "Document.Parameters[*]"
```

Windows

```
aws ssm describe-document ^
    --name "AWS-RunShellScript" ^
    --query "Document.Parameters[*]"
```

Schritt 3: Senden einfacher Befehle mit dem **`AWS-RunShellScript`**-Dokument

Führen Sie den folgenden Befehl aus, um IP-Informationen für einen Linux-verwalteten Knoten abzurufen.

Wenn Sie einen von Windows Server verwalteten Knoten anvisieren, ändern Sie den `document-name` zu `AWS-RunPowerShellScript` und ändern Sie den `command` von `ifconfig` zu `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
```

Abrufen der Befehlsinformation mit Antwortdaten

Mit dem folgenden Befehl wird die Befehls-ID verwendet, die vom vorherigen Befehl zurückgegeben wurde, um die Details und Antwortdaten der Ausführung des Befehls abzurufen. Das System gibt die Antwortdaten zurück, wenn der Befehl abgeschlossen ist. Wenn die Befehlsausführung `"Pending"` oder `"InProgress"` anzeigt, führen Sie diesen Befehl erneut aus, um die Antwortdaten zu sehen.

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
```

Benutzer identifizieren

Mit dem folgenden Befehl wird der Standard-Benutzer angezeigt, der die Befehle ausführt.

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")
```

Abrufen des Befehlsstatus

Mit dem folgenden Befehl wird die Befehls-ID verwendet, um den Status der Befehlsausführung auf dem verwalteten Knoten abzurufen. In diesem Beispiel wird die Befehls-ID verwendet, die im vorherigen Befehl zurückgegeben wurde.

Linux & macOS

```
aws ssm list-commands \
    --command-id "command-ID"
```

Windows

```
aws ssm list-commands ^
    --command-id "command-ID"
```

Abrufen der Befehlsdetails

Mit dem folgenden Befehl wird die Befehls-ID vom vorherigen Befehl verwendet, um den Status der Befehlsausführung pro verwalteten Knoten abzurufen.

Linux & macOS

```
aws ssm list-command-invocations \
    --command-id "command-ID" \
    --details
```

Windows

```
aws ssm list-command-invocations ^
    --command-id "command-ID" ^
    --details
```

Abrufen von Befehlsinformationen mit Antwortdaten für einen bestimmten verwalteten Knoten

Mit dem folgenden Befehl wird die Ausgabe der ursprünglichen `aws ssm send-command`-Anforderung für einen bestimmten verwalteten Knoten zurückgegeben.

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
```

Anzeigen der Python-Version

Mit dem folgenden Befehl wird die Version von Python zurückgegeben, die auf einem Knoten ausgeführt wird.

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}"'
```

Schritt 4: Ausführen eines einfachen Python-Skripts mit Run Command

Mit dem folgenden Befehl wird ein einfaches Python-Skript „Hello World“ unter Verwendung von Run Command ausgeführt.

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}"'
```

Schritt 5: Führen Sie ein Bash-Skript mit Run Command aus

Die Beispiele in diesem Abschnitt zeigen, wie Sie das folgende Bash-Skript mit Run Commandausführen.

Für Beispiele für die Verwendung von Run Command, um Skripts auszuführen, die an Remote-Speicherorten gespeichert sind, siehe Ausführen von Skripts von Amazon S3 und Ausführen von Skripts von 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
```

Dieses Skript installiert den AWS CodeDeploy-Agenten auf Amazon Linux und Red Hat Enterprise Linux(RHEL)-Instances, wie in Create an Amazon EC2 instance for CodeDeploy im AWS CodeDeploy-Benutzerhandbuch beschrieben.

Das Skript installiert den CodeDeploy-Agent von einem von AWSverwalteten S3-Bucket in der Region USA Ost (Ohio) (us-east-2), `aws-codedeploy-us-east-2`.

Führen Sie ein Bash-Skript in einem AWS CLI-Befehl aus

Das folgende Beispiel zeigt, wie Sie das Bash-Skript mit der Option `--parameters` in einen CLI-Befehl einbinden.

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"]}'
```

Führen Sie ein Bash-Skript in einer JSON-Datei aus

Im folgenden Beispiel wird der Inhalt des Bash-Skripts in einer JSON-Datei gespeichert, und die Datei wird mit der Option `--cli-input-json` in den Befehl aufgenommen.

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
```

Der Inhalt der referenzierten `installCodeDeployAgent.json`-Datei ist im folgenden Beispiel dargestellt.

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

Exemplarische Vorgehensweise: Verwenden Sie das mit AWS Tools for Windows PowerShellRun Command

In den folgenden Beispielen wird gezeigt, wie Sie mithilfe von Informationen AWS Tools for Windows PowerShell zu Befehlen und Befehlsparametern anzeigen, Befehle ausführen und den Status dieser Befehle anzeigen können. Diese Anleitung umfasst ein Beispiel für jedes der vordefinierten AWS Systems Manager -Dokumente.

> **Important**
>
> Nur vertrauenswürdige Administratoren sollten Systems Manager-vorkonfigurierte Dokumente in diesem Thema verwenden dürfen. Die in Systems-Manager-Dokumenten festgelegten Befehle oder Skripts werden mit einer Administratorberechtigung auf Ihren verwalteten Knoten ausgeführt. Wenn ein Benutzer berechtigt ist, eines der vordefinierten Systems Manager Manager-Dokumente (jedes Dokument, das mit beginnt AWS) auszuführen, hat dieser Benutzer auch Administratorzugriff auf den Knoten. Für alle anderen Benutzer sollten Sie restriktive Dokumente erstellen und sie mit bestimmten Benutzern teilen.

Themen

- Konfigurieren Sie die AWS Tools for Windows PowerShell Sitzungseinstellungen
- Listen Sie alle verfügbaren Dokumente auf
- Führen Sie PowerShell Befehle oder Skripts aus
- Installieren einer Anwendung mithilfe des AWS-InstallApplication-Dokuments
- Installieren Sie ein PowerShell Modul mithilfe des AWS-InstallPowerShellModule JSON-Dokuments
- Verbinden eines verwalteten Knotens mit einer Domain mithilfe des AWS-JoinDirectoryServiceDomain-JSON-Dokuments
- Senden Sie Windows-Metriken mithilfe des AWS-ConfigureCloudWatch Dokuments an Amazon CloudWatch Logs

- Aktualisieren von EC2Config mit dem AWS-UpdateEC2Config-Document
- Aktivieren oder deaktivieren Sie die automatische Windows-Aktualisierung mithilfe des AWS-ConfigureWindowsUpdate-Dokuments.
- Verwalten von Windows-Updates mit Run Command

Konfigurieren Sie die AWS Tools for Windows PowerShell Sitzungseinstellungen

Angeben Ihrer Anmeldeinformationen

Öffnen Sie Tools für Windows PowerShell auf Ihrem lokalen Computer und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen anzugeben. Sie müssen entweder über Administratorrechte für die verwalteten Knoten verfügen, die Sie konfigurieren möchten, oder Ihnen müssen die entsprechenden Berechtigungen in AWS Identity and Access Management (IAM) erteilt worden sein. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

```
Set-AWSCredentials –AccessKey key-name –SecretKey key-name
```

Legen Sie einen Standard fest AWS-Region

Führen Sie den folgenden Befehl aus, um die Region für Ihre PowerShell Sitzung festzulegen. Das Beispiel verwendet die Region USA Ost (Ohio) (us-east-2). Run Commandist in den in Systems Manager AWS-Regionen aufgelisteten Dienstendpunkten in der Allgemeine Amazon Web Services-Referenzverfügbar.

```
Set-DefaultAWSRegion `
    -Region us-east-2
```

Listen Sie alle verfügbaren Dokumente auf

Mit diesem Befehl werden alle verfügbaren Dokumente für Ihrem Konto aufgelistet.

```
Get-SSMDocumentList
```

Führen Sie PowerShell Befehle oder Skripts aus

Mit Run Command und dem `AWS-RunPowerShell`-Dokument können Sie Befehle oder Skripts auf einem verwalteten Knoten ausführen, als ob Sie lokal angemeldet wären. Sie können Befehle ausgeben oder einen Pfad zu einem lokalen Skript eingeben, um den Befehl auszuführen.

> **Note**
>
> Informationen zum Neustarten von verwalteten Knoten bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Anzeigen der Beschreibung und verfügbaren Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-RunPowerShellScript"
```

Anzeigen weiterer Informationen über Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-RunPowerShellScript" | Select -ExpandProperty Parameters
```

Senden Sie einen Befehl mithilfe des **`AWS-RunPowerShellScript`**-Dokuments

Mit dem folgenden Befehl werden der Inhalt des `"C:\Users"`-Verzeichnisses und der Inhalt des `"C:\"`-Verzeichnisses auf zwei verwalteten Knoten angezeigt.

```
$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:\')}
```

Abrufen der Befehlsabfragedetails

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung auf beiden verwalteten Knoten abzurufen. In diesem Beispiel wird die `CommandId` verwendet, die im vorherigen Befehl zurückgegeben wurde.

```
Get-SSMCommand `
    -CommandId $runPSCommand.CommandId
```

Der Status des Befehls in diesem Beispiel kann „Erfolgreich“, „Ausstehend“ oder „Ausstehend“ lauten InProgress.

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` vom vorherigen Befehl verwendet, um den Status der Befehlsausführung pro verwalteten Knoten abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $runPSCommand.CommandId
```

Abrufen von Befehlsinformationen mit Antwortdaten für einen bestimmten verwalteten Knoten

Mit dem folgenden Befehl wird die Ausgabe des ursprünglichen `Send-SSMCommand` für einen bestimmten verwalteten Knoten zurückgegeben.

```
Get-SSMCommandInvocation `
    -CommandId $runPSCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Abbrechen eines Befehls

Mit dem folgenden Befehl wird `Send-SSMCommand` für das `AWS-RunPowerShellScript`-Dokument abgebrochen.

```
$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
```

Überprüfen des Befehlsstatus

Mit dem folgenden Befehl wird der Status des `Cancel`-Befehls überprüft

```
Get-SSMCommand `
    -CommandId $cancelCommand.CommandId
```

Installieren einer Anwendung mithilfe des **`AWS-InstallApplication`**-Dokuments

Mit Run Command und dem `AWS-InstallApplication`-Dokument können Sie Anwendungen auf verwalteten Knoten installieren, reparieren oder deinstallieren. Der Befehl erfordert den Pfad oder die Adresse für ein MSI.

> **Note**
>
> Informationen zum Neustarten von verwalteten Knoten bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Anzeigen der Beschreibung und verfügbaren Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallApplication"
```

Anzeigen weiterer Informationen über Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallApplication" | Select -ExpandProperty Parameters
```

Senden Sie einen Befehl mithilfe des **`AWS-InstallApplication`**-Dokuments

Mit dem folgenden Befehl wird eine Version von Python auf Ihrerm verwalteten Knoten im unbeaufsichtigten Modus installiert und die Ausgabe in einer lokalen Textdatei auf dem Laufwerk `C:` protokolliert.

```
$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'}
```

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $installAppCommand.CommandId `
    -Details $true
```

Abrufen von Befehlsinformationen mit Antwortdaten für einen bestimmten verwalteten Knoten

Mit dem folgenden Befehl werden die Ergebnisse der Python-Installation zurückgegeben.

```
Get-SSMCommandInvocation `
    -CommandId $installAppCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Installieren Sie ein PowerShell Modul mithilfe des **`AWS-InstallPowerShellModule`** JSON-Dokuments

Sie können es verwendenRun Command, um PowerShell Module auf verwalteten Knoten zu installieren. Weitere Informationen zu PowerShell Modulen finden Sie unter PowerShell Windows-Module.

Anzeigen der Beschreibung und verfügbaren Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallPowerShellModule"
```

Anzeigen weiterer Informationen über Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallPowerShellModule" | Select -ExpandProperty Parameters
```

Installieren Sie ein PowerShell Modul

Mit dem folgenden Befehl wird die Datei EZOut.zip heruntergeladen, installiert und anschließend wird ein zusätzlicher Befehl zum Installieren von XPS Viewer ausgeführt. Schließlich wird die Ausgabe dieses Befehls in einen S3-Bucket mit dem Namen "demo-ssm-output-bucket" hochgeladen.

```
$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 demo-ssm-output-bucket
```

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $installPSCommand.CommandId `
    -Details $true
```

Abrufen von Befehlsinformationen mit Antwortdaten für den verwalteten Knoten

Mit dem folgenden Befehl wird die Ausgabe des ursprünglichen `Send-SSMCommand`-Befehls für die spezielle `CommandId` zurückgegeben.

```
Get-SSMCommandInvocation `
    -CommandId $installPSCommand.CommandId `
    -Details $true | Select -ExpandProperty CommandPlugins
```

Verbinden eines verwalteten Knotens mit einer Domain mithilfe des **`AWS-JoinDirectoryServiceDomain`**-JSON-Dokuments

Mit Run Command dieser Option können Sie einen verwalteten Knoten schnell einer AWS Directory Service Domäne hinzufügen. Erstellen Sie ein Verzeichnis vor dem Ausführen dieses Befehls. Wir empfehlen außerdem, sich mit der AWS Directory Service besser vertraut zu machen. Weitere Informationen finden Sie im Administrationshandbuch zu AWS Directory Service.

Sie können nur einen verwalteten Knoten mit einer Domain verbinden. Sie können keinen Knoten aus einer Domain entfernen.

> **Note**
>
> Informationen zu verwalteten Knoten bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Anzeigen der Beschreibung und verfügbaren Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-JoinDirectoryServiceDomain"
```

Anzeigen weiterer Informationen über Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-JoinDirectoryServiceDomain" | Select -ExpandProperty Parameters
```

Verbinden eines verwalteten Knotens mit einer Domain

Der folgende Befehl verbindet einen verwalteten Knoten mit der angegebenen AWS Directory Service Domain und lädt alle generierten Ausgaben in den Amazon Simple Storage Service (Amazon S3) -Beispiel-Bucket hoch.

```
$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 demo-ssm-output-bucket
```

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $domainJoinCommand.CommandId `
    -Details $true
```

Abrufen von Befehlsinformationen mit Antwortdaten für den verwalteten Knoten

Dieser Befehl gibt die Ausgabe des ursprünglichen `Send-SSMCommand` für die spezifische `CommandId` zurück.

```
Get-SSMCommandInvocation `
    -CommandId $domainJoinCommand.CommandId `
```

```
    -Details $true | Select -ExpandProperty CommandPlugins
```

Senden Sie Windows-Metriken mithilfe des **AWS-ConfigureCloudWatch** Dokuments an Amazon CloudWatch Logs

Sie können Windows Server Nachrichten in den Anwendungs-, System-, Sicherheits- und Event Tracing for Windows (ETW) -Protokollen an Amazon CloudWatch Logs senden. Wenn Sie die Protokollierung zum ersten Mal aktivieren, sendet Systems Manager alle Protokolle, die innerhalb von 1 Minute generiert werden, sobald Sie mit dem Hochladen von Protokollen für die Anwendungs-, System-, Sicherheits- und ETW-Protokolle beginnen. Protokolle, die davor auftraten, werden nicht berücksichtigt. Wenn Sie die Protokollierung deaktivieren und später wieder aktivieren, sendet Systems Manager Protokolle ab dem Zeitpunkt, an dem die Unterbrechung stattfand. Für alle benutzerdefinierten Protokolldateien und IIS- (Internet Information Services)-Protokolle liest Systems Manager die Protokolldateien von Anfang an. Darüber hinaus kann Systems Manager auch Leistungsindikatordaten an CloudWatch Logs senden.

Wenn Sie zuvor die CloudWatch Integration in EC2Config aktiviert haben, überschreiben die Systems Manager Manager-Einstellungen alle Einstellungen, die lokal auf dem verwalteten Knoten in der `C:\Program Files\Amazon\EC2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json` Datei gespeichert sind. Weitere Informationen zur Verwendung von EC2Config zur Verwaltung von Leistungsindikatoren und Protokollen auf einem einzelnen verwalteten Knoten finden Sie unter Erfassung von Metriken und Protokollen von Amazon EC2 EC2-Instances und lokalen Servern mit dem CloudWatch Agenten im Amazon-Benutzerhandbuch. CloudWatch

Anzeigen der Beschreibung und verfügbaren Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureCloudWatch"
```

Anzeigen weiterer Informationen über Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureCloudWatch" | Select -ExpandProperty Parameters
```

Senden Sie Anwendungsprotokolle an CloudWatch

Mit dem folgenden Befehl wird der verwaltete Knoten konfiguriert und die Windows-Anwendungsprotokolle dorthin CloudWatch verschoben.

```
$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"]}}}'}
```

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $cloudWatchCommand.CommandId `
    -Details $true
```

Abrufen von Befehlsinformationen mit Antwortdaten für einen bestimmten verwalteten Knoten

Der folgende Befehl gibt die Ergebnisse der CloudWatch Amazon-Konfiguration zurück.

```
Get-SSMCommandInvocation `
    -CommandId $cloudWatchCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Senden Sie Leistungsindikatoren an die CloudWatch Verwendung des Dokuments **`AWS-ConfigureCloudWatch`**

Mit dem folgenden Demonstrationsbefehl werden Leistungsindikatoren in hochgeladen. CloudWatch Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

```
$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"]}}}'}
```

Aktualisieren von EC2Config mit dem **`AWS-UpdateEC2Config`**-Document

Mit Run Command und dem `AWS-EC2ConfigUpdate`-Dokument können Sie den EC2Config-Service aktualisieren, der auf Ihren von Windows Server verwalteten Knoten ausgeführt wird. Mit diesem Befehl kann der EC2Config-Service auf die neueste Version oder eine von Ihnen angegebene Version aktualisiert werden.

Anzeigen der Beschreibung und verfügbaren Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-UpdateEC2Config"
```

Anzeigen weiterer Informationen über Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-UpdateEC2Config" | Select -ExpandProperty Parameters
```

Aktualisieren von EC2Config auf die neueste Version

```
$ec2ConfigCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-UpdateEC2Config"
```

Abrufen von Befehlsinformationen mit Antwortdaten für den verwalteten Knoten

Dieser Befehl gibt die Ausgabe des angegebenen Befehls aus dem vorherigen `Send-SSMCommand` zurück.

```
Get-SSMCommandInvocation `
    -CommandId $ec2ConfigCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Aktualisieren von EC2Config auf eine bestimmte Version

Mit dem folgenden Befehl wird EC2Config auf eine ältere Version runtergestuft.

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-UpdateEC2Config" `
    -Parameter @{'version'='4.9.3519'; 'allowDowngrade'='true'}
```

Aktivieren oder deaktivieren Sie die automatische Windows-Aktualisierung mithilfe des **`AWS-ConfigureWindowsUpdate`**-Dokuments.

Mit Run Command und dem `AWS-ConfigureWindowsUpdate`-Dokument können Sie automatische Windows-Updates auf Ihren von Windows Server verwalteten Knoten aktivieren oder deaktivieren. Mit diesem Befehl wird der Windows Update-Agent konfiguriert, um Windows-Updates an dem Tag und in der Stunde, die Sie angeben, herunterzuladen und zu installieren. Wenn ein Update einen Neustart erfordert, startet der verwaltete Knoten automatisch 15 Minuten nach der Installation der Updates neu. Mit diesem Befehl können Sie konfigurieren, dass Windows Update auf Updates prüft, diese aber nicht installiert. Das `AWS-ConfigureWindowsUpdate`-Dokument ist mit Windows Server 2008, 2008 R2, 2012, 2012 R2 und 2016 kompatibel.

Anzeigen der Beschreibung und verfügbaren Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureWindowsUpdate"
```

Anzeigen weiterer Informationen über Parameter

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureWindowsUpdate" | Select -ExpandProperty Parameters
```

Aktivieren des automatischen Windows Updates

Mit dem folgenden Befehl wird Windows Update konfiguriert, um Updates automatisch täglich um 22.00 Uhr herunterzuladen und zu installieren.

```
$configureWindowsUpdateCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-ConfigureWindowsUpdate" `
```

```
    -Parameters @{'updateLevel'='InstallUpdatesAutomatically';
 'scheduledInstallDay'='Daily'; 'scheduledInstallTime'='22:00'}
```

Anzeigen des Befehlsstatus zum Aktivieren von automatischen Windows Updates

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung für die Aktivierung von Windows Automatic Updates abzurufen.

```
Get-SSMCommandInvocation `
    -Details $true `
    -CommandId $configureWindowsUpdateCommand.CommandId | Select -ExpandProperty
 CommandPlugins
```

Deaktivieren des automatischen Windows Updates

Mit dem folgenden Befehl wird die Windows-Update-Benachrichtigungsebene herabgesetzt, damit das System prüft, ob Updates vorliegen, diese jedoch nicht automatisch auf dem verwalteten Knoten installiert.

```
$configureWindowsUpdateCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-ConfigureWindowsUpdate" `
    -Parameters @{'updateLevel'='NeverCheckForUpdates'}
```

Anzeigen des Befehlsstatus zum Deaktivieren von automatischen Windows Updates

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung für die Aktivierung von Windows Automatic Updates abzurufen.

```
Get-SSMCommandInvocation `
    -Details $true `
    -CommandId $configureWindowsUpdateCommand.CommandId | Select -ExpandProperty
 CommandPlugins
```

Verwalten von Windows-Updates mit Run Command

Mit Run Command und dem `AWS-InstallWindowsUpdates`-Dokument können Sie Updates für von Windows Server verwaltete Knoten verwalten. Dieser Befehl scannt nach oder installiert fehlende Updates auf Ihren verwalteten Knoten und führt nach der Installation optional einen Neustart durch.

Sie können auch die entsprechenden Klassifizierungen und Schweregrade für Aktualisierungen angeben, die in Ihrer Umgebung installiert werden sollen.

> **Note**
>
> Informationen zum Neustarten von verwalteten Knoten bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Die folgenden Beispiele zeigen, wie Sie die angegebenen Windows Update-Verwaltungsaufgaben durchführen.

Suche nach allen fehlenden Windows-Updates

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Scan'}
```

Installieren von bestimmten Windows-Updates

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Install';'IncludeKbs'='kb-ID-1,kb-ID-2,kb-
ID-3';'AllowReboot'='True'}
```

Installieren wichtiger fehlender Windows-Updates

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Install';'SeverityLevels'='Important';'AllowReboot'='True'}
```

Installieren fehlender Windows-Updates mit bestimmten Ausschlüssen

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
```

```
    -Parameters @{'Action'='Install';'ExcludeKbs'='kb-ID-1,kb-
ID-2';'AllowReboot'='True'}
```

Fehlerbehebung von Systems Manager Run Command

Run Command, eine Funktion von AWS Systems Manager, gibt bei jeder Befehlsausführung Statusdetails an. Weitere Informationen zu den Befehlsstatus-Details finden Sie unter Grundlegendes zu Befehlsstatus. Sie können auch die Informationen in diesem Thema verwenden, um Probleme mit Run Command zu beheben.

Themen

- Einige meiner verwalteten Knoten fehlen
- Ein Schritt in meinem Skript ist fehlgeschlagen, der Gesamtstatus wird jedoch als "Succeeded" (Erfolgreich) angezeigt.
- SSM Agent wird nicht ordnungsgemäß ausgeführt.

Einige meiner verwalteten Knoten fehlen

Auf der Seite Run a command (Einen Befehl ausführen) können Sie, nachdem Sie ein auszuführendes SSM-Dokument ausgewählt und im Abschnitt Targets (Ziele) die manuelle Auswahl von Instances gewählt haben, wird eine Liste von verwalteten Knoten angezeigt, die Sie für die Ausführung des Befehls auswählen können.

Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

Nach der Erstellung, der Aktivierung, dem erneuten Hochfahren oder dem Neustart eines verwalteten Knotens, dem Installieren von Run Command auf einem Knoten oder dem Anfügen eines AWS Identity and Access Management (IAM)-Instance-Profils an einen Knoten, kann es einige Minuten dauern, bis der verwaltete Knoten zur Liste hinzugefügt wird.

Ein Schritt in meinem Skript ist fehlgeschlagen, der Gesamtstatus wird jedoch als "Succeeded" (Erfolgreich) angezeigt.

Mit Run Command können Sie festlegen, wie Ihre Skripte mit Exit-Codes umgehen. Standardmäßig wird der Beendigungscode des letzten in einem Skript ausgeführten Befehls als Beendigungscode für das gesamte Skript gemeldet. Sie können jedoch eine bedingte Anweisung einschließen, damit das

Skript beendet wird, wenn ein Befehl vor dem letzten Befehl fehlschlägt. Weitere Informationen und Beispiele finden Sie unter Angabe von Beendigungscodes in Befehlen.

SSM Agent wird nicht ordnungsgemäß ausgeführt.

Bei Problemen beim Ausführen von Befehlen mittels Run Command liegt möglicherweise ein Problem mit SSM Agent vor. Informationen zum Untersuchen von Problemen mit SSM Agent finden Sie unter Fehlerbehebung für SSM Agent.

AWS Systems Manager State Manager

State Manager, eine Fähigkeit von AWS Systems Manager, ist ein sicherer und skalierbarer Konfigurationsmanagement-Service, der den Prozess automatisiert, Ihre verwalteten Knoten und andere AWS Ressourcen in einem von Ihnen definierten Zustand zu halten. Um mit State Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich State Manager aus.

> **Note**
>
> State Manager und Maintenance Windows können ähnliche Arten von Updates für Ihre verwalteten Knoten ausführen. Welche Option Sie wählen, hängt davon ab, ob Sie die System-Compliance automatisieren oder zeitkritische Aufgaben mit hoher Priorität während der von Ihnen angegebenen Zeiträume ausführen müssen.
> Weitere Informationen finden Sie unter Auswahl zwischen State Manager und Maintenance Windows.

Welche Vorteile bietet State Manager meiner Organisation?

Durch die Verwendung vorkonfigurierter Systems-Manager-Dokumente (SSM-Dokumente), bietet State Manager die folgenden Vorteile für die Verwaltung Ihrer Knoten:

- Bootstrap von Knoten mit bestimmter Software beim Startup.
- Herunterladen und Aktualisieren von Agenten nach einem definierten Zeitplan, einschließlich des SSM Agent.
- Konfigurieren von Netzwerkeinstellungen.
- Verbinden Sie Knoten mit einer Microsoft-Active-Directory-Domain.

- Ausführen von Skripts auf verwalteten Linux-, macOS- und Windows-verwalteten Knoten während des gesamten Lebenszyklus.

Um Konfigurationsabweichungen zwischen anderen AWS Ressourcen zu bewältigen, können Sie Automation, eine Funktion von Systems Manager, verwenden, mit der State Manager Sie die folgenden Arten von Aufgaben ausführen können:

- Fügen Sie eine Systems Manager Rolle an Amazon Elastic Compute Cloud (Amazon EC2)-Instances an, um sie auf verwaltete Knoten zu ändern.
- Erzwingen Sie die gewünschten Eingangs- und Ausgangsregeln für eine Sicherheitsgruppe.
- Erstellen oder löschen Sie Amazon DynamoDB-Backups.
- Erstellen oder löschen Sie Amazon Elastic Block Store (Amazon EBS)-Snapshots.
- Deaktivieren Sie Lese- und Schreibberechtigungen für Amazon Simple Storage Service (Amazon S3)-Buckets.
- Starten, Stoppen oder starten Sie verwaltete Knoten und Amazon Relational Database Service (Amazon RDS)-Instances neu.
- Anwenden von Patches auf Linux, macOS und Windows AMIs.

Weitere Informationen zur Verwendung von State Manager mit Automation-Runbooks finden Sie unter Planen von Automatisierungen mit State Manager-Zuordnungen.

An wen richtet sich State Manager?

State Managerist für jeden AWS Kunden geeignet, der die Verwaltung und Steuerung seiner AWS Ressourcen verbessern und Konfigurationsabweichungen reduzieren möchte.

Über welche Features verfügt State Manager?

Nachstehend sind einige der wichtigsten Features von State Manager aufgelistet:

- State Manager-Zuordnungen

  Eine State Manager Zuordnung ist eine Konfiguration, die Sie Ihren AWS Ressourcen zuweisen. Die Konfiguration definiert den Status, den Sie auf Ihren Ressourcen beibehalten möchten. Beispiel: Eine Zuordnung kann angeben, dass auf einem verwalteten Knoten Antiviren-Software installiert sein und ausgeführt werden muss oder dass bestimmte Ports geschlossen sein müssen.

Eine Assoziation gibt einen Zeitplan an, wann die Konfiguration und die Ziele für die Assoziation angewendet werden sollen. Beispielsweise könnte eine Zuordnung für Antivirensoftware einmal täglich auf allen verwalteten Knoten in einem AWS-Konto ausgeführt werden. Wenn die Software nicht auf einem Knoten installiert ist, könnte die Assoziation State Manager anweisen, sie zu installieren. Wenn die Software installiert ist, aber der Service nicht ausgeführt wird, könnte die Assoziation State Manager anweisen, den Service zu starten.

- Flexible Planungs-Optionen

  State Manager bietet die folgenden Optionen zum Planen, wenn eine Assoziation ausgeführt wird:

  - Sofortige oder verzögerte Verarbeitung

    Wenn Sie eine Assoziation erstellen, führt das System diese standardmäßig sofort auf den angegebenen Ressourcen aus. Nach der ersten Ausführung wird die Assoziation gemäß dem von Ihnen festgelegten Zeitplan in Intervallen ausgeführt.

    Sie können State Manager anweisen, eine Assoziation nicht sofort auszuführen, indem Sie die Apply association only at next specified Cron-interval (Übernehmen der Assoziation erst für das nächste angegebene Cron-Intervall)-Option in der Konsole oder im `ApplyOnlyAtCronInterval`-Parameter von der Befehlszeile aus verwenden.

  - Cron- und Rate-Ausdrücke

    Wenn Sie eine Zuordnung erstellen, geben Sie den Zeitpunkt an, zu dem State Manager die Konfiguration anwendet. State Manager unterstützt Standardausdrücke für Cron- und Rate-Ausdrücke für die Planung, wenn eine Zuordnung ausgeführt wird. State Manager unterstützt auch Cron-Ausdrücke, die einen Wochentag und das Zahlenzeichen (#) enthalten, um den x-ten Tag eines Monats festzulegen, um eine Zuordnung auszuführen, und das (L)-Zeichen, um den letzten X Tag des Monats anzugeben.

    > **Note**
    >
    > State Manager unterstützt derzeit nicht die Angabe von Monaten in Cron-Ausdrücken für Zuordnungen.

    Um weiter zu steuern, wann eine Assoziation ausgeführt wird, z. B. wenn Sie zwei Tage nach dem Patch-Dienstag eine Assoziation ausführen möchten, können Sie einen Offset angeben.

Ein Offset definiert, wie viele Tage nach dem geplanten Tag gewartet werden müssen, um eine Assoziation auszuführen.

Weitere Informationen zum Erstellen von Cron- und Rate-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

- Mehrere Targeting-Optionen

  Eine Assoziation gibt auch die Ziele für die Assoziation an. State Managerunterstützt die gezielte Ausrichtung von AWS Ressourcen mithilfe von Tags AWS Resource Groups, einzelnen Knoten-IDs oder allen verwalteten Knoten im aktuellen AWS-Region und AWS-Konto.

- Amazon-S3-Support

  Speichern Sie die Befehlsausgabe von Zuordnungsausführungen in einem Amazon-S3-Bucket Ihrer Wahl. Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

- EventBridge Unterstützung

  Diese Systems Manager Manager-Funktion wird in EventBridge Amazon-Regeln sowohl als Ereignistyp als auch als Zieltyp unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager.

Entstehen Kosten für die Verwendung von State Manager?

State Manager ist ohne Aufpreis erhältlich.

Erste Schritte mit State Manager

Führen Sie die folgenden Aufgaben aus, sich in ersten Schritten mit State Manager vertraut zu machen.

Aufgabe	Weitere Informationen
Systems Manager einrichten	Einrichten AWS Systems Manager
Weitere Informationen zu State Manager	Informationen zu State Manager
Erstellen und Zuweisen einer State Manager-Zuordnung zu Ihren Knoten	Arbeiten mit Zuordnungen in Systems Manager

Weitere Informationen

- Bekämpfung von Konfigurationsabweichungen mithilfe von Amazon EC2 Systems Manager und Windows DSC PowerShell
- Konfigurieren von Amazon EC2-Instances in einer Auto Scaling-Gruppe mit State Manager

Themen

- Informationen zu State Manager
- Arbeiten mit Zuordnungen in Systems Manager
- Walkthroughs zum AWS Systems ManagerState Manager

Informationen zu State Manager

State Manager, eine Funktion von AWS Systems Manager, ist ein sicherer und skalierbarer Service, der den Prozess automatisiert, bei dem verwaltete Knoten in einer Hybrid- und Multi-Cloud-Infrastruktur in einem von Ihnen definierten Zustand belassen werden.

So funktioniert State Manager:

1. Ermitteln Sie den Status, den Sie auf Ihre AWS Ressourcen anwenden möchten.

   Möchten Sie sicherstellen, dass Ihre verwalteten Knoten mit bestimmten Anwendungen wie Antiviren- oder Malware-Anwendungen konfiguriert sind? Möchten Sie die den Prozess zur Aktualisierung des SSM Agent oder andere AWS -Pakete wie z. B. `AWSPVDriver` automatisieren? Müssen Sie sicherstellen, dass bestimmte Ports geöffnet oder geschlossen sind? Bestimmen Sie zunächst den StatusState Manager, den Sie auf Ihre AWS Ressourcen anwenden möchten. Der Status, den Sie anwenden möchten, legt fest, welches SSM-Dokument Sie verwenden können, um eine State Manager-Zuordnung zu erstellen.

   Eine State Manager Zuordnung ist eine Konfiguration, die Sie Ihren AWS Ressourcen zuweisen. Die Konfiguration definiert den Status, den Sie auf Ihren Ressourcen beibehalten möchten. Beispiel: Eine Zuordnung kann angeben, dass auf einem verwalteten Knoten Antiviren-Software installiert sein und ausgeführt werden muss oder dass bestimmte Ports geschlossen sein müssen.

   Eine Assoziation gibt einen Zeitplan an, wann die Konfiguration und die Ziele für die Assoziation angewendet werden sollen. Beispielsweise könnte eine Zuordnung für Antivirensoftware einmal täglich auf allen verwalteten Knoten in einem AWS-Konto ausgeführt werden. Wenn die Software nicht auf einem Knoten installiert ist, könnte die Assoziation State Manager anweisen, sie zu

installieren. Wenn die Software installiert ist, aber der Service nicht ausgeführt wird, könnte die Assoziation State Manager anweisen, den Service zu starten.

2. Stellen Sie fest, ob ein vorkonfiguriertes SSM-Dokument Ihnen helfen kann, den gewünschten Status für Ihre AWS Ressourcen zu erreichen.

   Systems Manager umfasst Dutzende von vorkonfigurierten SSM-Dokumenten, die Sie verwenden können, um eine Zuordnung zu erstellen. Vorkonfigurierte Dokumente sind bereit, allgemeine Aufgaben wie das Installieren von Anwendungen, das Konfigurieren von Amazon, das Ausführen von AWS Systems Manager Automatisierungen CloudWatch, das Ausführen von Shell-Skripts PowerShell und das Verbinden verwalteter Knoten mit einer Verzeichnisdienstdomäne für Active Directory auszuführen.

   Sie können alle SSM-Dokumente in der Systems Manager-Konsole anzeigen. Wählen Sie den Namen eines Dokuments an, um mehr darüber zu erfahren. Nachfolgend finden Sie zwei Beispiele: `AWS-ConfigureAWSPackage` und `AWS-InstallApplication`.

3. Erstellen einer Assoziation.

   Sie können eine Zuordnung mithilfe der Systems Manager Manager-Konsole, der AWS Command Line Interface (AWS CLI), AWS Tools for Windows PowerShell (Tools für Windows PowerShell) oder der Systems Manager Manager-API erstellen. Beim Erstellen einer Assoziation geben Sie die folgenden Informationen an:

   - Ein Name für die Assoziation.
   - Die Parameter für das SSM-Dokument (z. B. den Pfad zu der zu installierenden Anwendung oder zu dem Skript, das auf den Knoten ausgeführt werden soll).
   - Ziele für die Zuordnung. Sie können verwaltete Knoten als Ziel auswählen, indem Sie Tags angeben, einzelne Instance-IDs oder eine Gruppe in AWS Resource Groups wählen. Sie können auch alle verwalteten Knoten im aktuellen AWS-Region und als Ziel verwenden AWS-Konto.
   - Einen Zeitplan für die Häufigkeit der Statusanwendung. Sie können einen Cron- oder Rate-Ausdruck festlegen. Weitere Informationen zum Erstellen von Zeitplänen mit cron- und Rate-Ausdrücken für Zuordnungen finden Sie unter Cron- und Rate-Ausdrücke für Zuordnungen.

     > **Note**
     >
     > State Manager unterstützt derzeit nicht die Angabe von Monaten in Cron-Ausdrücken für Assoziationen.

Wenn Sie den Befehl ausführen, um die Assoziation zu erstellen, bindet Systems Manager die von Ihnen angegebenen Informationen (Zeitplan, Ziele, SSM-Dokument und Parameter) an die anvisierten Ressourcen. Der Status der Zuordnung wird zunächst als „Pending (ausstehend)“ angezeigt, während das System versucht, alle Ziele zu erreichen und sofort den in der Zuordnung angegebenen Status anzuwenden.

> **Note**
>
> Wenn Sie eine neue Zuordnung erstellen, die ausgeführt werden soll, solange eine frühere Zuordnung noch ausgeführt wird, führt dies zu einer Unterbrechung der früheren Zuordnung und die neue Zuordnung wird ausgeführt.

Systems Manager meldet den Status der Anforderung zum Erstellen von Assoziationen auf den Ressourcen. Sie können Statusdetails in der Konsole oder (für verwaltete Knoten) mithilfe der DescribeInstanceAssociationsStatusAPI-Operation anzeigen. Wenn Sie die Ausgabe des Befehls beim Erstellen einer Zuordnung in Amazon Simple Storage Service (Amazon S3) auswählen, können Sie die Ausgabe auch im angegebenen Amazon S3-Bucket anzeigen.

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

> **Note**
>
> API-Vorgänge, die während der Ausführung einer Zuordnung durch das SSM-Dokument initiiert werden, werden in AWS CloudTrail nicht protokolliert.

4. Überwachen und aktualisieren Sie.

Nach dem Erstellen der Zuordnung wendet State Manager den Status entsprechend des in der Zuordnung definierten Zeitplans erneut an. Sie können den Status Ihrer Zuordnungen auf der State Manager-Seite in der Konsole oder mit dem direkten Aufruf der Zuordnungs-ID anzeigen, die von Systems Manager beim Erstellen der Zuordnung generiert wurde. Weitere Informationen finden Sie unter Anzeigen von Zuordnungsverläufen. Sie können Ihre Zuordnungsdokumente aktualisieren und Sie bei Bedarf erneut anwenden. Sie können auch mehrere Versionen einer Zuordnung erstellen. Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung.

Wann werden Zuordnungen auf Ressourcen angewendet?

Wenn Sie eine Zuordnung erstellen, geben Sie ein SSM-Dokument an, das die Konfiguration, eine Liste der Zielressourcen und einen Zeitplan für die Anwendung der Konfiguration definiert. Standardmäßig führt State Manager die Zuordnung aus, wenn Sie sie erstellen und dann nach Ihrem Zeitplan. State Manager versucht auch, die Zuordnung in den folgenden Situationen auszuführen:

- Zuordnung bearbeiten – State Manager führt die Zuordnung aus, nachdem ein Benutzer seine Änderungen bearbeitet und in einem der folgenden Zuordnungsfelder gespeichert hat: `DOCUMENT_VERSION`, `PARAMETERS`, `SCHEDULE_EXPRESSION`, `OUTPUT_S3_LOCATION`.
- Dokumentbearbeitung – State Manager führt die Zuordnung aus, nachdem ein Benutzer Änderungen am SSM-Dokument, das den Konfigurationsstatus der Zuordnung definiert, bearbeitet und gespeichert hat. Insbesondere wird die Zuordnung nach den folgenden Änderungen am Dokument ausgeführt:
  - Ein Benutzer gibt eine neue $DEFAULT-Dokumentversion an, und die Zuordnung wurde mit der $DEFAULT Version erstellt.
  - Ein Benutzer aktualisiert ein Dokument und die Zuordnung wurde mit der $LATEST-Version erstellt.
  - Ein Benutzer löscht das Dokument, das beim Erstellen der Zuordnung angegeben wurde.
- Änderung des Parameter Store-Parameterwerts – State Manager führt die Zuordnung aus, nachdem ein Benutzer den Wert eines in der Zuordnung definierten Parameters bearbeitet hat.
- Manueller Start – State Manager führt die Zuordnung aus, wenn sie vom Benutzer entweder über die Systems-Manager-Konsole oder programmgesteuert initiiert wird.
- Zieländerungen — State Manager führt die Zuordnung aus, nachdem eine der folgenden Aktivitäten auf einem Zielknoten stattgefunden hat:
  - Ein verwalteter Knoten ist zum ersten Mal online.
  - Ein verwalteter Knoten wird online geschaltet, nachdem ein geplantes Zuordnungsprogramm verpasst wurde.
  - Ein verwalteter Knoten wird online geschaltet, nachdem er länger als 30 Tage angehalten wurde.

> **Note**
>
> Ziel-Updates wirken sich nicht auf Zuordnungen aus, die mit Systems Manager Automation erstellt wurden.

Arbeiten mit Zuordnungen in Systems Manager

In diesem Abschnitt wird beschrieben, wie Sie State Manager-Assoziation mit der AWS Systems Manager-Konsole, der AWS Command Line Interface (AWS CLI) und AWS Tools for PowerShell erstellen und verwalten.

Themen

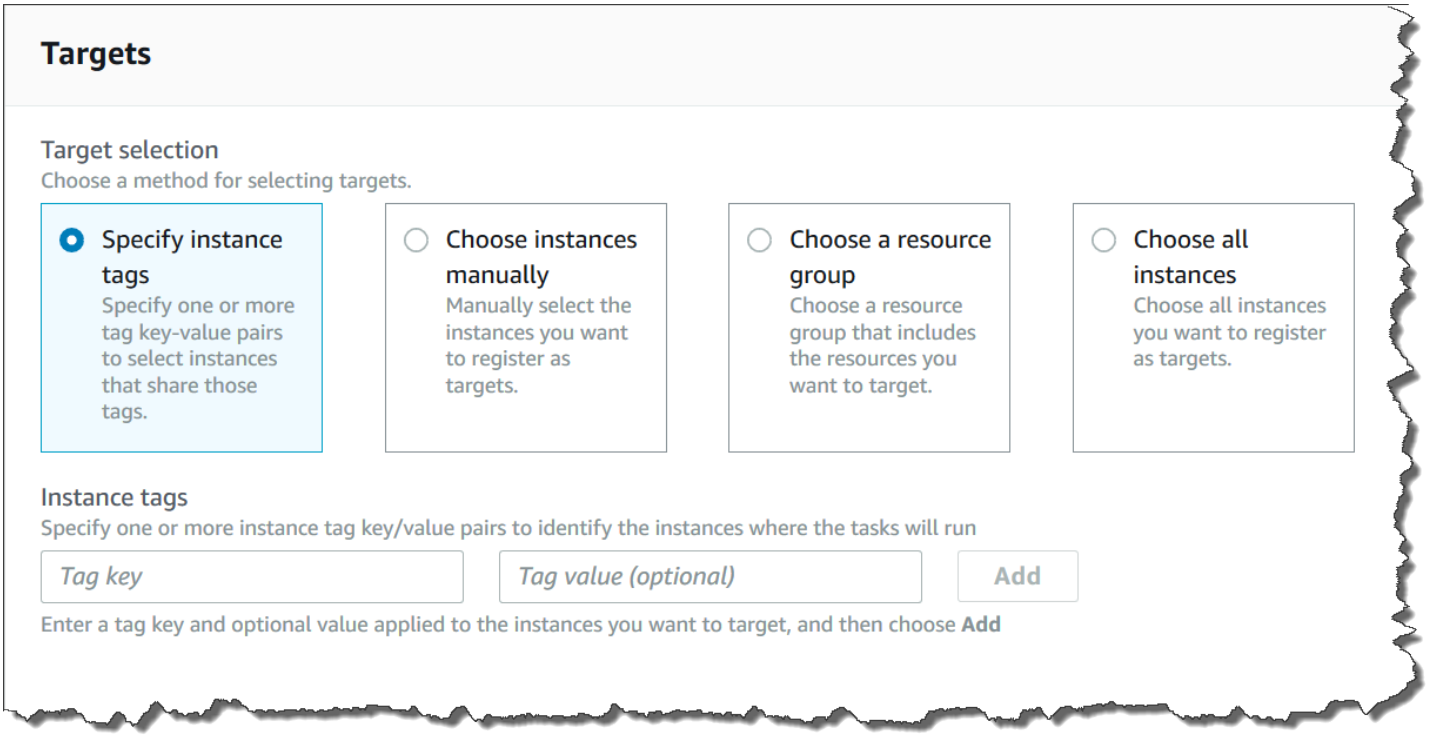
- Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen
- Erstellen von Zuordnungen
- Bearbeiten und Erstellen einer neuen Version einer Zuordnung
- Löschen von Zuordnungen
- Ausführen von Auto-Scaling-Gruppen mit Zuordnungen
- Anzeigen von Zuordnungsverläufen
- Arbeiten mit Zuordnungen mithilfe von IAM

Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen

In diesem Thema werden Features von State Manager, eine Funktion von AWS Systems Manager, beschrieben, mit denen Sie eine Assoziation für Dutzende oder Hunderte von Knoten bereitstellen und gleichzeitig steuern können, wie viele Knoten die Assoziation zum geplanten Zeitpunkt ausführen.

Targets (Ziele)

Wenn Sie eine State Manager-Assoziation erstellen, wählen Sie im Abschnitt Targets (Ziele) der Systems Manager-Konsole, welche Knoten mit der Assoziation konfiguriert werden sollen, wie hier gezeigt.



Wenn Sie mithilfe eines Befehlszeilen-Tools wie AWS Command Line Interface (AWS CLI) eine Zuordnung erstellen, geben Sie den Parameter `targets` an. Wenn Sie Knoten als Ziel auswählen, können Sie Dutzende, Hunderte oder Tausende von Knoten mit einer Assoziation konfigurieren, ohne einzelne Knoten-IDs angeben oder auswählen zu müssen.

Jeder verwaltete Knoten kann von maximal 20 Zuordnungen betroffen sein.

State Manager enthält die folgenden Zieloptionen bei der Erstellung einer Zuordnung.

Tags angeben

Verwenden Sie diese Option, um einen Tag-Schlüssel und (optional) einen Tag-Wert anzugeben, die Ihren Knoten zugewiesen sind. Wenn Sie die Anforderung ausführen, versucht das System, die Assoziation auf allen Knoten zu erstellen, die dem angegebenen Tag-Schlüssel und -Wert entsprechen. Wenn Sie mehrere Tag-Werte angegeben haben, zielt die Assoziation auf jeden Knoten mit mindestens einem dieser Tag-Werte ab. Wenn das System die Zuordnung erstmals erstellt, führt es die Zuordnung aus. Nach dieser erstmaligen Ausführung führt das System die Zuordnung dem angegebenen Zeitplan entsprechend aus.

Wenn Sie neue Knoten erstellen und diesen Konten den angegebenen Tag-Schlüssel und -Wert zuweisen, wendet das System die Assoziation automatisch an und führt sie sofort und anschließend dem angegebenen Zeitplan entsprechend aus. Dies gilt, wenn die Zuordnung ein Befehls- oder

Richtliniendokument verwendet und nicht angewendet wird, wenn die Zuordnung ein Automation-Runbook verwendet. Wenn Sie die angegebenen Tags aus einem Knoten löschen, führt das System die Assoziation für diese Knoten nicht mehr aus.

> **Note**
>
> Wenn Sie Automation-Runbooks mit State Manager und die Tagging-Einschränkung verhindert, dass Sie ein bestimmtes Ziel erreichen, sollten Sie die Verwendung von Automation-Runbooks mit Amazon EventBridge in Betracht ziehen. Weitere Informationen finden Sie unter Ausführen von Automatisierungen basierend auf Ereignissen. Weitere Informationen zur Verwendung von State Manager mithilfe von Runbooks finden Sie unter Planen von Automatisierungen mit State Manager-Zuordnungen.

Als bewährte Methode empfehlen wir die Verwendung von Tags, wenn Sie Zuordnungen erstellen, die ein Befehls- oder Richtlinien-Dokument verwenden. Wir empfehlen auch die Verwendung von Tags, wenn Sie Zuordnungen zu Auto-Scaling-Gruppen erstellen. Weitere Informationen finden Sie unter Ausführen von Auto-Scaling-Gruppen mit Zuordnungen.

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - Wenn Sie eine Zuordnung in der Konsole erstellen und Knoten mit Hilfe von Tags anvisieren, können Sie nur einen Tag-Schlüssel angeben. Wenn Sie die Konsole verwenden und Ihre Knoten mit mehr als einem Tag-Schlüssel anvisieren möchten, weisen Sie die Tag-Schlüssel einer AWS Resource Groups-Gruppe zu und fügen Sie die Knoten zu dieser Gruppe hinzu. Sie können dann die Option Ressourcengruppe in der Zielliste auswählen, wenn Sie die State Manager-Zuordnung erstellen.
> - Sie können mit der AWS CLI maximal fünf Tag-Schlüssel angeben. Wenn Sie die AWS CLI verwenden, müssen alle im `create-association`-Befehl angegebenen Tag-Schlüssel dem Knoten aktuell zugewiesen sein. Sind sie das nicht, kann State Manager den Knoten nicht für eine Zuordnung anvisieren. Weitere Informationen zum Zuweisen von Tags zu Knoten finden Sie unter Markieren von Systems Manager-Ressourcen.

Manuelles Auswählen von Knoten

Verwenden Sie diese Option, um die Knoten, auf denen Sie die Assoziation erstellen möchten, manuell auszuwählen. Im Bereich Instances werden alle von Systems Manager verwalteten Knoten im aktuellen AWS-Konto und in der aktuellen AWS-Region angezeigt. Sie können beliebig viele Knoten manuell auswählen. Wenn das System die Zuordnung erstmals erstellt, führt es die Zuordnung aus. Nach dieser erstmaligen Ausführung führt das System die Zuordnung dem angegebenen Zeitplan entsprechend aus.

> **Note**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

Eine Ressourcengruppe auswählen

Verwenden Sie diese Option, um eine Assoziation für alle Knoten zu erstellen, die von einer Tag-basierten AWS Resource Groups-Abfrage oder einer Stack-basierten AWS CloudFormation-Abfrage zurückgegeben werden.

Unten folgen Details zum Auswählen von Ressourcengruppen als Ziel für eine Zuordnung.

- Wenn Sie einer Gruppe neue Knoten hinzufügen, ordnet das System die Knoten automatisch der Assoziation zu, die die Ressourcengruppe zum Ziel hat. Wenn das System die Änderung erkennt, wendet es die Assoziation auf die Knoten an. Nach dieser erstmaligen Ausführung führt das System die Zuordnung dem angegebenen Zeitplan entsprechend aus.
- Wenn Sie eine Zuordnung erstellen, die auf eine Ressourcengruppe abzielt und der `AWS::SSM::ManagedInstance`-Ressourcentyp für diese Gruppe angegeben wurde, läuft die Zuordnung in einer Hybrid- und Multi-Cloud-Umgebung standardmäßig sowohl auf Instances der Amazon Elastic Compute Cloud (Amazon EC2) als auch auf Nicht-EC2-Knoten.
- Wenn Sie eine Zuordnung erstellen, die auf eine Ressourcengruppe abzielt, dürfen der Ressourcengruppe nicht mehr als fünf Tag-Schlüssel zugewiesen oder mehr als fünf Werte für einen Tag-Schlüssel angegeben werden. Wenn eine dieser Bedingungen auf die Tags und Schlüssel zutrifft, die Ihrer Ressourcengruppe zugewiesen sind, kann die Zuordnung nicht ausgeführt werden und gibt einen `InvalidTarget`-Fehler zurück.
- Wenn Sie eine Ressourcengruppe löschen, führen alle Instances in dieser Gruppe die Zuordnung nicht mehr aus. Es ist ratsam, Zuordnungen zu löschen, die die Gruppe zum Ziel haben.

- Sie können für eine Zuordnung maximal eine einzelne Ressourcengruppe als Ziel auswählen. Mehrere oder verschachtelte Gruppen werden nicht unterstützt.
- Nachdem Sie eine Zuordnung erstellt haben, aktualisiert State Manager die Zuordnung in regelmäßigen Abständen mit Informationen zu Ressourcen in der Ressourcengruppe. Wenn Sie einer Ressourcengruppe neue Ressourcen hinzufügen, hängt es von verschiedenen Faktoren ab, wann das System die Zuordnung auf die neuen Ressourcen anwendet. Sie können den Status der Zuordnung auf der Seite State Manager der Systems Manager-Konsole bestimmen.

> **Warning**
>
> Ein AWS Identity and Access Management (IAM)-Benutzer, eine -Gruppe oder eine -Rolle mit der Berechtigung zum Erstellen einer Zuordnung, die eine Ressourcengruppe von Amazon-EC2-Instances zum Ziel hat, hat automatisch die Kontrolle auf Stammebene für alle Instances in der Gruppe. Sie sollten nur vertrauenswürdigen Administratoren die Berechtigung erteilen, Zuordnungen zu erstellen.

Weitere Informationen zu Resource Groups finden Sie unter Was ist AWS Resource Groups? im AWS Resource Groups-Benutzerhandbuch.

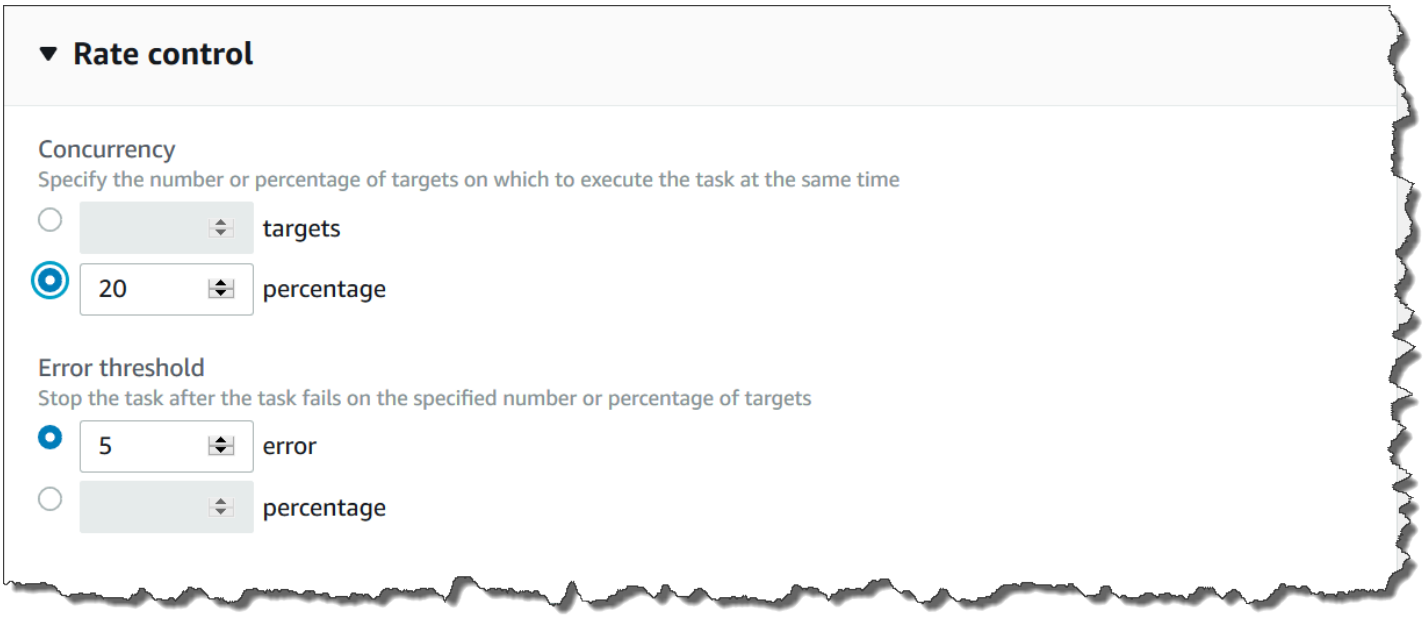
Wählen Sie alle Knoten

Verwenden Sie diese Option, um alle Knoten im aktuellen AWS-Konto und in der aktuellen AWS-Region als Ziel auszuwählen. Wenn Sie die Anforderung ausführen, versucht das System, die Assoziation auf allen Knoten im aktuellen AWS-Konto und in der aktuellen AWS-Region zu erstellen. Wenn das System die Zuordnung erstmals erstellt, führt es die Zuordnung aus. Nach dieser erstmaligen Ausführung führt das System die Zuordnung dem angegebenen Zeitplan entsprechend aus. Wenn Sie neue Instances erstellen, wendet das System die Assoziation automatisch an und führt sie sofort und anschließend dem angegebenen Zeitplan entsprechend aus.

Ratensteuerungen

Sie können die Ausführung einer Assoziation für Ihre Knoten steuern, indem Sie einen Gleichzeitigkeitswert und einen Fehlerschwellenwert angeben. Der Gleichzeitigkeitswert gibt an, wie viele Knoten die Assoziation gleichzeitig ausführen können. Der Fehlerschwellenwert gibt an, wie viele Assoziationsausführungen fehlschlagen dürfen, bevor Systems Manager einen Befehl an alle mit dieser Assoziation konfigurierten Knoten sendet, um die Ausführung der Assoziation zu beenden. Der Befehl verhindert, dass die Zuordnung vor der nächsten geplanten Zuordnung ausgeführt wird.

Die Gleichzeitigkeits- und die Fehlergrenzwertfeature werden gemeinsam als Ratensteuerungen bezeichnet.



Nebenläufigkeit

Durch Angabe eines Gleichzeitigkeitswerts können die Auswirkungen der Ausführung auf Ihre Knoten begrenzt werden, indem Sie angeben, dass jeweils nur eine bestimmte Anzahl von Knoten eine Assoziation gleichzeitig verarbeiten kann. Sie können entweder eine absolute Anzahl an verwalteten Knoten, z. B. 20, oder einen Prozentsatz des Ziel-Knotensatzes, beispielsweise 10 %, angeben.

Bitte beachten Sie die folgenden Einschränkungen und Begrenzungen für die Gleichzeitigkeitsfunktion bei State Manager:

- Wenn Sie eine Assoziation erstellen, indem Sie Ziele angeben, aber keinen Gleichzeitigkeitswert festlegen, dann gibt State Manager automatisch eine maximale Gleichzeitigkeit von 50 Knoten vor.
- Wenn eine Assoziation ausgeführt wird, die die Gleichzeitigkeitsfunktion verwendet, und ein neuer Knoten online geht, der den Zielkriterien entspricht, führen diese neuen Knoten die Assoziation aus, wenn damit der Gleichzeitigkeitswert nicht überschritten wird. Wenn der Gleichzeitigkeitswert überschritten wird, wird der Knoten für das aktuelle Assoziationsausführungsintervall ignoriert. Die Knoten werden dann zum nächsten geplanten Intervall bei normaler Beachtung der Gleichzeitigkeitsbeschränkung ausgeführt.
- Wenn Sie eine Assoziation aktualisieren, die die Gleichzeitigkeitsfunktion verwendet, und diese Assoziation wird gerade auf einer oder mehreren Knoten ausgeführt, dann erhalten diese Knoten die Erlaubnis, die Ausführung der Assoziation abzuschließen. Zuordnungen, deren

Ausführung noch nicht begonnen hat, werden nicht mehr ausgeführt. Nachdem die Ausführung laufender Assoziationen abgeschlossen wurde, führen alle Ziel-Knoten die Assoziation sofort erneut aus, da sie aktualisiert wurde. Auch bei dieser erneuten Ausführung gilt der festgelegte Gleichzeitigkeitswert.

Fehlerschwellenwerte

Ein Fehlergrenzwert gibt an, wie viele Assoziationsausführungen auftreten dürfen, bevor Systems Manager einen Befehl zu jedem mit dieser Assoziation konfigurierten Knoten sendet. Der Befehl verhindert, dass die Zuordnung vor der nächsten geplanten Zuordnung ausgeführt wird. Sie können entweder eine absolute Anzahl an Fehlern, z. B. 10, oder einen Prozentsatz des festgelegten Ziels, beispielsweise 10 % festlegen.

Wenn Sie z. B. die absolute Zahl von drei Fehlern angeben, sendet State Manager einen Befehl zum Anhalten, wenn der vierte Fehler zurückgegeben wird. Wenn Sie 0 angeben, sendet State Manager den Befehl zum Anhalten, wenn der erste Fehler gemeldet wird.

Wenn Sie einen Fehlerschwellenwert von 10 % für 50 Zuordnungen angeben, sendet State Manager den Befehl zum Anhalten, wenn der sechste Fehler zurückgegeben wird. Zuordnungen, die bereits ausgeführt werden, wenn ein Fehlerschwellenwert erreicht wird, werden noch abgeschlossen, es besteht jedoch die Möglichkeit, dass einige dieser Zuordnungen fehlschlagen. Um sicherzustellen, dass nicht mehr Fehler als im Fehlerschwellenwert angegeben auftreten, setzen Sie den Wert für die Concurrency (Gleichzeitigkeit) auf 1, sodass die Zuordnungen jeweils einzeln ausgeführt werden.

Es gelten die folgenden Einschränkungen und Begrenzungen für Fehlerschwellenwert in State Manager:

- Fehlerschwellenwerte werden für das aktuelle Intervall übernommen.
- Informationen zu den einzelnen Fehlern werden mit detaillierten Informationen zu den Arbeitsschritten im Zuordnungsverlauf aufgezeichnet.
- Wenn Sie eine Zuordnung unter Verwendung von Zielen erstellen, aber keinen Fehlerschwellenwert angeben, dann gibt State Manager automatisch einen Schwellenwert von 100 % Fehlern vor.

Erstellen von Zuordnungen

State Manager, eine Funktion von AWS Systems Manager, hilft Ihnen dabei, Ihre AWS Ressourcen in einem von Ihnen definierten Zustand zu halten und Konfigurationsabweichungen zu reduzieren.

Um dies zu tun, verwendet State Manager Assoziationen. Eine Zuordnung ist eine Konfiguration, die Sie Ihren AWS Ressourcen zuweisen. Die Konfiguration definiert den Status, den Sie auf Ihren Ressourcen beibehalten möchten. Beispiel: Eine Zuordnung kann angeben, dass auf einem verwalteten Knoten Antiviren-Software installiert sein und ausgeführt werden muss oder dass bestimmte Ports geschlossen sein müssen.

Eine Assoziation gibt einen Zeitplan an, wann die Konfiguration und die Ziele für die Assoziation angewendet werden sollen. Beispielsweise könnte eine Zuordnung für Antivirensoftware einmal täglich auf allen verwalteten Knoten in einem AWS-Konto ausgeführt werden. Wenn die Software nicht auf einem Knoten installiert ist, könnte die Assoziation State Manager anweisen, sie zu installieren. Wenn die Software installiert ist, aber der Service nicht ausgeführt wird, könnte die Assoziation State Manager anweisen, den Service zu starten.

> **Note**
>
> Sie können einer Assoziation bei der Erstellung Tags zuweisen, indem Sie ein Befehlszeilentool wie AWS CLI oder verwenden AWS Tools for PowerShell. Das Hinzufügen von Tags zu einer Zuordnung über die Systems-Manager-Konsole wird nicht unterstützt. Weitere Informationen zu Tags erhalten Sie unter Markieren von Systems Manager-Ressourcen.

In den folgenden Verfahren wird beschrieben, wie eine Assoziation erstellt wird, die ein `Command`- oder ein `Policy`-Dokument verwendet, um verwaltete Knoten anzuvisieren. Informationen zum Erstellen einer Zuordnung, die mithilfe eines Automatisierungs-Runbooks auf Knoten oder andere AWS Ressourcentypen abzielt, finden Sie unterPlanen von Automatisierungen mit State Manager-Zuordnungen.

Zuordnungsziele und Ratensteuerungen

Eine Zuordnung gibt an, welche verwalteten Knoten oder Ziele die Zuordnung erhalten sollen. State Manager enthält mehrere Funktionen, mit denen Sie Ihre verwalteten Knoten gezielt ausrichten und steuern können, wie die Zuordnung für diese Ziele bereitgestellt wird. Weitere Informationen zu Zielen und Ratensteuerungen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.

Ausgeführte Zuordnungen

Standardmäßig wird von State Manager eine Zuordnung sofort nach deren Erstellung und dann gemäß dem von Ihnen definierten Zeitplan ausgeführt.

Das System führt auch Zuordnungen nach den folgenden Regeln aus:

- State Manager versucht, die Assoziation während eines Intervalls auf allen angegebenen oder als Ziel ausgewählten Knoten auszuführen.
- Wenn eine Assoziation in einem Intervall nicht ausgeführt wird (weil beispielsweise die Anzahl der Knoten, die die Zuordnung gleichzeitig ausführen können, durch einen Gleichzeitigkeitswert begrenzt wird), versucht State Manager, die Assoziation im nächsten Intervall auszuführen.
- State Manager führt die Zuordnung nach Änderungen an der Konfiguration, den Zielknoten, Dokumenten oder Parametern der Zuordnung aus. Weitere Informationen finden Sie unter Wann werden Zuordnungen auf Ressourcen angewendet?.
- State Manager zeichnet einen Verlauf für alle übersprungenen Datensätze an. Sie können den Verlauf auf der Registerkarte Execution History (Ausführungsverlauf) anzeigen.

Planen von Zuordnungen

Sie können Zuordnungen so planen, dass sie in einfachen Intervallen ausgeführt werden, z. B. alle 10 Stunden, oder Sie können erweiterte Zeitpläne erstellen, indem Sie benutzerdefinierte Cron- und Rate-Ausdrücke verwenden. Sie können auch verhindern, dass Zuordnungen ausgeführt werden, wenn Sie diese zum ersten Mal erstellen.

Verwenden von Cron- und Rate-Ausdrücken zur Planung von Ausführungen von Zuordnungen

Zusätzlich zu den standardmäßigen Cron- und Rate-Ausdrücken werden von State Manager auch Cron-Ausdrücke unterstützt, die einen Wochentag und das Nummernzeichen (#) enthalten, um den x-ten Tag eines Monats für die Ausführung einer Assoziation zu bestimmen. Hier ist ein Beispiel, das am dritten Dienstag jeden Monats um 23:30 Uhr UTC einen Cron-Zeitplan ausführt:

```
cron(30 23 ? * TUE#3 *)
```

Hier ist ein Beispiel, das am zweiten Donnerstag jeden Monats um Mitternacht UTC läuft:

```
cron(0 0 ? * THU#2 *)
```

State Manager unterstützt auch das (L)-Zeichen, um den letzten X Tag des Monats anzugeben. Hier ist ein Beispiel, das am letzten Dienstag jeden Monats um Mitternacht UTC einen Cron-Zeitplan ausführt:

```
cron(0 0 ? * 3L *)
```

Um weiter zu steuern, wann eine Assoziation ausgeführt wird, z. B. wenn Sie zwei Tage nach dem Patch-Dienstag eine Assoziation ausführen möchten, können Sie einen Offset angeben. Ein Offset definiert, wie viele Tage nach dem geplanten Tag gewartet werden müssen, um eine Assoziation auszuführen. Wenn Sie beispielsweise einen Cron-Zeitplan mit `cron(0 0 ? * THU#2 *)` angegeben haben, können Sie die Nummer 3 im Schedule offset (Planversatz)-Feld angeben, um die Assoziation jeden Sonntag nach dem zweiten Donnerstag im Monat auszuführen.

> **Note**
>
> Um Offsets zu verwenden, müssen Sie entweder Zuordnung nur beim nächsten angegebenen Cron-Intervall in der Konsole anwenden auswählen oder den `ApplyOnlyAtCronInterval`-Parameter über die Befehlszeile angeben. Wenn eine dieser Optionen aktiviert ist, führt State Manager die Zuordnung nicht sofort nach dem Erstellen aus.

Weitere Informationen zu cron- und Rate-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Erstellen einer Zuordnung (Konsole)

Im folgenden Verfahren wird beschrieben, wie mithilfe der Systems Manager-Konsole eine State Manager-Zuordnung erstellt wird.

> **Warning**
>
> Wenn Sie eine Zuordnung erstellen, können Sie eine AWS Ressourcengruppe verwalteter Knoten als Ziel für die Zuordnung auswählen. Wenn ein AWS Identity and Access Management (IAM-) Benutzer, eine Gruppe oder eine Rolle berechtigt ist, eine Zuordnung zu erstellen, die auf eine Ressourcengruppe verwalteter Knoten abzielt, hat dieser Benutzer, diese Gruppe oder Rolle automatisch die Kontrolle über alle Knoten in der Gruppe auf Stammebene. Sie sollten nur vertrauenswürdigen Administratoren die Berechtigung erteilen, Assoziationen zu erstellen.

So erstellen Sie eine State Manager-Zuordnung

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie Create association (Zuordnung erstellen) aus.
4. Geben Sie im Feld Name einen Namen an.
5. Wählen Sie in der Liste Document (Dokument) die Option neben dem Namen des Dokuments aus. Beachten Sie den Dokumenttyp. Dieses Verfahren gilt für `Command`- und `Policy`-Dokumente. Weitere Informationen zum Erstellen einer Zuordnung, die ein Automation-Runbook verwendet, finden Sie unter Planen von Automatisierungen mit State Manager-Zuordnungen.

   > **Important**
   >
   > State Manager unterstützt nicht das Ausführen von Zuordnungen, die eine neue Version eines Dokuments verwenden, wenn dieses Dokument von einem anderen Konto freigegeben wird. State Manager führt immer die `default`-Version eines Dokuments aus, wenn es von einem anderen Konto freigegeben wird, obwohl die Systems-Manager-Konsole anzeigt, dass eine neue Version verarbeitet wurde. Wenn Sie eine Zuordnung mit einer neuen Version eines Dokuments ausführen möchten, das von einem anderen Konto freigegeben wurde, müssen Sie die Dokumentversion auf `default` einstellen.

6. Geben Sie für Parameters (Parameter) die erforderlichen Eingabeparameter an.
7. (Optional) Wählen Sie einen CloudWatch Alarm aus, den Sie bei Ihrem Verband zur Überwachung beantragen möchten.

   > **Note**
   >
   > Bitte beachten Sie die folgenden Informationen über diesen Schritt.
   >
   > - Die Liste der Alarme zeigt maximal 100 Alarme. Wenn Sie Ihren Alarm nicht in der Liste sehen, verwenden Sie den, AWS Command Line Interface um die Zuordnung zu erstellen. Weitere Informationen finden Sie unter Erstellen einer Zuordnung (Befehlszeile).
   > - Um Ihrem Befehl einen CloudWatch Alarm anzuhängen, muss der IAM-Principal, der die Zuordnung erstellt, über die entsprechende Berechtigung für die

`iam:createServiceLinkedRole` Aktion verfügen. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden.

- Ausstehende Befehlsaufrufe oder Automatisierungen werden nicht ausgeführt, wenn Ihr Alarm aktiviert wird.

8. Wählen Sie für Targets (Ziele) eine Option aus. Weitere Informationen zur Verwendung von Zielen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.
9. Wählen Sie im Abschnitt Specify schedule (Zeitplan angeben) entweder On Schedule (Nach Zeitplan) oder No schedule (Kein Zeitplan) aus. Wenn Sie On schedule (Auf Zeitplan) auswählen, verwenden Sie die verfügbaren Schaltflächen zum Erstellen eines cron- oder rate-Zeitplans für die Zuordnung.

   Wenn Sie nicht möchten, dass eine Zuordnung unmittelbar nach der Erstellung ausgeführt wird, wählen Sie Apply association only at the next specified Cron interval (Zuordnung erst beim nächsten angegebenen Cron-Intervall anwenden).
10. (Optional) Im Schedule offset (Planversatz), geben Sie eine Zahl zwischen 1 und 6 an.
11. Im Abschnitt Advanced options (Erweiterte Optionen) wählen Sie mit Compliance severity (Compliance-Schweregrad) einen Schweregrad für die Zuordnung und mit Change Calendars (Änderungskalender) einen Änderungskalender für die Zuordnung aus.

    In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem Schweregrad, den Sie hier angeben. Weitere Informationen finden Sie unter Informationen zu State Manager-Zuordnungs-Compliance.

    Der Änderungskalender bestimmt, wann die Zuordnung ausgeführt wird. Wenn der Kalender geschlossen ist, wird die Zuordnung nicht angewendet. Wenn der Kalender geöffnet ist, wird die Zuordnung entsprechend ausgeführt. Weitere Informationen finden Sie unter AWS Systems Manager Change Calendar.
12. Wählen Sie im Abschnitt Rate control (Ratensteuerung) Optionen für die Ausführung der Assoziation auf mehreren Knoten aus. Weitere Informationen zu Ratensteuerungen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.

    Wählen Sie im Abschnitt Concurrency (Gleichzeitigkeit) eine Option aus:

    - Wählen Sie Targets (Ziele) aus, um eine absolute Anzahl von Zielen einzugeben, die die Zuordnung gleichzeitig ausführen können.

- Wählen Sie Percentage (Prozentsatz) aus, um einen Prozentsatz der Ziele anzugeben, die die Zuordnung gleichzeitig ausführen können.

Wählen Sie im Abschnitt Error threshold (Fehlerschwellenwert) eine Option aus:

- Wählen Sie Errors (Fehler) aus und geben Sie die absolute Anzahl erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
- Wählen Sie Percentage (Prozentsatz) aus und geben Sie den Prozentsatz erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.

13. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Dienstrolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

Im Folgenden finden Sie die minimalen Berechtigungen, die erforderlich sind, um Amazon S3-Ausgabe für eine Zuordnung zu aktivieren. Sie können den Zugriff weiter einschränken, indem Sie IAM-Richtlinien an Benutzer oder Rollen innerhalb eines Kontos anfügen. Ein Amazon EC2-Instance-Profil sollte mindestens eine IAM-Rolle mit der von `AmazonSSMManagedInstanceCore` verwalteten Richtlinie und der folgenden Inline-Richtlinie haben.

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

```
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
    ]
}
```

Für minimale Berechtigungen muss der Amazon S3-Bucket, in den Sie exportieren, über die von der Amazon S3-Konsole definierten Standardeinstellungen verfügen. Weitere Informationen zum Erstellen eines Amazon S3-Buckets finden Sie unter Erstellen eines Buckets im Amazon S3-Benutzerhandbuch.

> **Note**
>
> API-Vorgänge, die während der Ausführung einer Zuordnung durch das SSM-Dokument initiiert werden, werden in AWS CloudTrail nicht protokolliert.

14. Wählen Sie Create Association.

> **Note**
>
> Wenn Sie die von Ihnen erstellte Zuordnung löschen, wird die Zuordnung nicht mehr auf Zielen dieser Zuordnung ausgeführt.

Erstellen einer Zuordnung (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die AWS CLI (unter Linux oder Windows) oder Tools für verwenden PowerShell , um eine State Manager Zuordnung zu erstellen. Dieser Abschnitt enthält einige Beispiele, die zeigen, wie Ziele und Ratensteuerungen verwendet werden. Mit Zielen und Ratensteuerungen können Sie Dutzenden oder Hunderten von Knoten eine Assoziation zuweisen, während Sie die Ausführung dieser Assoziationen steuern. Weitere Informationen zu Zielen und Ratensteuerungen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.

Bevor Sie beginnen

Der Parameter `targets` ist ein Array von Suchkriterien, die Knoten mit einer Kombination aus `Key` und `Value`, die Sie angeben, als Ziel auswählen. Wenn Sie planen, mithilfe des Parameters `targets` eine Assoziation für Dutzende oder Hunderte von Knoten zu erstellen, überprüfen Sie die folgenden Optionen für die Zielauswahl, bevor Sie mit dem Verfahren beginnen.

Bestimmte Knoten durch Angabe von IDs anvisieren

```
--targets Key=InstanceIds,Values=instance-id-1,instance-id-2,instance-id-3
```

```
--targets
 Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE,i-07782c72faEXAMPLE
```

Instances mithilfe von -Tags als Ziel auswählen

```
--targets Key=tag:tag-key,Values=tag-value-1,tag-value-2,tag-value-3
```

```
--targets Key=tag:Environment,Values=Development,Test,Pre-production
```

Zielknoten mithilfe von AWS Resource Groups

```
--targets Key=resource-groups:Name,Values=resource-group-name
```

```
--targets Key=resource-groups:Name,Values=WindowsInstancesGroup
```

Zielt auf alle Instanzen in der aktuellen Version ab AWS-Konto und AWS-Region

```
--targets Key=InstanceIds,Values=*
```

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - State Manager unterstützt nicht das Ausführen von Zuordnungen, die eine neue Version eines Dokuments verwenden, wenn dieses Dokument von einem anderen Konto

freigegeben wird. State Manager führt immer die `default`-Version eines Dokuments aus, wenn es von einem anderen Konto freigegeben wird, obwohl die Systems-Manager-Konsole anzeigt, dass eine neue Version verarbeitet wurde. Wenn Sie eine Zuordnung mit einer neuen Version eines Dokuments ausführen möchten, das von einem anderen Konto freigegeben wurde, müssen Sie die Dokumentversion auf `default` einstellen.

- Sie können mit der AWS CLI maximal fünf Tag-Schlüssel angeben. Wenn Sie den verwenden AWS CLI, müssen alle im `create-association` Befehl angegebenen Tag-Schlüssel dem Knoten aktuell zugewiesen sein. Sind sie das nicht, kann State Manager den Knoten nicht für eine Zuordnung anvisieren. Weitere Informationen zum Zuweisen von Tags zu Knoten finden Sie unter Markieren von Systems Manager-Ressourcen.
- Beim Erstellen der Zuordnung geben Sie die auch den Zeitplan für die Ausführung an. Geben Sie den Zeitplan mit einem cron- oder Rate-Ausdruck an. Weitere Informationen zu cron- und Rate-Ausdrücken finden Sie unter Cron- und Rate-Ausdrücke für Zuordnungen.

So erstellen Sie eine Zuordnung

1. Installieren und konfigurieren Sie den AWS CLI oder den AWS Tools for PowerShell, falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

2. Verwenden Sie das folgende Format, um einen Befehl zu erstellen, der eine State Manager-Zuordnung erstellt. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

   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"
```

In dem folgenden Beispiel wird eine Assoziaition für Knoten erstellt, die mit `"Environment,Linux"` getaggt sind. Die Assoziation verwendet das Dokument `AWS-UpdateSSMAgent`, um den SSM Agent auf den Ziel-Knoten jeden Sonntagmorgen um 2:00 Uhr UTC zu aktualisieren. Diese Assoziation wird jeweils auf maximal 10 Knoten gleichzeitig ausgeführt. Die Ausführung dieser Assoziation wird außerdem für ein bestimmtes Ausführungsintervall auf weiteren Knoten gestoppt, wenn die Fehlerzählung 5 überschreitet. Der Zuordnung wird für Compliance-Berichte der Schweregrad Mittel zugewiesen.

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
```

Das folgende Beispiel zielt durch Angabe eines Platzhalterzeichens (*) auf Knoten-IDs ab. Dadurch kann Systems Manager eine Zuordnung auf allen Knoten im aktuellen AWS-Konto und erstellen AWS-Region. Diese Assoziation wird jeweils auf maximal 10 Knoten gleichzeitig ausgeführt. Die Ausführung dieser Assoziation wird außerdem für ein bestimmtes Ausführungsintervall auf weiteren Knoten gestoppt, wenn die Fehlerzählung 5 überschreitet. Der Zuordnung wird für Compliance-Berichte der Schweregrad Mittel zugewiesen. Diese Assoziation verwendet einen Zeitplan-Offset, was bedeutet, dass sie zwei Tage nach dem angegebenen Cron-Zeitplan ausgeführt wird. Sie enthält auch den `ApplyOnlyAtCronInterval`-Parameter, der erforderlich ist, um den Zeitplan-Offset zu verwenden, und was bedeutet, dass die Assoziation nicht sofort nach ihrer Erstellung ausgeführt wird.

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
```

Im folgenden Beispiel wird eine Assoziation für Knoten in Ressourcengruppen erstellt. Die Gruppe trägt den Namen "HR-Department". Die Assoziation verwendet das Dokument `AWS-UpdateSSMAgent`, um den SSM Agent auf den Ziel-Knoten jeden Sonntagmorgen um 2:00 Uhr UTC zu aktualisieren. Diese Assoziation wird jeweils auf maximal 10 Knoten gleichzeitig ausgeführt. Die Ausführung dieser Assoziation wird außerdem für ein bestimmtes Ausführungsintervall auf weiteren Knoten gestoppt, wenn die Fehlerzählung 5 überschreitet. Der Zuordnung wird für Compliance-Berichte der Schweregrad Mittel zugewiesen. Diese Assoziation wird entsprechend dem angegebenen Cron-Zeitplan ausgeführt. Sie wird nicht unmittelbar nach dem Erstellen der Zuordnung ausgeführt.

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
```

Im folgenden Beispiel wird eine Zuordnung erstellt, die auf Knoten ausgeführt wird, die mit einer bestimmten Knoten-ID gekennzeichnet sind. Die Assoziation verwendet das SSM Agent-Dokument zur Aktualisierung von SSM Agent für die Zielknoten, wenn der Änderungskalender geöffnet ist. Die Zuordnung überprüft den Kalenderstatus, wenn er ausgeführt wird. Wenn der Kalender beim Start geschlossen ist und die Zuordnung nur einmal ausgeführt wird, wird diese nicht erneut ausgeführt, da das Ausführungsfenster der Zuordnung abgelaufen ist. Wenn der Kalender geöffnet ist, wird die Zuordnung entsprechend ausgeführt.

> **Note**
>
> Wenn Sie neue Knoten zu den Tags oder Ressourcengruppen hinzufügen, auf die eine Assoziation wirkt, wenn der Änderungskalender geschlossen ist, wird die Assoziation auf diese Knoten angewendet, sobald der Änderungskalender geöffnet wird.

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)"
```

Im folgenden Beispiel wird eine Zuordnung erstellt, die auf Knoten ausgeführt wird, die mit einer bestimmten Knoten-ID gekennzeichnet sind. Die ZuoAssoziation verwendet das Dokument SSM Agent, um jeden Sonntag um 2:00 Uhr SSM Agent auf den Zielknoten zu aktualisieren. Diese Assoziation wird nur zum angegebenen Cron-Zeitplan ausgeführt, wenn der Änderungskalender geöffnet ist. Wenn die Zuordnung erstellt wird, überprüft sie den Kalenderstatus. Wenn der Kalender geschlossen ist, wird die Zuordnung nicht angewendet. Wenn das Intervall zum Anwenden der Zuordnung am Sonntag um 2:00 Uhr beginnt, prüft die Zuordnung, ob der Kalender geöffnet ist. Wenn der Kalender geöffnet ist, wird die Zuordnung entsprechend ausgeführt.

> **Note**
>
> Wenn Sie neue Knoten zu den Tags oder Ressourcengruppen hinzufügen, auf die eine Assoziation wirkt, wenn der Änderungskalender geschlossen ist, wird die Assoziation auf diese Knoten angewendet, sobald der Änderungskalender geöffnet wird.

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**
>
> Wenn Sie die von Ihnen erstellte Zuordnung löschen, wird die Zuordnung nicht mehr auf Zielen dieser Zuordnung ausgeführt. Wenn Sie den Parameter `apply-only-at-cron-interval` angegeben haben, können Sie diese Option auch zurücksetzen. Geben Sie dazu den Parameter `no-apply-only-at-cron-interval` an, wenn Sie die Zuordnung über die Befehlszeile aktualisieren. Dieser Parameter erzwingt die sofortige Ausführung der Zuordnung nach dem Aktualisieren der Zuordnung und gemäß dem angegebenen Intervall.

Bearbeiten und Erstellen einer neuen Version einer Zuordnung

Sie können eine State Manager-Zuordnung bearbeiten, um den Namen, den Zeitplan, den Schweregrad oder die Ziele zu ändern. Sie können die Ausgabe des Befehls auch in einen Amazon Simple Storage Service (Amazon S3)-Bucket schreiben. Nachdem Sie eine Zuordnung bearbeitet haben, erstellt State Manager eine neue Version. Sie können unterschiedliche Versionen, wie in den folgenden Schritten beschrieben, nach der Bearbeitung anzeigen.

Die folgenden Verfahren beschreiben, wie Sie mithilfe der Systems Manager Manager-Konsole () und AWS Command Line Interface AWS Tools for PowerShell (Tools für AWS CLI PowerShell) eine neue Version einer Zuordnung bearbeiten und erstellen.

> **Important**
>
> State Manager unterstützt nicht das Ausführen von Zuordnungen, die eine neue Version eines Dokuments verwenden, wenn dieses Dokument von einem anderen Konto freigegeben wird. State Manager läuft immer die `default`-Version eines Dokuments, wenn es von einem anderen Konto freigegeben wird, obwohl die Systems-Manager-Konsole anzeigt, dass eine neue Version verarbeitet wurde. Wenn Sie eine Zuordnung mit einer neuen Version eines Dokuments ausführen möchten, das von einem anderen Konto freigegeben wurde, müssen Sie die Dokumentversion auf `default` einstellen.

Bearbeiten einer Zuordnung (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine neue Version einer Zuordnung bearbeiten und erstellen.

> **Note**
>
> Dieser Vorgang erfordert, dass Sie über Schreibzugriff auf einen vorhandenen Amazon S3-Bucket verfügen. Wenn Sie Amazon S3 bisher nicht verwendet haben, bedenken Sie, dass Gebühren für die Nutzung von Amazon S3 anfallen. Weitere Informationen zum Erstellen eines Buckets finden Sie unter Erstellen eines Buckets.

So bearbeiten Sie eine State Manager-Zuordnung

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie die in Erstellen einer Zuordnung (Befehlszeile) erstellte Zuordnung und wählen Sie anschließend Edit (Bearbeiten).
4. Geben Sie im Feld Name einen neuen Namen ein.
5. Wählen Sie im Abschnitt Specify schedule eine neue Option.
6. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Dienstrolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

7. Wählen Sie Edit association. Konfigurieren Sie die Zuordnung so, dass sie Ihre aktuellen Anforderungen erfüllt.
8. Wählen Sie auf der Seite Associations den Namen der bearbeiteten Zuordnung und anschließend die Registerkarte Versions. Das System listet alle Version der Zuordnung auf, die Sie erstellt und bearbeitet haben.
9. Öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
10. Wählen Sie den Namen des Amazon S3-Buckets, den Sie zum Speichern von Befehlsausgaben angegeben haben, und wählen Sie dann den Ordner, dessen Name der ID dem Knoten entspricht, der die Assoziation ausgeführt hat. (Wenn Sie festgelegt haben, Ausgaben in einem Ordner im Bucket zu speichern, öffnen Sie diesen zuerst.)
11. Zeigen Sie die `stdout`-Datei in einer tieferen Ebenen im Ordner `awsrunPowerShell` an.
12. Wählen Sie Open oder Download, um den Hostnamen anzuzeigen.

Bearbeiten einer Zuordnung (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die AWS CLI (unter Linux oder Windows) verwenden oder AWS Tools for PowerShell eine neue Version einer Assoziation bearbeiten und erstellen.

So bearbeiten Sie eine State Manager-Zuordnung

1. Installieren und konfigurieren Sie die AWS CLI oder die AWS Tools for PowerShell, falls Sie dies noch nicht getan haben.

Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

2. Verwenden Sie das folgende Format, um einen Befehl zum Bearbeiten und Erstellen einer neuen Version einer vorhandenen State Manager-Zuordnung zu erstellen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

> **Important**
>
> Wenn Sie `UpdateAssociation` aufrufen, löscht das System alle optionalen Parameter aus der Anforderung und überschreibt die Zuordnung mit Nullwerten für diese Parameter. Dies ist beabsichtigt. Sie müssen alle optionalen Parameter im Aufruf angeben, auch wenn Sie die Parameter nicht ändern. Dies umfasst den `Name`-Parameter. Bevor Sie diese API-Aktion aufrufen, empfehlen wir Ihnen, die DescribeAssociationAPI-Operation aufzurufen und sich alle optionalen Parameter zu notieren, die für Ihren `UpdateAssociation` Aufruf erforderlich sind.

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
```

Im folgenden Beispiel wird eine vorhandene Zuordnung aktualisiert, um den Namen in `TestHostnameAssociation2` zu ändern. Die neue Zuordnungsversion wird stündlich ausgeführt und schreibt die Ausgabe der Befehle in den angegebenen Amazon S3-Bucket.

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=DOC-EXAMPLE-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=DOC-EXAMPLE-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 DOC-EXAMPLE-BUCKET `
  -S3Location_OutputS3KeyPrefix logs `
  -S3Location_OutputS3Region us-east-1 `
  -ScheduleExpression "cron(0 */1 * * ? *)"
```

Im folgenden Beispiel wird eine vorhandene Zuordnung aktualisiert, um den Namen in `CalendarAssociation` zu ändern. Die neue Zuordnung wird ausgeführt, wenn der Kalender geöffnet ist, und schreibt die Befehlsausgabe in den angegebenen Amazon S3-Bucket.

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=DOC-EXAMPLE-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=DOC-EXAMPLE-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 DOC-EXAMPLE-BUCKET `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar2"
```

Im folgenden Beispiel wird eine vorhandene Zuordnung aktualisiert, um den Namen in `MultiCalendarAssociation` zu ändern. Die neue Zuordnung wird ausgeführt, wenn die Kalender geöffnet sind, und schreibt die Befehlsausgabe in den angegebenen Amazon S3-Bucket.

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=DOC-EXAMPLE-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=DOC-EXAMPLE-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 DOC-EXAMPLE-BUCKET `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2"
```

3. Um die neue Version der Zuordnung anzuzeigen, führen Sie den folgenden Befehl aus.

   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 *
   ```

   Das System gibt unter anderem folgende Informationen zurück

   Linux & macOS

   ```
   {
   ```

```
    "AssociationDescription": {
        "ScheduleExpression": "cron(0 */1 * * ? *)",
        "OutputLocation": {
            "S3Location": {
                "OutputS3KeyPrefix": "logs",
                "OutputS3BucketName": "DOC-EXAMPLE-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": "DOC-EXAMPLE-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}
```

Löschen von Zuordnungen

Das folgende Verfahren beschreibt, wie Sie eine State Manager Zuordnung mithilfe der AWS Systems Manager Konsole löschen.

Löschen einer Zuordnung

Gehen Sie wie folgt vor, um eine Zuordnung mithilfe der AWS Systems Manager -Konsole zu löschen.

Löschen einer Zuordnung

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie eine Zuordnung aus und wählen Sie Löschen aus.

Ausführen von Auto-Scaling-Gruppen mit Zuordnungen

Die bewährte Methode beim Verwenden von Zuordnungen zum Ausführen von Auto-Scaling-Gruppen besteht darin, Tag-Ziele zu verwenden. Wenn Sie Tags nicht verwenden, könnten Sie das Zuordnungslimit erreichen.

Wenn alle Knoten mit demselben Schlüssel und demselben Wert versehen sind, benötigen Sie nur eine Assoziation, um Ihre Auto-Scaling-Gruppe auszuführen. Im folgenden Verfahren wird beschrieben, wie Sie so eine Zuordnung erstellen.

Erstellen einer Zuordnung, auf der Auto-Scaling-Gruppen ausgeführt wird

1. Stellen Sie sicher, dass alle Knoten in der Auto-Scaling-Gruppe mit demselben Schlüssel und demselben Wert versehen sind. Weitere Informationen zum Markieren von Knoten finden Sie unter Markieren von Auto-Scaling-Gruppen und Knoten im AWS Auto Scaling-Benutzerhandbuch.
2. Erstellen Sie eine Zurodnung unter Verwendung des Verfahrens in Arbeiten mit Zuordnungen in Systems Manager.

   Wenn Sie in der Konsole arbeiten, wählen Sie Specify instance tags (Instance-Tags angeben) im Feld Targets (Ziele). Geben Sie für Instance-Tags den Tag-Schlüssel und -Welt für Ihre Auto-Scaling-Gruppe ein.

   Wenn Sie die AWS Command Line Interface (AWS CLI) verwenden, geben Sie `--targets Key=tag:tag-key,Values=tag-value` an, bei denen Schlüssel und Wert mit dem übereinstimmen, mit dem Sie Ihre Knoten gekennzeichnet haben.

Anzeigen von Zuordnungsverläufen

Sie können alle Ausführungen für eine bestimmte Zuordnungs-ID mithilfe der API-Operation DescribeAssociationExecutions anzeigen. Verwenden Sie diesen Vorgang, um Status, detaillierten Status, Ergebnisse, Uhrzeit der letzten Ausführung und weitere Informationen zu einer State Manager-Zuordnung einzusehen. State Manager ist eine Funktion von AWS Systems Manager. Diese API-Operation enthält auch Filter, mit denen Sie entsprechend den von Ihnen festgelegten Kriterien nach Zuordnungen suchen können. Sie können beispielsweise genaue Angaben zu Datum und Uhrzeit machen und mithilfe eines GREATER_THAN-Filters Ausführungen anzeigen, die nach dem angegebenen Datum und der angegebenen Uhrzeit verarbeitet wurden.

Beispiel: Wenn eine Zuordnung nicht ausgeführt werden kann, können Sie die Details einer bestimmten Ausführung mithilfe der API-Operation DescribeAssociationExecutionTargets anzeigen. Diese Operation zeigt Ihnen die Ressourcen, wie z. B. Knoten-IDs, wo die Assoziation ausgeführt wurde, sowie die verschiedenen Assoziationsstatusarten an. Anschließend können Sie sehen, bei welchen Ressourcen oder Knoten eine Assoziation nicht ausgeführt werden konnte. Anhand der Ressourcen-ID können Sie dann die Details der Befehlsausführung anzeigen, um zu bestimmen, welcher Schritt in einem Befehl fehlgeschlagen ist.

Die Beispiele in diesem Abschnitt umfassen auch Informationen zur Verwendung der API-Operation StartAssociationsOnce, um eine Zuordnung nur einmal zum Zeitpunkt der Erstellung auszuführen. Sie können mithilfe dieser API-Operation fehlgeschlagenen Zuordnungsausführungen nachgehen. Wenn Sie sehen, dass eine Zuordnung fehlgeschlagen ist, können Sie eine Änderung an der Ressource vornehmen und dann die Zuordnung sofort ausführen, um zu sehen, ob die Änderung an der Ressource nun eine erfolgreiche Ausführung der Zuordnung zulässt.

> **Note**
>
> API-Vorgänge, die während der Ausführung einer Zuordnung durch das SSM-Dokument initiiert werden, werden in AWS CloudTrail nicht protokolliert.

Anzeigen von Zuordnungsverläufen (Konsole)

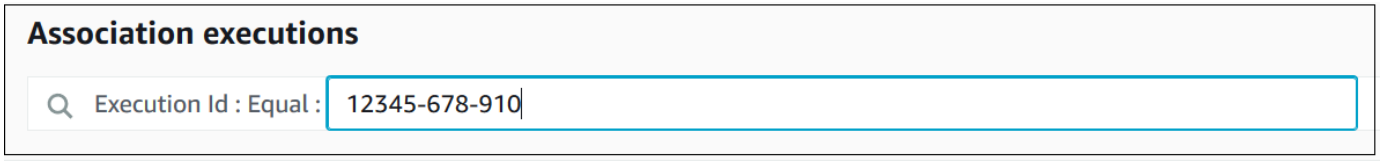
Mit dem folgenden Verfahren können Sie den Ausführungsverlauf für eine bestimmte Zuordnungs-ID und anschließend Ausführungsdetails für eine oder mehrere Ressourcen anzeigen.

So zeigen Sie den Ausführungsverlauf für eine bestimmte Zuordnungs-ID an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie State Manager.
3. Wählen Sie im Feld Association id (Zuordnungs-ID) eine Zuordnung aus, deren Verlauf Sie anzeigen möchten.
4. Klicken Sie auf die Schaltfläche View details (Details ansehen).
5. Wählen Sie die Registerkarte Execution history (Ausführungsverlauf).
6. Wählen Sie eine Zuordnung aus, für die Sie Ausführungsdetails auf Ressourcenebene anzeigen möchten. Wählen Sie z. B. eine Zuordnung mit dem Status Failed (Fehlgeschlagen) aus.

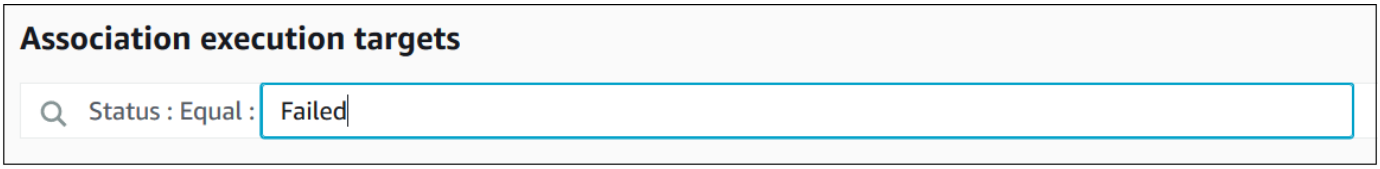
Anschließend können Sie die Ausführungsdetails für die Koten anzeigen, bei denen das Ausführen der Assoziation fehlgeschlagen ist.

Verwenden Sie die Suchfeldfilter zur Suche nach der Ausführung, für die Sie Details anzeigen möchten.



7. Wählen eine Ausführungs-ID aus. Die Seite Association execution targets (Zuordnungsausführungsziele) wird geöffnet. Diese Seite zeigt alle Ressourcen an, die die Zuordnung ausgeführt haben.
8. Wählen Sie eine Ressourcen-ID aus, um spezifische Informationen zu dieser Ressource anzuzeigen.

   Verwenden Sie die Suchfeldfilter zur Suche nach der Ressource, für die Sie Details anzeigen möchten.



9. Wenn Sie eine Zuordnung untersuchen, die nicht ausgeführt werden konnte, können Sie mit der Schaltfläche Apply association now (Zuordnung nun anwenden) eine Zuordnung nur einmal zum Zeitpunkt der Erstellung ausführen. Nachdem Sie Änderungen an der Ressource vorgenommen haben, bei der die Zuordnung nicht ausgeführt werden konnte, wählen Sie den Link Association ID (Zuordnungs-ID) im Navigations-Breadcrumb aus.
10. Klicken Sie auf die Schaltfläche Apply association now (Zuordnung nun anwenden). Wenn die Ausführung abgeschlossen ist, überprüfen Sie, ob die Zuordnungsausführung erfolgreich war.

Anzeigen von Zuordnungsverläufen (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie Sie die AWS Command Line Interface (AWS CLI) (unter Linux oder Windows) oder AWS Tools for PowerShell verwenden, um den Ausführungsverlauf für eine bestimmte Zuordnungs-ID anzuzeigen. Im Anschluss an dieses Verfahren wird beschrieben, wie Sie Ausführungsdetails für eine oder mehrere Ressourcen anzeigen.

So zeigen Sie den Ausführungsverlauf für eine bestimmte Zuordnungs-ID an

1. Installieren und konfigurieren Sie die AWS CLI oder AWS Tools for PowerShell, falls noch nicht erfolgt.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

2. Führen Sie den folgenden Befehl aus, um eine Liste von Ausführungen für eine bestimmte Zuordnungs-ID anzuzeigen.

   Linux & macOS

   ```
   aws ssm describe-association-executions \
     --association-id ID \
     --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN
   ```

   > **Note**
   >
   > Dieser Befehl wendet einen Filter an, um die Ergebnisse auf solche Ausführungen einzuschränken, die nach einem bestimmten Datum und einer bestimmten Uhrzeit aufgetreten sind. Wenn Sie alle Ausführungen für eine bestimmte Zuordnungs-ID anzeigen möchten, entfernen Sie den Parameter `--filters` und den Wert `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**
   >
   > Dieser Befehl wendet einen Filter an, um die Ergebnisse auf solche Ausführungen einzuschränken, die nach einem bestimmten Datum und einer bestimmten Uhrzeit aufgetreten sind. Wenn Sie alle Ausführungen für eine bestimmte Zuordnungs-

ID anzeigen möchten, entfernen Sie den Parameter `--filters` und den Wert `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**
>
> Dieser Befehl wendet einen Filter an, um die Ergebnisse auf solche Ausführungen einzuschränken, die nach einem bestimmten Datum und einer bestimmten Uhrzeit aufgetreten sind. Wenn Sie alle Ausführungen für eine bestimmte Zuordnungs-ID anzeigen möchten, entfernen Sie den Parameter `-Filter` und den Wert `@{"Key"="CreatedTime";"Value"="2019-06-01T19:15:38.372Z";"Type"="GREAT`

Das System gibt unter anderem folgende Informationen zurück

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
```

Sie können die Ergebnisse einschränken, indem Sie einen oder mehrere Filter verwenden. Das folgende Beispiel gibt alle Zuordnungen zurück, die vor einem bestimmten Datum und einer bestimmten Uhrzeit ausgeführt wurden.

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"}
```

Das folgende Beispiel gibt alle Zuordnungen zurück, die nach einem bestimmten Datum und einer bestimmten Uhrzeit erfolgreich ausgeführt wurden.

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. Führen Sie den folgenden Befehl aus, um alle Ziele anzuzeigen, an denen die betreffende Ausführung ausgeführt wurde.

   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
   ```

   Sie können die Ergebnisse einschränken, indem Sie einen oder mehrere Filter verwenden. Das folgende Beispiel gibt Informationen über alle Ziele zurück, an denen die betreffende Zuordnung nicht ausgeführt werden konnte.

   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"
    }
```

Das folgende Beispiel gibt Informationen über einen bestimmten verwalteten Knoten zurück, bei dem eine Assoziation nicht ausgeführt werden konnte.

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. Wenn Sie eine Zuordnung untersuchen, die nicht ausgeführt werden konnte, können Sie mit der API-Operation StartAssociationsOnce eine Zuordnung sofort und nur einmal ausführen. Nachdem Sie Änderungen an der Ressource vornehmen, bei der die Zuordnung nicht ausgeführt werden konnte, führen Sie den folgenden Befehl aus, um die Zuordnung sofort und nur einmal auszuführen.

   Linux & macOS

   ```
   aws ssm start-associations-once \
     --association-id ID
   ```

   Windows

   ```
   aws ssm start-associations-once ^
     --association-id ID
   ```

   PowerShell

   ```
   Start-SSMAssociationsOnce `
     -AssociationId ID
   ```

Arbeiten mit Zuordnungen mithilfe von IAM

State Manager, eine Funktion von AWS Systems Manager, verwendet Ziele, um auszuwählen, mit welchen Instances Sie Ihre Verknüpfungen konfigurieren. Ursprünglich wurden Zuordnungen erstellt,

indem ein Dokumentname (`Name`) und Instance-ID (`InstanceId`) angegeben wurden. Dadurch wurde eine Verknüpfung zwischen einem Dokument und einer Instanz oder einem verwalteten Knoten erstellt. Zuordnungen wurden durch diese Parameter identifiziert. Diese Parameter sind jetzt veraltet, werden aber weiterhin unterstützt. Die Ressourcen `instance` und `managed-instance` wurden als Ressourcen zu Aktionen mit `Name` und `InstanceId` hinzugefügt.

AWS Identity and Access Management Das Verhalten bei der Durchsetzung von Richtlinien (IAM) hängt vom Typ der angegebenen Ressource ab. Ressourcen für State Manager-Vorgänge werden nur basierend auf der übergebenen Anforderung erzwungen. State Manager führt keine tiefe Prüfung auf die Eigenschaften von Ressourcen in Ihrem Konto durch. Eine Anforderung wird nur anhand von Richtlinienressourcen validiert, wenn der Anforderungsparameter die angegebenen Richtlinienressourcen enthält. Wenn Sie beispielsweise eine Instance im Ressourcenblock angeben, wird die Richtlinie erzwungen, wenn die Anforderung den `InstanceId`-Parameter verwendet. Der `Targets`-Parameter für jede Ressource im Konto wird nicht für diese `InstanceId` überprüft.

Im Folgenden sind einige Fälle mit verwirrendem Verhalten dargestellt:

- DescribeAssociationDeleteAssociation, und UpdateAssociationverwenden Sie`instance`,, und `document` Ressourcen`managed-instance`, um die veraltete Art des Verweises auf Assoziationen anzugeben. Dies beinhaltet alle Zuordnungen, die mit dem veralteten `InstanceId`-Parameter erstellt wurden.
- CreateAssociationCreateAssociationBatch, und UpdateAssociationverwenden Sie `instance` und `managed-instance` Ressourcen, um die veraltete Art der Bezugnahme auf Assoziationen zu spezifizieren. Dies beinhaltet alle Zuordnungen, die mit dem veralteten `InstanceId`-Parameter erstellt wurden. Der `document`-Ressourcentyp ist Teil der veralteten Methode, auf Zuordnungen zu verweisen und ist eine tatsächliche Eigenschaft einer Zuordnung. Das bedeutet, dass Sie IAM-Richtlinien mit `Allow` oder `Deny` Berechtigungen für beide `Create` und `Update` Aktionen auf der Grundlage des Dokumentnamens erstellen können.

Weitere Informationen zur Verwendung von IAM-Richtlinien mit Systems Manager finden Sie unter Identity and Access Management für AWS Systems Manager oder Aktionen, Ressourcen und Bedingungsschlüssel für AWS Systems Manager in der Service Authorization-Referenz.

Walkthroughs zum AWS Systems ManagerState Manager

Die folgenden Anleitungen zeigen, wie State Manager-Zuordnungen mit der Systems Manager-Konsole oder der AWS Command Line Interface (AWS CLI) erstellt und konfiguriert werden. Die

Anleitungen veranschaulichen darüber hinaus, wie allgemeine Verwaltungsaufgaben mit State Manager, eine Funktion von AWS Systems Manager, automatisch ausgeführt werden können.

Themen

- Walkthrough: Creating Erstellen von Zuordnungen, die MOF-Dateien ausführen
- Exemplarische Vorgehensweise: Erstellen von Verknüpfungen, die Playbooks ausführen Ansible
- Exemplarische Vorgehensweise: Erstellen von Verknüpfungen, die Rezepte ausführen Chef
- Anleitung: Automatische Aktualisierung von SSM Agent (CLI)
- Anleitung: Automatische Aktualisierung von PV-Treibern auf EC2-Instances für Windows Server (Konsole)

Walkthrough: Creating Erstellen von Zuordnungen, die MOF-Dateien ausführen

Mithilfe des `AWS-ApplyDSCMofs` SSM-Dokuments können Sie MOF-Dateien (Managed Object Format) ausführenState Manager, um einen gewünschten Status auf verwalteten Windows Server-Knoten mit der AWS Systems Manager Fähigkeit von zu erzwingen. Das `AWS-ApplyDSCMofs`-Dokument weist zwei Ausführungsmodi auf. Mit dem ersten Modus können Sie die Assoziation so konfigurieren, dass sie die verwalteten Knoten scannt und meldet, wenn sie den in den MOF-Dateien definierten gewünschten Status aufweisen. Im zweiten Modus können Sie die MOF-Dateien ausführen und die Konfiguration Ihrer Knoten basierend auf den Ressourcen und ihren in den MOF-Dateien definierten Werten ändern. Mit dem `AWS-ApplyDSCMofs`-Dokument können Sie MOF-Konfigurationsdateien von Amazon Simple Storage Service (Amazon S3), einem lokal freigegebenen Verzeichnis, oder einer sicheren Website mit einer HTTPS-Domain herunterladen und ausführen.

State Manager protokolliert und meldet den Status der einzelnen MOF-Dateiausführungen bereits während die Zuordnungen ausgeführt werden. Darüber hinaus meldet State Manager die Ausgabe der MOF-Dateiausführungen als Compliance-Ereignis, das Sie auf der AWS Systems Manager Compliance-Seite anzeigen können.

Die Ausführung von MOF-Dateien basiert auf der Windows PowerShell Desired State Configuration (PowerShell DSC). PowerShell DSC ist eine deklarative Plattform, die für die Konfiguration, Bereitstellung und Verwaltung von Windows-Systemen verwendet wird. PowerShell DSC ermöglicht es Administratoren, in einfachen Textdokumenten, den sogenannten DSC-Konfigurationen, zu beschreiben, wie ein Server konfiguriert werden soll. Eine PowerShell DSC-Konfiguration ist ein spezielles PowerShell Skript, das angibt, was zu tun ist, aber nicht, wie es zu tun ist. Bei der Ausführung der Konfiguration wird eine MOF-Datei erzeugt. Die MOF-Datei kann auf einen

oder mehrere Server angewendet werden, um die gewünschte Konfiguration für diese Server zu erreichen. PowerShell DSC-Ressourcen übernehmen die eigentliche Aufgabe, die Konfiguration durchzusetzen. Weitere Informationen finden Sie unter Übersicht über die Konfiguration des PowerShell gewünschten Windows-Zustands.

Themen

- Verwenden von Amazon S3 zum Speichern von Artefakten
- Auflösen von Anmeldeinformationen in MOF-Dateien
- Verwenden von Token in MOF-Dateien
- Voraussetzungen
- Erstellen einer Zuordnung, die MOF-Dateien ausführt
- Fehlerbehebung
- Anzeigen von Details zur DSC-Ressourcen-Compliance

Verwenden von Amazon S3 zum Speichern von Artefakten

Wenn Sie Amazon S3 zum Speichern von PowerShell Modulen, MOF-Dateien, Compliance-Berichten oder Statusberichten verwenden, AWS Systems Manager SSM Agent müssen die von verwendete AWS Identity and Access Management (IAM) -Rolle `GetObject` und die `ListBucket` Berechtigungen für den Bucket vorhanden sein. Ohne diese Berechtigungen gibt das System einen Zugriff verweigert Fehler zurück. Unten finden Sie wichtige Informationen zum Speichern von Artefakten in Amazon S3.

- Wenn sich der Bucket in einem anderen befindet AWS-Konto, erstellen Sie eine Bucket-Ressourcenrichtlinie, die dem Konto (oder der IAM-Rolle) `GetObject` und Berechtigungen gewährt. `ListBucket`
- Wenn Sie benutzerdefinierte DSC-Ressourcen verwenden möchten, können Sie diese Ressourcen aus einem Amazon S3-Bucket herunterladen. Sie können sie auch automatisch aus der PowerShell Galerie installieren.
- Wenn Sie Amazon S3 als Modulquelle verwenden, laden Sie das Modul als Zip-Datei im folgenden Format mit Groß- und Kleinschreibung hoch: *`ModuleName`*_*`ModuleVersion`*.zip. Zum Beispiel: _1.0.0.zip MyModule.
- Alle Dateien müssen im sich im Stammverzeichnis des Buckets befinden. Ordnerstrukturen werden nicht unterstützt.

Auflösen von Anmeldeinformationen in MOF-Dateien

Anmeldeinformationen werden mithilfe von AWS Secrets Manager oder AWS Systems Manager Parameter Store aufgelöst. Auf diese Weise können Sie eine automatische Rotation der Anmeldeinformationen einrichten. Dies ermöglicht auch DSC, Anmeldeinformationen automatisch auf Ihren Servern zu verteilen ohne die MOF-Dateien erneut bereitstellen zu müssen.

Um ein AWS Secrets Manager Geheimnis in einer Konfiguration zu verwenden, erstellen Sie ein PSCredential-Objekt, wobei der Benutzername der SecretId oder SecretARN des Geheimnisses ist, das die Anmeldeinformationen enthält. Sie können für das Passwort einen beliebigen Wert angeben. Der Wert wird ignoriert. Im Folgenden sehen Sie ein Beispiel.

```
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
       }
    }
}
```

Kompilieren Sie Ihr MOF mithilfe der Einstellung in den Konfigurationsdaten PsAllowPlaintextPassword . Dies ist kein besonderes Risiko, weil die Anmeldeinformationen nur einen Bezeichner enthalten.

Stellen Sie in Secrets Manager sicher, dass der Knoten in einer von IAM verwalteten Richtlinie und optional in der Secret Resource Policy, falls vorhanden, GetSecretValue Zugriff hat. Für die Arbeit mit DSC muss das Geheimnis das folgende Format aufweisen.

```
{ 'Username': 'a_name', 'Password': 'a_password' }
```

Das Secret kann weitere Eigenschaften (z. B. Eigenschaften für die Rotation) haben, aber es muss mindestens den Benutzernamen und das Passwort enthalten.

Es wird empfohlen, eine Rotationsmethode für mehrere Benutzer zu verwenden, bei der Sie zwei verschiedene Benutzernamen und Kennwörter verwenden und die AWS Lambda Rotationsfunktion zwischen diesen wechselt. Diese Methode ermöglicht Ihnen, mehrere aktive Konten zu haben, ohne in Gefahr zu laufen, dass Benutzer bei einer Rotation ausgesperrt werden.

Verwenden von Token in MOF-Dateien

Token bieten Ihnen die Möglichkeit, Eigenschaftswerte von Ressourcen zu ändern, nachdem die MOF-Datei kompiliert wurde. Auf diese Weise können Sie häufig verwendete MOF-Dateien auf mehreren Servern mit ähnlichen Konfigurationen wiederverwenden.

Die Ersetzung der Token funktioniert nur für Ressourceneigenschaften des Typs `String`. Wenn Ihre Ressource jedoch eine eingebettete CIM-Knoten-Eigenschaft hat, löst sie auch Token von `String`-Eigenschaften in diesem CIM-Knoten auf. Sie können die Token-Ersetzung nicht für Zahlen oder Arrays verwenden.

Stellen Sie sich beispielsweise ein Szenario vor, in dem Sie die xComputerManagement Ressource verwenden und den Computer mithilfe von DSC umbenennen möchten. Normalerweise benötigen Sie eine dedizierte MOF-Datei für diesen Computer. Mit der Token-Unterstützung können Sie eine MOF-Datei erstellen und diese auf alle Ihre Knoten anwenden. Sie können in der `ComputerName`-Eigenschaft in der MOF-Datei anstelle des festkodierten Computernamens ein Token vom Typ Instance-Tag verwenden. Der Wert wird während beim Parsing der MOF-Datei aufgelöst. Sehen Sie sich das folgende -Beispiel an.

```
Configuration MyConfig
{
    xComputer Computer
    {
        ComputerName = '{tag:ComputerName}'
    }
}
```

Sie setzen dann ein Tag entweder für den verwalteten Knoten in der Systems Manager-Konsole oder ein Amazon Elastic Compute Cloud (Amazon EC2)-Tag in der Amazon EC2-Konsole. Wenn Sie das Dokument ausführen, ersetzt das Skript den Wert des Instanz-Tags durch das Token {tag:ComputerName}.

Sie können auch mehrere Tags in einer einzigen Eigenschaft kombinieren, wie im folgenden Beispiel gezeigt.

```
Configuration MyConfig
{
    File MyFile
    {
        DestinationPath = '{env:TMP}\{tag:ComputerName}'
        Type = 'Directory'
    }
}
```

Es gibt fünf verschiedene Arten von Token, die Sie verwenden können:

- tag: Amazon EC2-Tag oder Tag auf einem verwalteten Knoten.
- tagb64: Dies ist das gleiche wie tag, aber das System verwendet base64, um den Wert zu dekodieren. Auf diese Weise können Sie in Tag-Werten Sonderzeichen verwenden.
- env: Löst Umgebungsvariablen auf.
- ssm: Parameter Store-Werte. Es werden nur die Typen String und Secure String unterstützt.
- tagssm: Dies ist dasselbe wie Tag, aber wenn das Tag auf dem Knoten nicht festgelegt ist, versucht das System, den Wert aus einem Systems Manager-Parameter mit demselben Namen aufzulösen. Dies ist nützlich, wenn Sie einen „globalen Standardwert“ benötigen, den Sie auf einzelnen Knoten außer Kraft setzen möchten (z. B. bei One-Box-Bereitstellungen).

Es folgt ein Parameter Store-Beispiel, bei dem der Token-Typ `ssm` verwendet wird.

```
File MyFile
{
    DestinationPath = "C:\ProgramData\ConnectionData.txt"
    Content = "{ssm:%servicePath%/ConnectionData}"
}
```

Token spielen eine wichtige Rolle bei der Reduzierung von redundantem Code, weil MOF-Dateien generisch und wiederverwendbar werden. Wenn Sie serverspezifische MOF-Dateien vermeiden können, benötigen Sie auch keinen Service, um die MOF-Datei zu erstellen. Ein Service zur Erstellung von MOF-Dateien erhöht die Kosten, verlangsamt die Bereitstellungszeiten und erhöht das Risiko einer Konfigurationsdrift zwischen gruppierten Knoten aufgrund abweichender Modulversionen verglichen mit den Versionen, die zum Zeitpunkt der Kompilierung der MOF-Dateien auf dem Build-Server installiert wurden.

Voraussetzungen

Bevor Sie eine Assoziation erstellen, die MOF-Dateien ausführt, überprüfen Sie, ob Ihre verwalteten Knoten die folgenden Voraussetzungen erfüllen:

- Windows PowerShell Version 5.0 oder höher. Weitere Informationen finden Sie unter PowerShell Windows-Systemanforderungen auf Microsoft.com.
- AWS Tools for Windows PowerShell Version 3.3.261.0 oder höher
- SSM Agent, Version 2.2 oder höher.

Erstellen einer Zuordnung, die MOF-Dateien ausführt

So erstellen Sie eine Zuordnung, die MOF-Dateien ausführt

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie State Manager und dann Create association (Zuordnung wählen) aus.
4. Geben Sie im Feld Name einen Namen an. Dies ist zwar optional, wird aber empfohlen. Ein Name kann Ihnen helfen, den Zweck der Zuordnung zu verstehen, nachdem Sie sie erstellt haben. Der Name darf keine Leerzeichen enthalten.
5. Wählen Sie in der Liste Dokument die Option **`AWS-ApplyDSCMofs`** aus.
6. Geben Sie im Abschnitt Parameters (Parameter) die benötigten Angaben für die erforderlichen und die optionalen Eingabeparameter ein.

    a. Mofs To Apply (Anzuwendende MOF-Dateien): Geben Sie eine oder mehrere MOF-Dateien an, die mit dieser Zuordnung ausgeführt werden sollen. Um eine Liste von MOF-Dateien anzugeben, trennen Sie die Dateinamen mit Kommas. Sie können den Speicherort der MOF-Dateien alternativ wie folgt angeben:

    - Eine Amazon S3-Bucket-Bezeichnung. Bucketnamen müssen Kleinbuchstaben angegeben werden. Geben Sie diese Informationen in dem folgenden Format an.

    ```
    s3:DOC-EXAMPLE-BUCKET:MOF_file_name.mof
    ```

    Wenn Sie ein angeben möchten AWS-Region, verwenden Sie das folgende Format.

```
s3:bucket_Region:DOC-EXAMPLE-BUCKET:MOF_file_name.mof
```

- Eine sichere Website. Geben Sie diese Informationen in dem folgenden Format an.

  ```
  https://domain_name/MOF_file_name.mof
  ```

  Ein Beispiel.

  ```
  https://www.example.com/TestMOF.mof
  ```

- Ein Dateisystem auf einer lokalen Freigabe. Geben Sie diese Informationen in dem folgenden Format an.

  ```
  \server_name\shared_folder_name\MOF_file_name.mof
  ```

  Ein Beispiel.

  ```
  \StateManagerAssociationsBox\MOFs_folder\MyMof.mof
  ```

b. Service Path (Service-Pfad): (Optional) Ein Service-Pfad ist entweder das Präfix eines Amazon S3-Buckets, in den Sie Berichte und Statusinformationen schreiben möchten, oder ein Service-Pfad ist ein Pfad für Parameter Store-Parameter-basierte Tags. Wenn das System Parameter-basierte Tags auflöst, verwendet es {ssm:%servicePath %/*parameter_name*}, um den servicePath-Wert in den Parameternamen hineinzubringen. *Wenn Ihr Dienstpfad beispielsweise "WebServers/Production" lautet, löst das System den Parameter wie folgt auf: WebServers /Production/parameter_name.* Dies ist nützlich, wenn Sie in einem Konto mehrere Umgebungen ausführen.

c. Report Bucket Name (Bucket-Name für Berichte): (Optional) Geben Sie den Namen eines Amazon S3-Buckets ein, in den Sie Compliance-Daten schreiben möchten. Berichte werden in diesem Bucket im JSON-Format gespeichert.

> **Note**
>
> Sie können dem Bucketnamen ein Präfix voranstellen, um die Region anzugeben, in der sich der Bucket befindet. Beispiel: us-west-2:MyMOFBucket. Wenn Sie einen Proxy für Amazon S3-Endpunkte in einer bestimmten Region verwenden, die us-

east-1 nicht enthält, stellen Sie dem Bucket-Namen eine Region voran. Wenn dem Bucketname kein Präfix vorangestellt ist, wird die Bucketregion unter Verwendung des Endpunkts us-east-1 automatisch erkannt.

d. Mof Operation Mode (MOF-Betriebsmodus): Wählen Sie das State Manager-Verhalten beim Ausführen der Zuordnung **`AWS-ApplyDSCMofs`** aus:

- Apply (Anwenden): Korrigiert Knoten-Konfigurationen, die nicht konform sind.
- ReportOnly: Korrigieren Sie keine Knotenkonfigurationen, sondern protokollieren Sie stattdessen alle Konformitätsdaten und melden Sie Knoten, die nicht konform sind.

e. Status Bucket Name (Bucket-Name für Status): (Optional) Geben Sie den Namen eines Amazon S3-Buckets ein, in den Sie den MOF-Ausführungsstatus schreiben möchten. Diese Statusberichte sind Singleton-Zusammenfassungen des letzten Compliance-Laufs eines Knotens. Dies bedeutet, dass der Bericht überschrieben wird, wenn die Zuordnung das nächste Mal MOF-Dateien ausführt.

> **Note**
>
> Sie können dem Bucketnamen ein Präfix voranstellen, um die Region anzugeben, in der sich der Bucket befindet. Ein Beispiel: `us-west-2:DOC-EXAMPLE-BUCKET`. Wenn Sie einen Proxy für Amazon S3-Endpunkte in einer bestimmten Region verwenden, die us-east-1 nicht enthält, stellen Sie dem Bucket-Namen eine Region voran. Wenn dem Bucketname kein Präfix vorangestellt ist, wird die Bucketregion unter Verwendung des Endpunkts us-east-1 automatisch erkannt.

f. Name des Quell-Buckets für das Modul: (Optional) Geben Sie den Namen eines Amazon S3 S3-Buckets ein, der PowerShell Moduldateien enthält. Wenn Sie None (Keine) angeben, wählen Sie True (Wahr) für die nächste Option, Allow PS Gallery Module Source.

> **Note**
>
> Sie können dem Bucketnamen ein Präfix voranstellen, um die Region anzugeben, in der sich der Bucket befindet. Ein Beispiel: `us-west-2:DOC-EXAMPLE-BUCKET`. Wenn Sie einen Proxy für Amazon S3-Endpunkte in einer bestimmten Region verwenden, die us-east-1 nicht enthält, stellen Sie dem Bucket-Namen eine Region voran. Wenn dem Bucketname kein Präfix vorangestellt ist, wird die Bucketregion unter Verwendung des Endpunkts us-east-1 automatisch erkannt.

g. Quelle für PS Gallery-Module zulassen: (Optional) Wählen Sie True, um PowerShell Module von https://www.powershellgallery.com/ herunterzuladen. Wenn Sie Falsch wählen, geben Sie eine Quelle für die vorherige Option an ModuleSourceBucketName.

h. Proxy-URI: (Optional) Verwenden Sie diese Option, um MOF-Dateien von einem Proxy-Server herunterzuladen.

i. Reboot Behavior (Neustart-Verhalten): (Optional) Geben Sie eine der folgenden Neustart-Verhaltensweisen an, wenn die Ausführung Ihrer MOF-Datei einen Neustart erfordert:

- AfterMof: Startet den Knoten neu, nachdem alle MOF-Ausführungen abgeschlossen sind. Selbst wenn mehrere MOF-Ausführungen einen Neustart anfordern, wartet das System mit dem Neustart, bis alle MOF-Ausführungen abgeschlossen sind.
- Immediately (Sofort): Startet den Knoten neu, sobald eine MOF-Ausführung dies anfordert. Wenn mehrere MOF-Dateien ausgeführt werden, die einen Neustart anfordern, wird der Knoten mehrmals neu gestartet.
- Never (Nie): Knoten werden selbst dann nicht neu gestartet, wenn die MOF-Ausführung explizit einen Neustart anfordert.

j. Use Computer Name For Reporting (Computername für Berichte verwenden): (Optional) Aktivieren Sie diese Option, um in gemeldeten Compliance-Informationen den Namen des Computers aufzuführen. Der Standardwert ist false (falsch). Das bedeutet, dass das System bei der Meldung von Compliance-Informationen die Knoten-ID verwendet.

k. Turn on Verbose Logging (Verbose-Protokollierung aktivieren): (Optional) Wir empfehlen, die Verbose-Protokollierung zu aktivieren, wenn Sie MOF-Dateien zum ersten Mal bereitstellen.

> **Important**
>
> Wenn diese Option aktiviert ist, schreibt die ausführliche Protokollierung mehr Daten in Ihren Amazon S3-Bucket als die Standardprotokollierung für die Zuordnungsausführung. Dies kann dazu führen, dass die Leistung beeinträchtigt wird und etwas höhere Speichergebühren für Amazon S3 anfallen. Um diese Probleme beim Speichern großer Datenvolumen abzumildern, empfehlen wir, die Lebenszyklusrichtlinien für Ihren Amazon S3-Bucket zu aktivieren. Weitere Informationen finden Sie unter Wie erstelle ich eine Lebenszyklus-Richtlinie für einen S3-Bucket? im Benutzerhandbuch zu Amazon Simple Storage Service.

l. Turn on Debug Logging (Debug-Protokollierung aktivieren): (Optional) Es wird empfohlen, die Debug-Protokollierung zu aktivieren, um MOF-Fehler zu beheben. Wir empfehlen außerdem, diese Option bei normaler Nutzung zu deaktivieren.

> **Important**
>
> Wenn diese Option aktiviert ist, schreibt die Debug-Protokollierung mehr Daten in Ihren Amazon S3-Bucket als die Standardprotokollierung für die Zuordnungsausführung. Dies kann dazu führen, dass die Leistung beeinträchtigt wird und etwas höhere Speichergebühren für Amazon S3 anfallen. Um diese Probleme beim Speichern großer Datenvolumen abzumildern, empfehlen wir, die Lebenszyklusrichtlinien für Ihren Amazon S3-Bucket zu aktivieren. Weitere Informationen finden Sie unter Wie erstelle ich eine Lebenszyklus-Richtlinie für einen S3-Bucket? im Benutzerhandbuch zu Amazon Simple Storage Service.

m. Compliance Type (Compliance-Typ): (Optional) Geben Sie den Compliance-Typ für die Meldung von Compliance-Informationen an. Der Standard-Compliance-Typ lautet Custom:DSC. Wenn Sie mehrere Zuordnungen erstellen, die MOF-Dateien ausführen, müssen Sie für jede Zuordnung einen anderen Compliance-Typ angeben. Wenn Sie dies nicht tun, überschreiben die zusätzlichen Zuordnungen mit Custom:DSC jeweils die bereits zusammengestellten Compliance-Daten.

n. Pre Reboot Script (Skript vor dem Neustart): (Optional) Geben Sie ein Skript an, das ausgeführt werden soll, wenn die Konfiguration einen Neustart anfordert. Das Skript wird vor dem Neustart ausgeführt. Das Skript muss aus einer einzelnen Zeile bestehen. Trennen Sie zusätzliche Zeilen mithilfe von Semikolons.

7. Wählen Sie im Abschnitt Targets (Ziele) entweder Specifying tags (Angaben von Tags) oder Manually Selecting Instance (Manuelles Auswählen einer Instance) aus. Wenn Sie die Zielressourcen mithilfe von Tags ausgewählt haben, geben Sie einen Tag-Schlüssel und den Tag-Wert in die entsprechenden Felder ein. Weitere Informationen zur Verwendung von Zielen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.

8. Wählen Sie im Abschnitt Specify schedule (Zeitplan angeben) entweder On Schedule (Nach Zeitplan) oder No schedule (Kein Zeitplan) aus. Wenn Sie On Schedule (Nach Zeitplan) auswählen, verwenden Sie die verfügbaren Schaltflächen zum Erstellen eines cron- oder rate-Zeitplans für die Zuordnung.

9. Führen Sie im Abschnitt Advanced options (Erweiterte Optionen) Folgendes durch:

- Wählen Sie unter Compliance severity (Compliance -Schweregrad), einen Schweregrad für die Zuordnung aus. In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem Schweregrad, den Sie hier angeben. Weitere Informationen finden Sie unter Informationen zu State Manager-Zuordnungs-Compliance.

10. Konfigurieren Sie im Abschnitt Rate control (Ratensteuerung) Optionen für die Ausführung von State Manager-Zuordnungen in der Flotte der verwalteten Knoten. Weitere Informationen zu diesen Optionen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.

    Wählen Sie im Abschnitt Concurrency (Gleichzeitigkeit) eine Option aus:

    - Wählen Sie Targets (Ziele) aus, um eine absolute Anzahl von Zielen einzugeben, die die Zuordnung gleichzeitig ausführen können.
    - Wählen Sie Percentage (Prozentsatz) aus, um einen Prozentsatz der Ziele anzugeben, die die Zuordnung gleichzeitig ausführen können.

    Wählen Sie im Abschnitt Error threshold (Fehlerschwellenwert) eine Option aus:

    - Wählen Sie Errors (Fehler) aus und geben Sie die absolute Anzahl erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
    - Wählen Sie Percentage (Prozentsatz) aus und geben Sie den Prozentsatz erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.

11. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

    > **Note**
    >
    > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Dienstrolle, die dem verwalteten Knoten

zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

12. Wählen Sie Create Association.

State Manager erstellt die Assoziation und führt sie sofort auf den angegebenen Knoten aus. Nach der ersten Ausführung wird die Zuordnung gemäß dem festgelegten Zeitplan und entsprechend den folgenden Regeln in Intervallen ausgeführt:

- State Manager führt Assoziationen für Knoten aus, die online sind, wenn das Intervall gestartet wird und überspringt Offline-Knoten.
- State Manager versucht, die Assoziation während eines Intervalls auf allen konfigurierten Knoten auszuführen.
- Wenn eine Assoziation in einem Intervall nicht ausgeführt wird (weil beispielsweise die Anzahl der Knoten, die die Assoziation gleichzeitig ausführen können, durch einen Gleichzeitigkeitswert begrenzt wird), versucht State Manager, die Assoziation im nächsten Intervall auszuführen.
- State Manager zeichnet einen Verlauf für alle übersprungenen Datensätze an. Sie können den Verlauf auf der Registerkarte Execution History (Ausführungsverlauf) anzeigen.

> **Note**
>
> Das `AWS-ApplyDSCMofs` ist ein Systems Manager Befehlsdokument. Dies bedeutet, dass dieses Dokument auch mit Run Command, eine Funktion von AWS Systems Manager, ausgeführt werden kann. Weitere Informationen finden Sie unter AWS Systems Manager Run Command.

Fehlerbehebung

Dieser Abschnitt enthält Informationen zur Unterstützung bei der Behebung von Problemen, die möglicherweise beim Erstellen von Zuordnungen zur Ausführung von MOF-Dateien auftreten.

Aktivieren der erweiterten Protokollierung

Aktivieren Sie als ersten Schritt zur Fehlerbehebung die erweiterte Protokollierung. Führen Sie dazu die folgenden Schritte aus:

1. Stellen Sie sicher, dass die Zuordnung so konfiguriert ist, dass sie Befehlsausgaben entweder in Amazon S3 oder Amazon CloudWatch Logs (CloudWatch) schreibt.
2. Setzen Sie den Parameter Enable Verbose Logging auf „True“ fest.
3. Setzen Sie den Parameter Enable Debug Logging auf „True“ fest.

Wenn die Verbose- und die Debug-Protokollierung aktiviert ist, enthält die Stdout-Ausgabedatei Details über die Skriptausführung. Diese Ausgabedatei kann Sie bei der Suche, an welcher Stelle das Skript fehlgeschlagen ist, unterstützen. Die Stderr-Ausgabedatei enthält Fehler, die während der Skriptausführung aufgetreten sind.

Allgemeine Probleme

Dieser Abschnitt enthält Informationen über häufige Probleme, die beim Erstellen von Zuordnungen für die Ausführung von MOF-Dateien auftreten können, sowie Schritte, um diese Probleme zu beheben.

Meine MOF-Datei wurde nicht angewendet.

Wenn State Manager die Assoziation nicht auf Ihre Knoten anwenden konnte, überprüfen Sie zunächst die Stderr-Ausgabedatei. Diese Datei kann Ihnen helfen, die Ursache des Problems zu verstehen. Überprüfen Sie auch Folgendes:

- Der Knoten hat die erforderlichen Zugriffsberechtigungen für alle mit der MOF-Datei verbundenen Amazon S3-Buckets. Das heißt:
  - s3: GetObject Berechtigungen: Dies ist für MOF-Dateien in privaten Amazon S3 S3-Buckets und benutzerdefinierte Module in Amazon S3 S3-Buckets erforderlich.
  - s3: PutObject Berechtigung: Dies ist erforderlich, um Compliance-Berichte und den Compliance-Status in Amazon S3 S3-Buckets zu schreiben.
- Wenn Sie Tags verwenden, stellen Sie sicher, dass der Knoten die erforderliche IAM-Richtlinie hat. Die Verwendung von Tags erfordert, dass die Instance-IAM-Rolle über eine Richtlinie verfügt, die die Aktionen `ec2:DescribeInstances` und `ssm:ListTagsForResource` ermöglicht.
- Stellen Sie sicher, dass dem Knoten die erwarteten Tags oder SSM-Parameter zugewiesen wurden.
- Stellen Sie sicher, dass alle Tags und SSM-Parameter richtig geschrieben sind.
- Versuchen Sie, die MOF-Datei lokal auf dem Knoten auszuführen, um sicherzustellen, dass kein Problem bei der MOF-Datei selbst vorliegt.

Meine MOF-Datei schien fehlzuschlagen, aber die Systems Manager-Ausführung war erfolgreich.

Wenn das Dokument `AWS-ApplyDSCMofs` erfolgreich ausgeführt wurde, wird der Systems Manager-Ausführungsstatus als Success (Erfolg) angezeigt. Dieser Status sagt nichts über den Compliance-Status Ihres Knotens, gemessen an den Konfigurationsanforderungen in der MOF-Datei, aus. Um den Compliance-Status Ihrer Knoten anzuzeigen, zeigen Sie die Compliance-Berichte an. Sie können einen JSON-Bericht im Amazon S3-Bericht-Bucket anzeigen. Dies gilt nur für Run Command- und für State Manager-Ausführungen. Darüber hinaus können Sie Compliance-Details für State Manager auf der Systems Manager-Compliance-Seite anzeigen.

In der Stderr-Ausgabedatei finden sich Hinweise, dass bei dem Versuch, den Service zu erreichen, ein Fehler bei der Namensauflösung aufgetreten ist.

Dieser Fehler weist darauf hin, dass das Skript einen Remoteservice nicht erreichen kann. In den meisten Fällen dürfte das Skript Probleme haben, Amazon S3 zu erreichen. Dieses Problem tritt am häufigsten auf, wenn das Skript versucht, Compliance-Berichte oder den Compliance-Status in den Amazon S3-Bucket zu schreiben, der in den Dokumentparametern angegeben ist. In der Regel tritt dieser Fehler auf, wenn eine Datenverarbeitungsumgebung eine Firewall oder einen transparenten Proxy mit einer Zulassungsliste verwendet. So beheben Sie dieses Problem

- Verwenden Sie die regionsspezifische Bucket-Syntax für alle Amazon S3-Bucket-Parameter. Beispiel: Die Mofs to Apply-Parameter sollten in dem folgenden Format angegeben werden:

  s3:*`bucket-region`*:*`bucket-name`*:*`mof-file-name`*.mof.

  Ein Beispiel: `s3:us-west-2:DOC-EXAMPLE-BUCKET:my-mof.mof`

  Die Bucketnamen für Berichte, Statusinformationen und Modulquellen sollten in dem folgenden Format angegeben werden.

  `bucket-region`:*`bucket-name`*. Ein Beispiel: `us-west-1:DOC-EXAMPLE-BUCKET;`
- Wenn sich das Problem nicht durch Verwendung einer regionsspezifischen Syntax beheben lässt, stellen Sie sicher, dass die Ziel-Knoten in der gewünschten Region auf Simple Storage Service (Amazon S3) zugreifen können. So können Sie dies überprüfen
  1. Suchen Sie den Endpunktnamen für Amazon S3 in der entsprechenden Amazon S3-Region. Weitere Informationen finden Sie unter Amazon-S3-Service-Endpunkte im Allgemeine Amazon Web Services-Referenz.
  2. Melden Sie sich am Ziel-Knoten an und führen Sie den folgenden Ping-Befehl aus.

```
ping s3.s3-region.amazonaws.com
```

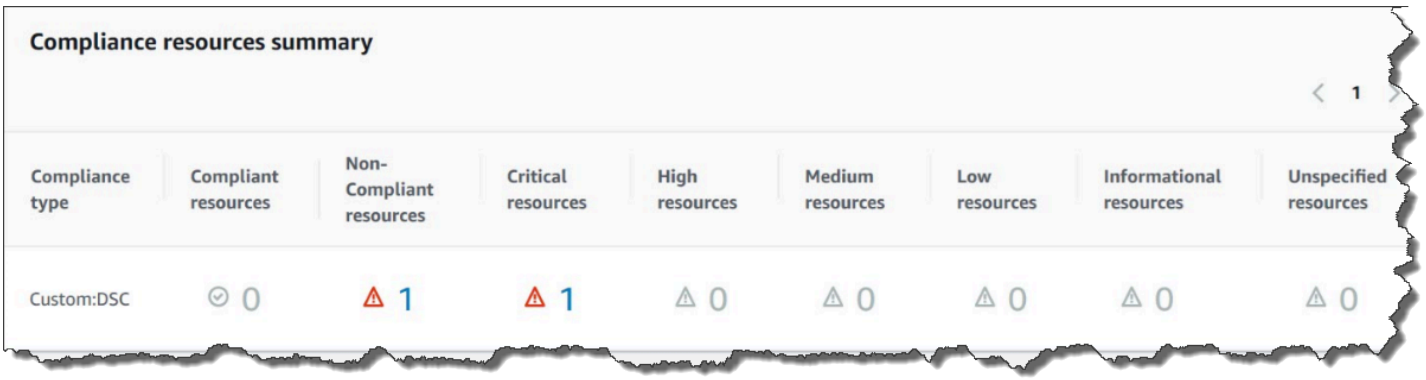
Wenn der Ping-Aufruf fehlgeschlagen ist, bedeutet dies, dass entweder Simple Storage Service (Amazon S3) nicht verfügbar ist, dass eine Firewall bzw. ein transparenter Proxy den Zugriff auf die Simple Storage Service (Amazon S3)-Region blockiert oder dass der Knoten nicht auf das Internet zugreifen kann.

Anzeigen von Details zur DSC-Ressourcen-Compliance

Systems Manager erfasst Compliance-Informationen zu DSC-Ressourcenfehlern im Amazon S3 Status-Bucket, den Sie bei der Ausführung des `AWS-ApplyDSCMofs`-Dokuments angegeben haben. Die Suche nach Informationen zu DSC-Ressourcenfehlern in einem Amazon S3-Bucket kann zeitaufwendig sein. Stattdessen können Sie diese Informationen auf der Systems Manager-Seite Compliance anzeigen.

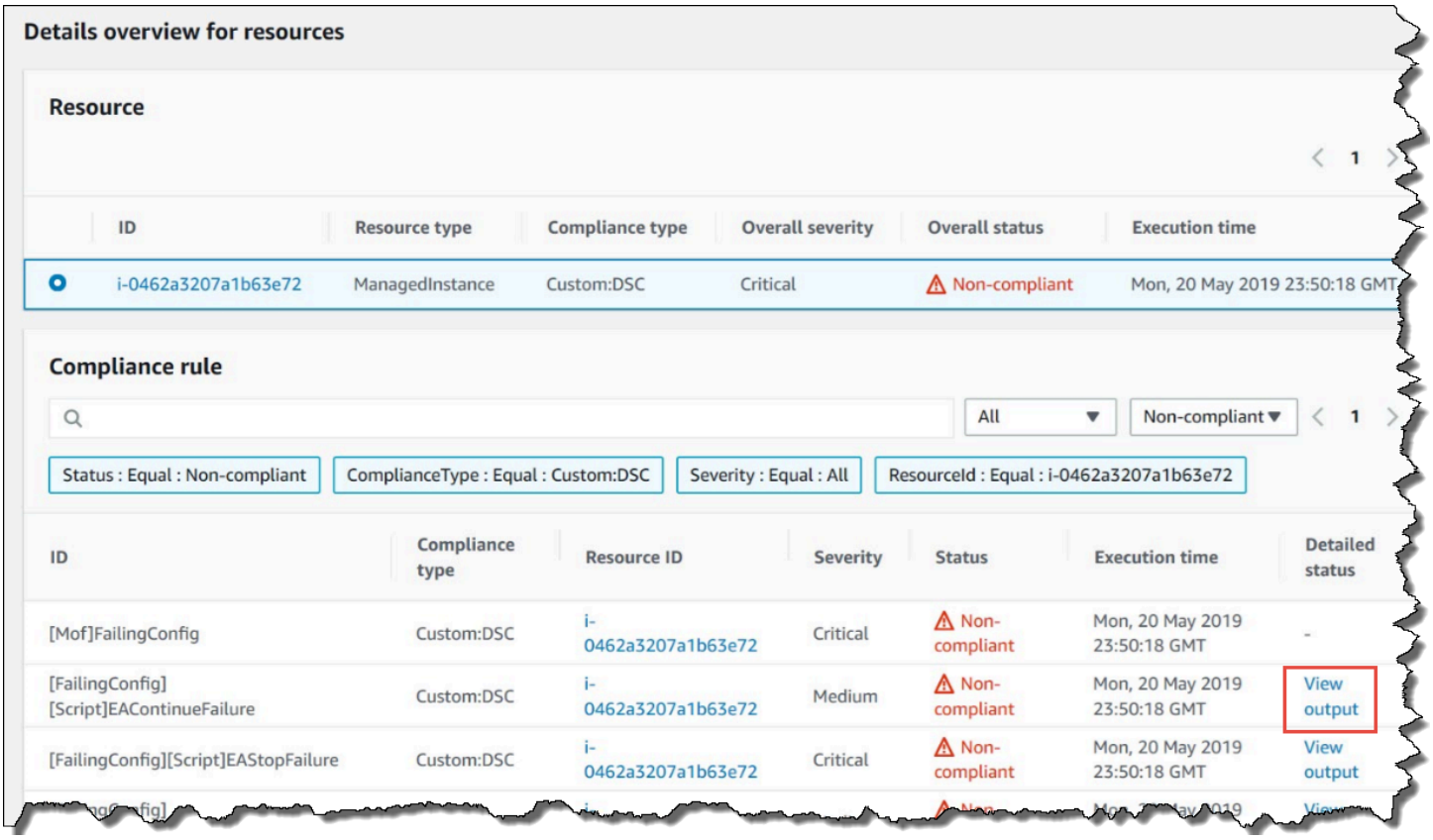
Der Bereich Compliance resources summary (Compliance-Ressourcen-Zusammenfassung) zeigt die Anzahl der Ressourcen an, die fehlgeschlagen sind. Im folgenden Beispiel ist das custom:DSC und eine Ressource ComplianceTypeist nicht konform.

> **Note**
>
> custom:DSC ist der Standardwert im Dokument. ComplianceType`AWS-ApplyDSCMofs` Dieser Wert ist anpassbar.

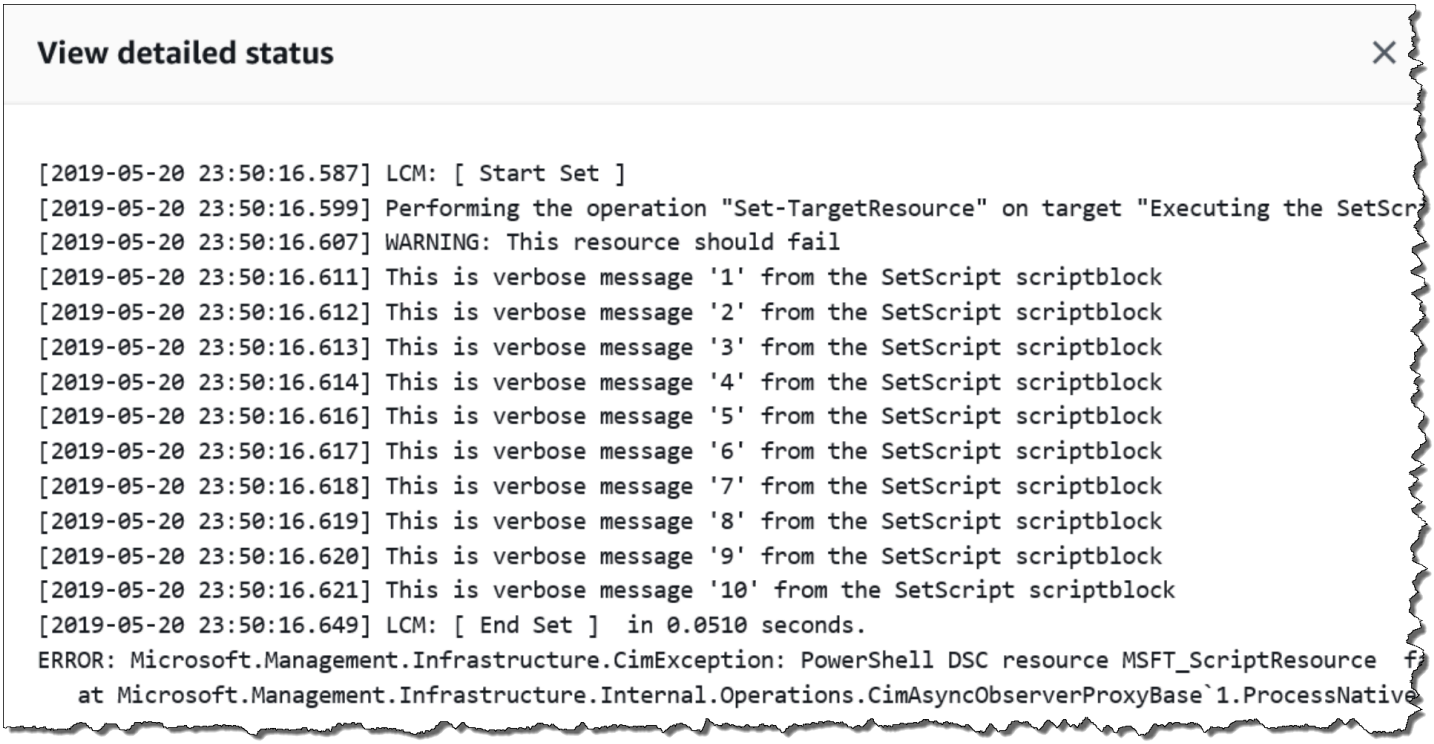


Im Abschnitt „Detailübersicht für Ressourcen“ werden Informationen über die AWS Ressource mit der nicht konformen DSC-Ressource angezeigt. Dieser Bereich enthält auch den MOF-Namen, Skript-

Ausführungsschritte und (falls zutreffend) den Link View output (Ausgabe anzeigen) zur Ansicht detaillierter Statusinformationen.



Der Link View output zeigt die letzten 4.000 Zeichen des detaillierten Status an. Systems Manager beginnt mit der Ausnahme als erstem Element und scannt dann durch die ausführlichen Nachrichten und stellt so viele wie möglich voran, bis das Kontingent von 4.000 Zeichen erreicht wird. Dieser Vorgang zeigt die Protokollmeldungen an, die vor dem Auslösen der Ausnahme ausgegeben wurden. Dabei handelt es sich um die relevantesten Nachrichten für die Fehlerbehebung.



Weitere Informationen zum Anzeigen von Compliance-Informationen finden Sie unter AWS Systems Manager-Compliance.

Situationen, die die Compliance-Berichterstellung beeinflussen

Wenn die State Manager-Zuordnung fehlschlägt, werden keine Compliance-Daten gemeldet. Genauer gesagt, meldet Systems Manager doesn't keine Compliance-Elemente, wenn eine MOF-Datei nicht verarbeitet werden kann, da die Zuordnungen fehlschlagen. Beispiel: Wenn Systems Manager versucht, eine MOF-Datei von einem Amazon S3-Bucket herunterzuladen, und der Knoten keine Berechtigung zum Zugriff besitzt, schlägt die Zuordnung fehl und es werden keine Compliance-Daten gemeldet.

Wenn eine Ressource in einer zweiten MOF-Datei fehlschlägt, dann meldet Systems Manager Bericht-Compliance-Daten. Beispiel: Wenn ein MOF versucht, eine Datei auf einem nicht vorhandenen Laufwerk zu erstellen, dann meldet Systems Manager Compliance, da das `AWS-ApplyDSCMofs`-Dokument vollständig verarbeitet werden kann. Dies bedeutet, dass die Zuordnung erfolgreich ausgeführt wird.

Exemplarische Vorgehensweise: Erstellen von Verknüpfungen, die Playbooks ausführen Ansible

Mithilfe des SSM-Dokuments können Sie State Manager Verknüpfungen erstellen, die Ansible `AWS-ApplyAnsiblePlaybooks` Playbooks ausführen. State Managerist eine Fähigkeit von. AWS Systems Manager Dieses Dokument bietet die folgenden Vorteile für die Ausführung von Playbooks:

- Unterstützung für die Ausführung komplexer Playbooks
- Support für das Herunterladen von Playbooks von GitHub und Amazon Simple Storage Service (Amazon S3)
- Unterstützung der komprimierten Playbook-Struktur
- Erweiterte Protokollierung
- Möglichkeit, anzugeben, welches Playbook ausgeführt werden soll, wenn Playbooks gebündelt werden

> **Note**
>
> Systems Manager enthält zwei SSM-Dokumente, mit denen Sie State Manager Verknüpfungen erstellen können, die Ansible Playbooks ausführen: `AWS-RunAnsiblePlaybook` und. `AWS-ApplyAnsiblePlaybooks` Das `AWS-RunAnsiblePlaybook`-Dokument ist veraltet. Es bleibt für Legacy-Zwecke in Systems Manager verfügbar. Wir empfehlen, dass Sie das `AWS-ApplyAnsiblePlaybooks`-Dokument aufgrund der hier beschriebenen Verbesserungen verwenden. Verknüpfungen, die Ansible Playbooks ausführen, werden auf nicht unterstützt. macOS

Unterstützung für die Ausführung komplexer Playbooks

Das `AWS-ApplyAnsiblePlaybooks`-Dokument unterstützt gebündelte, komplexe Playbooks, da es die gesamte Dateistruktur vor der Ausführung des angegebenen Haupt-Playbooks in ein lokales Verzeichnis kopiert. Sie können Quell-Playbooks in Zip-Dateien oder in einer Verzeichnisstruktur bereitstellen. Die Zip-Datei oder das Verzeichnis kann in GitHub Amazon S3 gespeichert werden.

Unterstützung für das Herunterladen von Playbooks von GitHub

Das `AWS-ApplyAnsiblePlaybooks`-Dokument verwendet das `aws:downloadContent`-Plug-in zum Herunterladen von Playbook-Dateien. Dateien können GitHub in einer einzelnen Datei oder als

kombinierter Satz von Playbook-Dateien gespeichert werden. Um Inhalte herunterzuladenGitHub, geben Sie Informationen zu Ihrem GitHub Repository im JSON-Format an. Ein Beispiel.

```
{
   "owner":"TestUser",
   "repository":"GitHubTest",
   "path":"scripts/python/test-script",
   "getOptions":"branch:master",
   "tokenInfo":"{{ssm-secure:secure-string-token}}"
}
```

Unterstützung für das Herunterladen von Playbooks von Amazon S3

Sie können Ansible Playbooks auch in Amazon S3 entweder als einzelne ZIP-Datei oder als Verzeichnisstruktur speichern und herunterladen. Geben Sie den Pfad zur Datei an, um Inhalte von Amazon S3 herunterzuladen. Nachfolgend finden Sie zwei Beispiele.

Beispiel 1: Herunterladen einer bestimmten Playbook-Datei

```
{
   "path":"https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/playbook.yml"
}
```

Beispiel 2: Herunterladen des Inhalts eines Verzeichnisses

```
{
   "path":"https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/ansible/webservers/"
}
```

> **Important**
>
> Wenn Sie Amazon S3 angeben, muss das Instance-Profil AWS Identity and Access Management (IAM) auf Ihren verwalteten Knoten mit der `AmazonS3ReadOnlyAccess` Richtlinie konfiguriert werden. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

Unterstützung der komprimierten Playbook-Struktur

Mit dem AWS-ApplyAnsiblePlaybooks-Dokument können Sie komprimierte ZIP-Dateien im heruntergeladenen Paket ausführen. Das Dokument prüft, ob die heruntergeladenen Dateien eine komprimierte Datei im ZIP-Format enthalten. Wenn eine ZIP-Datei gefunden wird, dekomprimiert das Dokument die Datei automatisch und führt dann die angegebene Automatisierung aus. Ansible

Erweiterte Protokollierung

Das AWS-ApplyAnsiblePlaybooks-Dokument enthält einen optionalen Parameter für die Angabe verschiedener Protokollierungsebenen. Geben Sie -v für niedrige Ausführlichkeit, -vv oder -vvv für mittlere Ausführlichkeit und -vvvv für die Protokollierung auf Debug-Ebene an. Diese Optionen sind direkt den Ausführlichkeitsoptionen zugeordnetAnsible.

Möglichkeit, anzugeben, welches Playbook ausgeführt werden soll, wenn Playbooks gebündelt werden

Das AWS-ApplyAnsiblePlaybooks-Dokument enthält einen erforderlichen Parameter, um anzugeben, welches Playbook ausgeführt werden soll, wenn mehrere Playbooks gebündelt werden. Diese Option bietet Flexibilität für die Ausführung von Playbooks, um verschiedene Anwendungsfälle zu unterstützen.

Installierte Abhängigkeiten

Wenn Sie True für den InstallDependenciesParameter angeben, überprüft Systems Manager, ob auf Ihren Knoten die folgenden Abhängigkeiten installiert sind:

- Ubuntu Server/Debian Server: apt-Get (Paketverwaltung), Python 3, Ansible Entpacken
- Amazon Linux: Ansible
- RHEL: Python 3Ansible, Entpacken

Wenn eine oder mehrere dieser Abhängigkeiten nicht gefunden werden, installiert Systems Manager sie automatisch.

Erstellen Sie eine Assoziation, die Ansible Playbooks ausführt (Konsole)

Das folgende Verfahren beschreibt, wie Sie mit der Systems Manager Manager-Konsole eine State Manager Assoziation erstellen, die Ansible Playbooks mithilfe des AWS-ApplyAnsiblePlaybooks Dokuments ausführt.

So erstellen Sie eine Assoziation, die Ansible Playbooks ausführt (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie State Manager und dann Create association (Zuordnung wählen) aus.
4. Geben Sie unter Name einen Namen an, der Ihnen hilft, sich an den Zweck der Zuordnung zu erinnern.
5. Wählen Sie in der Liste Dokument die Option **`AWS-ApplyAnsiblePlaybooks`** aus.
6. Wählen Sie im Abschnitt Parameter für Source Type entweder GitHuboder S3 aus.

   GitHub

   Wenn Sie möchten GitHub, geben Sie die Repository-Informationen im folgenden Format ein.

   ```
   {
      "owner":"user_name",
      "repository":"name",
      "path":"path_to_directory_or_playbook_to_download",
      "getOptions":"branch:branch_name",
      "tokenInfo":"{{(Optional)_token_information}}"
   }
   ```

   S3

   Wenn Sie S3 auswählen, geben Sie Pfadinformationen im folgenden Format ein.

   ```
   {
      "path":"https://s3.amazonaws.com/path_to_directory_or_playbook_to_download"
   }
   ```
7. Wählen Sie unter Install Dependencies (Abhängigkeiten installieren) eine Option aus.
8. (Optional) Geben Sie unter Playbook File (Playbook-Datei) einen Dateinamen ein. Wenn eine Zip-Datei das Playbook enthält, geben Sie einen relativen Pfad zur Zip-Datei an.
9. (Optional) Geben Sie unter Zusätzliche Variablen Variablen ein, an die Sie zur Ansible Laufzeit senden möchtenState Manager.
10. (Optional) Wählen Sie unter Check (Prüfen) eine Option aus.
11. (Optional) Wählen Sie für Verbose eine Option aus.

12. Wählen Sie für Targets (Ziele) eine Option aus. Weitere Informationen zur Verwendung von Zielen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.
13. Wählen Sie im Abschnitt Specify schedule (Zeitplan angeben) entweder On schedule (Nach Zeitplan) oder No schedule (Kein Zeitplan) aus. Wenn Sie On schedule (Nach Zeitplan) auswählen, verwenden Sie die verfügbaren Schaltflächen zum Erstellen eines cron- oder rate-Zeitplans für die Zuordnung.
14. Wählen Sie im Abschnitt Advanced options (Erweiterte Optionen) für Compliance severity (Compliance-Schweregrad) einen Schweregrad für die Zuordnung aus. In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem Schweregrad, den Sie hier angeben. Weitere Informationen finden Sie unter Informationen zu State Manager-Zuordnungs-Compliance.
15. Konfigurieren Sie im Abschnitt Rate control (Ratensteuerung) Optionen für die Ausführung von State Manager-Zuordnungen in der Flotte von verwalteten Knoten. Weitere Informationen über Ratensteuerungen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.

    Wählen Sie im Abschnitt Concurrency (Gleichzeitigkeit) eine Option aus:

    - Wählen Sie Targets (Ziele) aus, um eine absolute Anzahl von Zielen einzugeben, die die Zuordnung gleichzeitig ausführen können.
    - Wählen Sie Percentage (Prozentsatz) aus, um einen Prozentsatz der Ziele anzugeben, die die Zuordnung gleichzeitig ausführen können.

    Wählen Sie im Abschnitt Error threshold (Fehlerschwellenwert) eine Option aus:

    - Wählen Sie Errors (Fehler) aus und geben Sie die absolute Anzahl erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
    - Wählen Sie Percentage (Prozentsatz) aus und geben Sie den Prozentsatz erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
16. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

17. Wählen Sie Create Association.

> **Note**
>
> Wenn Sie auf einer oder mehreren Knoten eine Assoziation anhand von Tags erstellen und von einem dieser Knoten die Tags entfernen, wird die Assoziation auf diesem Knoten nicht mehr ausgeführt. Die Assoziazion zwischen dem Knoten und dem State Manager-Dokument ist aufgehoben.

Erstellen Sie eine Assoziation, die Ansible Playbooks (CLI) ausführt

Das folgende Verfahren beschreibt, wie Sie mit AWS Command Line Interface (AWS CLI) mithilfe des Dokuments eine State Manager Assoziation erstellen, die Ansible `AWS-ApplyAnsiblePlaybooks` Playbooks ausführt.

So erstellen Sie eine Assoziation, die Ansible Playbooks (CLI) ausführt

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie einen der folgenden Befehle aus, um eine Assoziation zu erstellen, die Ansible Playbooks ausführt, indem sie mithilfe von Tags auf Knoten abzielt. Ersetzen Sie jeden

Beispiel Platzhalter für Ressourcen mit Ihren eigenen Informationen. Befehl (A) gibt GitHub als Quelltyp an. Befehl (B) gibt Amazon S3 als Quelltyp an.

(A) GitHub Quelle

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"
```

Ein Beispiel.

```
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) S3-Quelle

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"
```

Ein Beispiel.

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" \
    --targets "Key=tag:OS,Values=Linux" \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/DOC-EXAMPLE-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**
>
> State Manager-Zuordnungen unterstützen nicht alle Cron- und Rate-Ausdrücke. Weitere Informationen zum Erstellen von Cron- und Rate-Ausdrücken für Zuordnungen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Das System versucht, die Assoziation auf den Knoten zu erstellen und den Status sofort anzuwenden.

3. Führen Sie den folgenden Befehl aus, um einen aktualisierten Status der soeben erstellten Zuordnung anzuzeigen.

```
aws ssm describe-association --association-id "ID"
```

Exemplarische Vorgehensweise: Erstellen von Verknüpfungen, die Rezepte ausführen Chef

Mithilfe des `AWS-ApplyChefRecipes` SSM-Dokuments können Sie State Manager Verknüpfungen erstellen, die Chef Rezepte ausführen. State Managerist eine Fähigkeit von. AWS Systems Manager Sie können mit dem `AWS-ApplyChefRecipes` SSM-Dokument eine Ausrichtung auf Linux-basierte verwaltete Systems Manager-Knoten verwenden. Dieses Dokument bietet die folgenden Vorteile beim Ausführen von Chef Rezepten:

- Unterstützt mehrere Versionen von Chef (Chef11 bis Chef 18).
- Installiert die Chef Client-Software automatisch auf den Zielknoten.
- Führt optional Systems Manager-Compliance-Prüfungen für Ziel-Knoten aus und speichert die Ergebnisse der Compliance-Prüfungen in einem Amazon Simple Storage Service (Amazon S3)-Bucket.
- Führt mehrere Cookbooks und Rezepte in einem einzigen Durchlauf des Dokuments aus.
- Führt optional Rezepte im `why-run`-Modus aus, um anzuzeigen, welche Rezepte sich auf Ziel-Knoten ändern, ohne Änderungen vorzunehmen.
- Wendet optional benutzerdefinierte JSON-Attribute auf `chef-client`-Durchläufe an.
- Wendet optional benutzerdefinierte JSON-Attribute aus einer Quelldatei an, die an einem von Ihnen angegebenen Ort gespeichert ist.

Sie können Git - GitHub, HTTP - oder Amazon S3 S3-Buckets als Download-Quellen für Chef Kochbücher und Rezepte verwenden, die Sie in einem `AWS-ApplyChefRecipes` Dokument angeben.

> **Note**
>
> Verknüpfungen, die Chef Rezepte ausführen, werden auf nicht unterstützt. macOS

Voraussetzungen: Einrichtung von Zuordnung, Repository und Cookbooks

Bevor Sie ein `AWS-ApplyChefRecipes` Dokument erstellen, bereiten Sie Ihre Chef Kochbücher und Ihr Kochbuch-Repository vor. Wenn Sie noch kein Chef Kochbuch haben, das Sie verwenden möchten, können Sie zunächst ein `HelloWorld` Testkochbuch verwenden, das für Sie vorbereitet AWS wurde. Das `AWS-ApplyChefRecipes`-Dokument verweist bereits standardmäßig auf dieses Cookbook. Ihre Cookbooks sollten ähnlich wie die folgende Verzeichnisstruktur eingerichtet werden. Im folgenden Beispiel finden Sie Beispiele für Chef Kochbücher, die Chef Supermarketauf der Website verfügbar sind. `jenkins` `nginx` Chef

Kochbücher auf der Chef SupermarketWebsite AWS können zwar nicht offiziell unterstützt werden, aber viele von ihnen funktionieren mit dem `AWS-ApplyChefRecipes` Dokument. Im Folgenden finden Sie Beispiele für Kriterien, die Sie bestimmen müssen, wenn Sie ein Community-Cookbook testen:

- Das Cookbook sollte die Linux-basierten Betriebssysteme der Systems Manager-verwalteten Knoten unterstützen, auf die Sie zielen.
- Das Kochbuch sollte für die von Ihnen verwendete Chef Client-Version (Chef11 bis Chef 18) gültig sein.
- Das Kochbuch ist mit einem Chef Infra Client Chef-Server kompatibel und benötigt keinen.

Stellen Sie sicher, dass Sie die `Chef.io` Website erreichen können, sodass alle Kochbücher, die Sie in Ihrer Ausführungsliste angeben, installiert werden können, wenn das Systems Manager Manager-Dokument (SSM-Dokument) ausgeführt wird. Die Verwendung eines eingebetteten `cookbooks`-Ordners wird zwar unterstützt, ist aber nicht erforderlich. Sie können Cookbooks direkt unter der Root-Ebene speichern.

```
<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**
>
> Bevor Sie eine State Manager Zuordnung erstellen, die Chef Rezepte ausführt, sollten Sie sich bewusst sein, dass beim Ausführen des Dokuments die Chef Clientsoftware auf Ihren von Systems Manager verwalteten Knoten installiert wird, es sei denn, Sie setzen den Wert der ChefClientversion auf`None`. Bei diesem Vorgang wird ein Installationsskript von verwendetChef, um Chef Komponenten in Ihrem Namen zu installieren. Bevor Sie ein `AWS-ApplyChefRecipes` Dokument ausführen, stellen Sie sicher, dass Ihr Unternehmen alle geltenden gesetzlichen Anforderungen, einschließlich der Lizenzbedingungen für die Verwendung von Chef Software, erfüllt. Weitere Informationen finden Sie Chefauf der Website.

Systems Manager kann Compliance-Berichte an einen S3-Bucket oder die Systems Manager-Konsole übermitteln oder Compliance-Ergebnisse als Antwort auf Systems Manager-API-Befehle zur Verfügung stellen. Zum Ausführen von Systems Manager-Compliance-Berichten muss das Instance-Profil, das an Systems Manager-verwaltete Knoten angefügt ist, über Berechtigungen zum Schreiben in den S3-Bucket verfügen. Das Instance-Profil muss über die Berechtigung zur Nutzung der Systems Manager `PutComplianceItem`-API verfügen. Weitere Informationen zur Systems Manager-Compliance finden Sie unter AWS Systems Manager-Compliance.

Protokollieren der Dokumentausführung

Wenn Sie ein Systems Manager Manager-Dokument (SSM-Dokument) mithilfe einer State Manager Zuordnung ausführen, können Sie die Zuordnung so konfigurieren, dass die Ausgabe des Dokumentenlaufs ausgewählt wird, und Sie können die Ausgabe an Amazon S3 oder Amazon CloudWatch Logs (CloudWatch Logs) senden. Um die Problembehebung zu vereinfachen, wenn die Ausführung einer Zuordnung abgeschlossen ist, stellen Sie sicher, dass die Zuordnung so konfiguriert ist, dass die Befehlsausgabe entweder in einen Amazon S3 S3-Bucket oder in CloudWatch Logs geschrieben wird. Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

Anwenden von JSON-Attributen auf Ziele bei der Ausführung eines Rezepts

Sie können JSON-Attribute für Ihren Chef Client angeben, die während eines Zuordnungslaufs auf Zielknoten angewendet werden sollen. Beim Einrichten der Zuordnung können Sie unformatiertes JSON oder den Pfad zu einer in Amazon S3 gespeicherten JSON-Datei angeben.

Verwenden Sie JSON-Attribute, wenn Sie beispielsweise die Art und Weise, wie das Rezept ausgeführt wird, anpassen möchten, ohne das Rezept selbst ändern zu müssen:

- Überschreiben einer kleinen Anzahl von Attributen

  Verwenden Sie benutzerdefiniertes JSON, um zu vermeiden, dass Sie mehrere Versionen eines Rezepts verwalten müssen, um kleine Unterschiede zu berücksichtigen.
- Bereitstellung variabler Werte

  Verwenden Sie benutzerdefiniertes JSON, um Werte anzugeben, die sich von ändern können run-to-run. Wenn Ihre Chef Kochbücher beispielsweise eine Drittanbieteranwendung konfigurieren, die Zahlungen akzeptiert, können Sie benutzerdefiniertes JSON verwenden, um die URL des Zahlungsendpunkts anzugeben.

Angeben von Attributen in unformatiertem JSON

Im Folgenden finden Sie ein Beispiel für das Format, das Sie verwenden können, um benutzerdefinierte JSON-Attribute für Ihr Chef Rezept anzugeben.

```
{"filepath":"/tmp/example.txt", "content":"Hello, World!"}
```

Angabe eines Pfads zu einer JSON-Datei

Im Folgenden finden Sie ein Beispiel für das Format, mit dem Sie den Pfad zu benutzerdefinierten JSON-Attributen für Ihr Chef Rezept angeben können.

```
{"sourceType":"s3", "sourceInfo":"someS3URL1"}, {"sourceType":"s3",
 "sourceInfo":"someS3URL2"}
```

Git als Quelle für Cookbooks verwenden

Das `AWS-ApplyChefRecipes` Dokument verwendet das `aws:downloadContent`Plugin zum Herunterladen von Chef Kochbüchern. Um Inhalte aus Git herunterzuladen, geben Sie Informationen

über Ihr Git-Repository im JSON-Format an, wie im folgenden Beispiel. Ersetzen Sie jeden *example-resource-placeholder* durch Ihre eigenen Informationen.

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

Verwenden von GitHub als Quelle für Cookbooks

Das `AWS-ApplyChefRecipes`-Dokument verwendet das `aws:downloadContent`-Plugin, um Cookbooks herunterzuladen. Um Inhalte herunterzuladenGitHub, geben Sie wie im folgenden Beispiel Informationen zu Ihrem GitHub Repository im JSON-Format an. Ersetzen Sie jeden *example-resource-placeholder* durch Ihre eigenen Informationen.

```
{
   "owner":"TestUser",
   "repository":"GitHubCookbookRepository",
   "path":"cookbooks/HelloWorld",
   "getOptions":"branch:refs/head/main",
   "tokenInfo":"{{ssm-secure:token-secure-string-parameter}}"
}
```

HTTP als Quelle für Cookbooks verwenden

Sie können Chef Kochbücher an einem benutzerdefinierten HTTP-Speicherort entweder als einzelne `tar.gz` Datei `.zip` oder als Verzeichnisstruktur speichern. Um Inhalte über HTTP herunterzuladen, geben Sie den Pfad zu der Datei oder dem Verzeichnis im JSON-Format wie im folgenden Beispiel an. Ersetzen Sie jeden *example-resource-placeholder* durch Ihre eigenen Informationen.

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

Verwenden von Amazon S3 als Quelle für Cookbooks

Sie können Chef Kochbücher auch in Amazon S3 entweder als einzelne `tar.gz` Datei `.zip` oder als Verzeichnisstruktur speichern und herunterladen. Um Inhalte von Amazon S3 herunterzuladen, geben Sie den Pfad zu der Datei im JSON-Format wie in den folgenden Beispielen an. Ersetzen Sie jeden *`example-resource-placeholder`* durch Ihre eigenen Informationen.

Beispiel 1: Herunterladen eines bestimmten Cookbooks

```
{
   "path":"https://s3.amazonaws.com/chef-cookbooks/HelloWorld.zip"
}
```

Beispiel 2: Herunterladen des Inhalts eines Verzeichnisses

```
{
   "path":"https://s3.amazonaws.com/chef-cookbooks-test/HelloWorld"
}
```

> **Important**
>
> Wenn Sie Amazon S3 angeben, muss das Instance-Profil AWS Identity and Access Management (IAM) auf Ihren verwalteten Knoten mit der `AmazonS3ReadOnlyAccess` Richtlinie konfiguriert werden. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

Themen

- Erstellen Sie eine Assoziation, die Chef Rezepte ausführt (Konsole)
- Erstellen Sie eine Assoziation, die Chef Rezepte ausführt (CLI)
- Anzeigen von Details zur Chef-Ressourcen-Compliance

Erstellen Sie eine Assoziation, die Chef Rezepte ausführt (Konsole)

Das folgende Verfahren beschreibt, wie Sie mit der Systems Manager Manager-Konsole eine State Manager Assoziation erstellen, die Chef Kochbücher mithilfe des `AWS-ApplyChefRecipes` Dokuments ausführt.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie State Manager und dann Create association (Zuordnung wählen) aus.
4. Geben Sie unter Name einen Namen ein, der Ihnen hilft, sich an den Zweck der Zuordnung zu erinnern.
5. Wählen Sie in der Liste Dokument die Option **`AWS-ApplyChefRecipes`** aus.
6. Wählen Sie unter Parameter für Quelltyp entweder Git GitHub, HTTP oder S3 aus.
7. Geben Sie unter Quelleninfo die Informationen zur Cookbook-Quelle in dem Format ein, das dem in Schritt 6 ausgewählten Quellentyp entspricht. Weitere Informationen finden Sie unter den folgenden Themen:

   - the section called “Git als Quelle für Cookbooks verwenden”
   - the section called “Verwenden von GitHub als Quelle für Cookbooks”
   - the section called “HTTP als Quelle für Cookbooks verwenden”
   - the section called “Verwenden von Amazon S3 als Quelle für Cookbooks”
8. Listen Sie in der Run list (Ausführungsliste) die auszuführenden Rezepte im folgenden Format auf. Trennen Sie jedes Rezept durch ein Komma wie gezeigt. Geben Sie kein Leerzeichen nach dem Komma ein. Ersetzen Sie jeden *`example-resource-placeholder`* durch Ihre eigenen Informationen.

   ```
   recipe[cookbook-name1::recipe-name],recipe[cookbook-name2::recipe-name]
   ```
9. (Optional) Geben Sie benutzerdefinierte JSON-Attribute an, die der Chef Client an Ihre Zielknoten weitergeben soll.

   a. Fügen Sie im Inhalt der JSON-Attribute alle Attribute hinzu, die der Chef Client an Ihre Zielknoten weitergeben soll.

   b. Fügen Sie in JSON-Attributquellen die Pfade zu allen Attributen hinzu, die der Chef Client an Ihre Zielknoten weitergeben soll.

   Weitere Informationen finden Sie unter the section called “Anwenden von JSON-Attributen auf Ziele bei der Ausführung eines Rezepts”.
10. Geben Sie für die ChefClient-Version eine Chef Version an. Gültige Werte sind 11 bis 18 oder None. Wenn Sie eine Zahl zwischen 11 18 (einschließlich) angeben, installiert Systems Manager

die richtige Chef Client-Version auf Ihren Zielknoten. Wenn Sie angeben`None`, installiert Systems Manager den Chef Client nicht auf den Zielknoten, bevor die Rezepte des Dokuments ausgeführt werden.

11. (Optional) Geben Sie für ChefClient-Argumente zusätzliche Argumente an, die für die Version von, die Chef Sie verwenden, unterstützt werden. Um mehr über unterstützte Argumente zu erfahren, führen Sie die Ausführung `chef-client -h` auf einem Knoten aus, auf dem der Chef Client ausgeführt wird.
12. (Optional) Aktivieren Sie Why-run, um Änderungen anzuzeigen, die bei der Ausführung der Rezepte an Ziel-Knoten vorgenommen wurden, ohne dass die Ziel-Knoten tatsächlich geändert werden.
13. Wählen Sie für Compliance severity (Schweregrad der Compliance) den Schweregrad der Systems Manager-Compliance-Ergebnisse aus, die gemeldet werden sollen. In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem festgelegten Schweregrad. Compliance-Berichte werden in einem S3-Bucket gespeichert, den Sie als Wert des Parameters Compliance report bucket (Compliance-Berichts-Bucket) angeben (Schritt 14). Weitere Informationen zur Compliance finden Sie unter Arbeiten mit Compliance in dieser Anleitung.

    Bei Konformitätsscans wird die Abweichung zwischen der Konfiguration, die in Ihren Chef Rezepten angegeben ist, und den Knotenressourcen gemessen. Gültige Werte sind `Critical`, `High`, `Medium`, `Low`, `Informational`, `Unspecified` oder `None`. Um die Compliance-Berichterstattung zu überspringen, wählen Sie `None`.
14. Geben Sie unter Compliance type (Compliance-Typ) den Compliance-Typ an, für den die Ergebnisse gemeldet werden sollen. Gültige Werte sind `Association` für State Manager-Zuordnungen oder `Custom:`*`custom-type`*. Der Standardwert ist `Custom:Chef`.
15. Geben Sie für den Compliance-Berichts-Bucket den Namen eines S3-Buckets ein, in dem Informationen zu jedem Chef Durchlauf gespeichert werden sollen, der von diesem Dokument ausgeführt wird, einschließlich der Ressourcenkonfiguration und der Konformitätsergebnisse.
16. Konfigurieren Sie in Rate control (Ratensteuerung) Optionen für die Ausführung von State Manager-Zuordnungen in der Flotte von verwalteten Knoten. Weitere Informationen über Ratensteuerungen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.

    Wählen Sie unter Concurrency (Gleichzeitigkeit) eine Option aus:

- Wählen Sie Targets (Ziele) aus, um eine absolute Anzahl von Zielen einzugeben, die die Zuordnung gleichzeitig ausführen können.
- Wählen Sie Percentage (Prozentsatz) aus, um einen Prozentsatz der Ziele anzugeben, die die Zuordnung gleichzeitig ausführen können.

Wählen Sie unter Error threshold (Fehlerschwelle) eine Option aus:

- Wählen Sie Errors (Fehler) aus und geben Sie die absolute Anzahl erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
- Wählen Sie Percentage (Prozentsatz) aus und geben Sie den Prozentsatz erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.

17. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

18. Wählen Sie Create Association.

Erstellen Sie eine Assoziation, die Chef Rezepte ausführt (CLI)

Das folgende Verfahren beschreibt, wie Sie mit AWS Command Line Interface (AWS CLI) mithilfe des `AWS-ApplyChefRecipes` Dokuments eine State Manager Assoziation erstellen, die Chef-Kochbücher ausführt.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie einen der folgenden Befehle aus, um eine Assoziation zu erstellen, die Chef Kochbücher auf Zielknoten mit den angegebenen Tags ausführt. Verwenden Sie den Befehl, der für Ihren Quellentyp des Cookbooks und Ihr Betriebssystem geeignet ist. Ersetzen Sie jeden *example-resource-placeholder* durch Ihre eigenen Informationen.

   a. Git-Quelle

      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. GitHub Quelle

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"
```

Ein Beispiel.

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. HTTP-Quelle

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. Amazon-S3-Quelle

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": ["DOC-EXAMPLE-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": ["DOC-EXAMPLE-BUCKET"]}' ^
    --association-name "name" ^
    --schedule-expression "cron_or_rate_expression"
```

Ein Beispiel.

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets "Key=tag:OS,Values= Linux" \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path
\":\"https://s3.amazonaws.com/DOC-EXAMPLE-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/DOC-EXAMPLE-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 *)"
```

Das System erstellt die Zuordnung und führt die Zuordnung auf den Zielknoten aus, es sei denn, Ihr angegebener cron- oder rate-Ausdruck verhindert dies.

> **Note**
>
> State Manager-Zuordnungen unterstützen nicht alle Cron- und Rate-Ausdrücke. Weitere Informationen zum Erstellen von Cron- und Rate-Ausdrücken für Zuordnungen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

3. Führen Sie den folgenden Befehl aus, um den Status der Zuordnung, die Sie gerade erstellt haben, anzuzeigen.

```
aws ssm describe-association --association-id "ID"
```

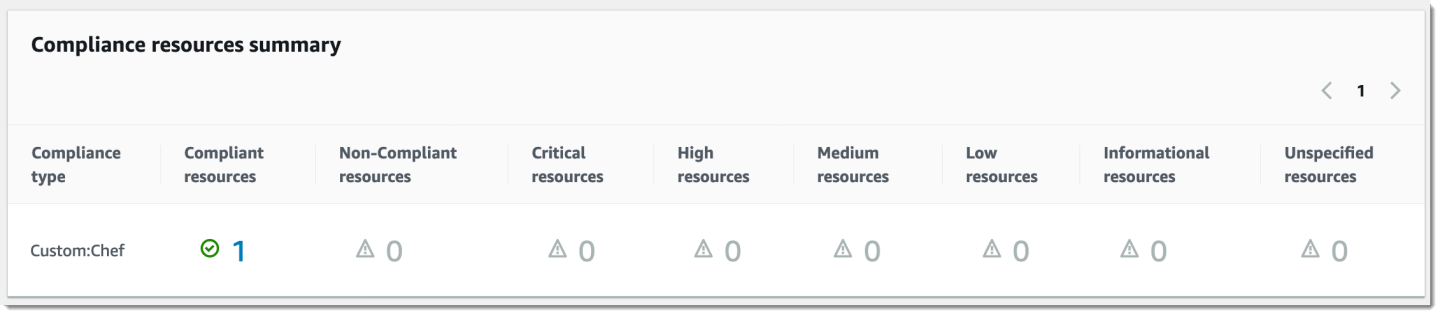
Anzeigen von Details zur Chef-Ressourcen-Compliance

Systems Manager erfasst Compliance-Informationen über Chef verwaltete Ressourcen im Bucketwert des Amazon S3 S3-Compliance-Berichts, den Sie bei der Ausführung des `AWS-ApplyChefRecipes` Dokuments angegeben haben. Die Suche nach Informationen über Chef Ressourcenausfälle in einem S3-Bucket kann zeitaufwändig sein. Stattdessen können Sie diese Informationen auf der Systems Manager-Seite Compliance anzeigen.

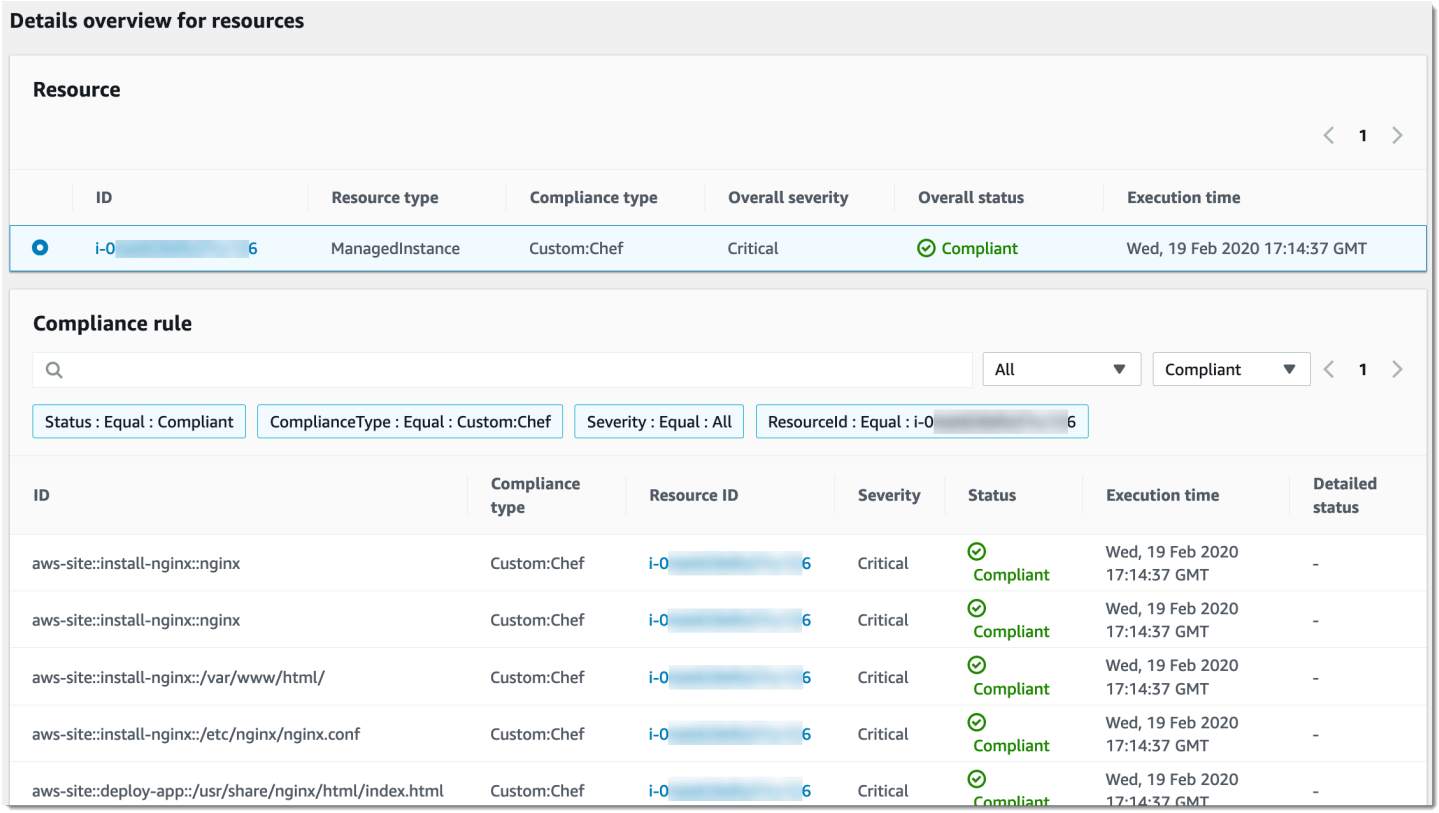
Ein Systems Manager Manager-Konformitätsscan sammelt Informationen über Ressourcen auf Ihren verwalteten Knoten, die bei der letzten Chef Ausführung erstellt oder überprüft wurden. Die Ressourcen können unter anderem Dateien, Verzeichnisse, `systemd`-Services, `yum`-Pakete, Vorlagendateien, `gem`-Pakete und abhängige Cookbooks umfassen.

Der Bereich Compliance resources summary (Compliance-Ressourcen-Zusammenfassung) zeigt die Anzahl der Ressourcen an, die fehlgeschlagen sind. Im folgenden Beispiel ComplianceTypelautet Custom: Chef und eine Ressource ist nicht konform.

> **Note**
>
> `Custom:Chef`ist der ComplianceTypeStandardwert im `AWS-ApplyChefRecipes` Dokument. Dieser Wert ist anpassbar.



Der Abschnitt „Detailübersicht für Ressourcen“ enthält Informationen über die AWS Ressource, die nicht richtlinientreu ist. Dieser Abschnitt enthält auch den Chef Ressourcentyp, für den die Konformität ausgeführt wurde, den Schweregrad des Problems, den Konformitätsstatus und gegebenenfalls Links zu weiteren Informationen.



View output zeigt die letzten 4.000 Zeichen des detaillierten Status an. Systems Manager beginnt mit der Ausnahme als erstem Element, sucht nach ausführlichen Meldungen und zeigt diese an, bis das Kontingent von 4.000 Zeichen erreicht ist. Dieser Vorgang zeigt die Protokollmeldungen an, die vor dem Auslösen der Ausnahme ausgegeben wurden. Dabei handelt es sich um die relevantesten Nachrichten für die Fehlerbehebung.

Weitere Informationen zum Anzeigen von Compliance-Informationen finden Sie unter AWS Systems Manager-Compliance.

Zuordnungsfehler beeinflussen die Compliance-Berichterstattung

Wenn die State Manager-Zuordnung fehlschlägt, werden keine Compliance-Daten gemeldet. Wenn Systems Manager beispielsweise versucht, ein Chef Kochbuch aus einem S3-Bucket herunterzuladen, für den der Knoten keine Zugriffsberechtigung hat, schlägt die Zuordnung fehl und Systems Manager meldet keine Compliance-Daten.

Anleitung: Automatische Aktualisierung von SSM Agent (CLI)

Mit den folgenden Schritten können Sie eine State Manager-Zuordnung mithilfe der AWS Command Line Interface erstellen. Durch die Zuordnung wird der SSM Agent automatisch entsprechend einem von Ihnen angegebenen Zeitplan aktualisiert. Mehr über SSM Agent erfahren Sie unter Arbeiten mit

SSM Agent. Informationen zum Anpassen des Aktualisierungszeitplans für SSM Agent mithilfe der Konsole finden Sie unter Automatische Aktualisierung von SSM Agent.

Um über SSM Agent Updates benachrichtigt zu werden, abonnieren Sie die Seite SSM Agent Versionshinweise auf GitHub.

Bevor Sie beginnen

Bevor Sie die folgenden Schritte ausführen, stellen Sie sicher, dass Sie mindestens eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance für Linux, macOS oder Windows Server ausführen, die für Systems Manager konfiguriert ist. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

Wenn Sie eine Zuordnung erstellen, indem Sie entweder die AWS CLI oder verwenden AWS Tools for Windows PowerShell, verwenden Sie den Parameter , `--Targets` um Instances anzuvisieren, wie im folgenden Beispiel gezeigt. Verwenden Sie nicht den Parameter `--InstanceID`. Der Parameter `--InstanceID` ist veraltet.

So erstellen Sie eine Zuordnung für die automatische Aktualisierung von SSM Agent

1. Installieren und konfigurieren Sie die AWS Command Line Interface (AWS CLI), falls noch nicht geschehen.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um eine Zuordnung zu erstellen, indem Sie Instances mithilfe von Amazon Elastic Compute Cloud (Amazon EC2)-Tags anvisieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Der `Schedule`-Parameter legt einen Zeitplan für die Ausführung der Zuordnung an jedem Sonntagmorgen um 2:00 Uhr (UTC) fest.

   State Manager-Zuordnungen unterstützen nicht alle Cron- und Rate-Ausdrücke. Weitere Informationen zum Erstellen von Cron- und Rate-Ausdrücken für Zuordnungen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System 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 *)"
```

Sie können auf mehrere Instances abzielen, indem Sie Instance-IDs in einer durch Kommas getrennten Liste angeben.

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 *)"
```

Sie können die Version des SSM Agent angeben, den Sie aktualisieren möchten.

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
```

Das System gibt unter anderem folgende Informationen zurück

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

Das System versucht, die Zuordnung für die Instances zu erstellen und wendet den Status nach der Erstellung an. Der Zuordnungsstatus lautet `Pending` (Schwebend).

3. Führen Sie den folgenden Befehl aus, um einen aktualisierten Status der erstellten Zuordnung anzuzeigen.

```
aws ssm list-associations
```

Wenn auf Ihren Instances nicht die neueste Version von SSM Agent ausgeführt wird, wird der Status `Failed` angezeigt. Bei Veröffentlichung einer neuen SSM Agent-Version wird der neue Agent automatisch von der Zuordnung installiert und der Status zeigt `Success` an.

Anleitung: Automatische Aktualisierung von PV-Treibern auf EC2-Instances für Windows Server (Konsole)

Amazon Windows-Amazon Machine Images (AMIs) enthalten eine Reihe von Treibern, die den Zugriff auf virtualisierte Hardware ermöglichen. Diese Treiber werden von Amazon Elastic Compute Cloud (Amazon EC2) verwendet, um Instance-Speicher und Amazon Elastic Block Store (Amazon EBS)-Volumes ihren Geräten zuzuordnen. Wir empfehlen, die aktuellen Treiber zu installieren, um die Stabilität und Leistung Ihrer EC2-Instances für Windows Server zu verbessern. Weitere Informationen zu PV-Treibern finden Sie unter AWS PV-Treiber.

Die folgende exemplarische Vorgehensweise zeigt Ihnen, wie Sie eine State Manager Zuordnung so konfigurieren, dass neue AWS PV-Treiber automatisch heruntergeladen und installiert werden, sobald die Treiber verfügbar sind. State Managerist eine Fähigkeit von. AWS Systems Manager

Bevor Sie beginnen

Bevor Sie die folgenden Schritte ausführen, stellen Sie sicher, dass Sie mindestens eine Amazon-EC2-Instance für Windows Server ausführen, die für Systems Manager verwaltet konfiguriert ist. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

So erstellen Sie eine State Manager-Zuordnung, die automatisch PV-Treiber aktualisiert

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie Create association (Zuordnung erstellen) aus.
4. Geben Sie im Feld Name einen aussagekräftigen Namen für die Assoziation ein.
5. Wählen Sie in der Liste Dokument die Option `AWS-ConfigureAWSPackage` aus.
6. Gehen Sie im Bereich Parameter wie folgt vor:

   - Wählen Sie für Action (Aktion) die Option Install (Installieren).
   - Wählen Sie für Installation type (Art der Installation) Uninstall and reinstall (Deinstallieren und neu installieren).

> **Note**
>
> Direkte Upgrades werden für dieses Paket nicht unterstützt. Es muss deinstalliert und neu installiert werden.

- Geben Sie unter Name **`AWSPVDriver`** ein.

Sie müssen nichts für Version und Zusätzliche Argumente eingeben.

7. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

> **Note**
>
> Wenn Sie Ziel-Instances mittels Tags auswählen und Tags angeben, die Linux-Instances zugeordnet sind, ist die Zuordnung zwar auf der Windows-Instance erfolgreich, schlägt jedoch auf den Linux-Instances fehl. Der Gesamtstatus der Zuordnung zeigt Failed (Fehler) an.

8. Wählen Sie im Bereich Zeitplan angeben aus, ob die Zuordnung nach einem von Ihnen konfigurierten Zeitplan oder nur einmal ausgeführt werden soll. Aktualisierte PV-Treiber werden mehrere Male pro Jahr veröffentlicht. Wenn Sie möchten, können Sie die Zuordnung einmal pro Monat ausführen lassen.
9. Wählen Sie im Bereich Erweiterte Optionen für den Schweregrad der Einhaltung einen Schweregrad für die Zuordnung aus. In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem Schweregrad, den Sie hier angeben. Weitere Informationen finden Sie unter Informationen zu State Manager-Zuordnungs-Compliance.
10. Für Rate control (Ratenregelung):

- Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

11. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

12. (Optional) Wählen Sie im Bereich CloudWatch Alarm unter Alarmname einen CloudWatch Alarm aus, der auf Ihre Assoziation zur Überwachung angewendet werden soll.

> **Note**
>
> Bitte beachten Sie die folgenden Informationen über diesen Schritt.
>
> - Die Liste der Alarme zeigt maximal 100 Alarme. Wenn Sie Ihren Alarm nicht in der Liste sehen, verwenden Sie den, AWS Command Line Interface um die Zuordnung zu erstellen. Weitere Informationen finden Sie unter Erstellen einer Zuordnung (Befehlszeile).
> - Um Ihrem Befehl einen CloudWatch Alarm anzuhängen, muss der IAM-Principal, der die Zuordnung erstellt, über die entsprechende Berechtigung für die `iam:createServiceLinkedRole` Aktion verfügen. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden.
> - Ausstehende Befehlsaufrufe oder Automatisierungen werden nicht ausgeführt, wenn Ihr Alarm aktiviert wird.

13. Wählen Sie Create association (Zuordnung erstellen) und dann Close (Schließen) aus. Das System versucht, die Zuordnung auf den Instances zu erstellen und den Status sofort anzuwenden.

    Wenn Sie die Zuordnung auf einer oder mehreren Amazon-EC2-Instances für Windows Server erstellt haben, ändert sich der Status zu Success (Erfolg). Wenn Ihre Instances nicht ordnungsgemäß für Systems Manager konfiguriert sind, oder wenn Sie versehentlich Linux-Instances ausgewählt haben, wird der Status Failed (Fehler) angezeigt.

    Wenn der Status Failed (Fehlgeschlagen) lautet, wählen Sie die Zuordnungs-ID aus, wählen Sie die Registerkarte Resource (Ressourcen) und überprüfen Sie dann, ob die Zuordnung auf Ihren EC2-Instances für Windows Server erfolgreich erstellt wurde. Wenn EC2-Instances für den Status Fehlgeschlagen Windows Server anzeigen, stellen Sie sicher, dass die auf der Instance ausgeführt SSM Agent wird, und stellen Sie sicher, dass die Instance mit einer AWS Identity and Access Management (IAM) -Rolle für Systems Manager konfiguriert ist. Weitere Informationen finden Sie unter Einrichten AWS Systems Manager.

AWS Systems Manager Patch Manager

Patch Manager, eine Funktion von AWS Systems Manager, automatisiert den Prozess des Patchens verwalteter Knoten sowohl mit sicherheitsrelevanten Updates als auch mit anderen Arten von Updates.

> **Important**
>
> Ab dem 22. Dezember 2022 stellt Systems Manager Unterstützung für Patch-Richtlinien bereit, die die neue und empfohlene Methode zur Konfiguration Ihrer Patching-Vorgänge sind. Mit einer einzelnen Patch-Richtlinienkonfiguration können Sie Patches für alle Konten in allen Regionen in Ihrer Organisation, nur für die von Ihnen ausgewählten Konten und Regionen oder für ein einzelnes Konto-Region-Paar definieren. Weitere Informationen finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien.

Sie können Patch Manager verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows Server ist der Anwendungssupport auf Updates für Microsoft-Anwendungen beschränkt.) Sie können mit Patch Manager Service Packs auf Windows-Instances installieren und Nebenversionsupgrades auf Linux-Knoten ausführen. Sie können Flotten von Amazon Elastic Compute Cloud (Amazon EC2)-Instances, Edge-Geräte, On-Premises-Server und virtuelle Maschinen (VMs) nach Betriebssystemtyp patchen. Dazu gehören unterstützte Versionen mehrerer Betriebssysteme, wie unter Patch Manager-Voraussetzungen aufgeführt. Sie können Instances nur auf Patches hin durchsuchen und dann einen Bericht zu fehlenden Patches anzeigen oder automatisch alle fehlenden Patches installieren. Um mit Patch Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Patch Manager aus.

> **Note**
>
> AWS testet keine Patches, bevor sie in verfügbar gemacht werden. Patch Manager Der Patch Manager unterstützt außerdem keine Upgrades von Hauptversionen von Betriebssystemen wie Windows Server 2016 auf Windows Server 2019 oder SUSE Linux Enterprise Server (SLES) 12.0 auf SLES 15.0.
>
> Für Linux-basierte Betriebssysteme, die einen Schweregrad für Patches melden, verwendet Patch Manager den vom Softwareherausgeber gemeldeten Schweregrad für den Update-Hinweis oder den einzelnen Patch. Patch Manager leitet keinen Schweregrad aus Drittquellen

wie dem Common Vulnerability Scoring System (CVSS) oder aus Metriken ab, die von der National Vulnerability Database (NVD) veröffentlicht werden.

Patch-Baselines

Patch Manager verwendet Patch-Baselines, die Regeln für die automatische Genehmigung von Patches innerhalb weniger Tage nach ihrer Veröffentlichung enthalten, zusätzlich zu den optionalen Listen der genehmigten und abgelehnten Patches. Wenn ein Patching-Vorgang ausgeführt wird, vergleicht Patch Manager die Patches, die derzeit auf einen verwalteten Knoten angewendet werden, mit denen, die gemäß den in der Patch-Baseline festgelegten Regeln angewendet werden sollten. Sie können auswählen, dass Patch Manager Ihnen nur einen Bericht über fehlende Patches anzeigt (ein `Scan`-Vorgang), oder Sie können auswählen, dass Patch Manager automatisch alle Patches installiert, die es auf einem verwalteten Knoten findet (ein `Scan and install`-Vorgang).

Methoden für das Patchen von Vorgängen

Patch Manager bietet derzeit vier Methoden zum Ausführen von `Scan`- und `Scan and install`-Operationen:

- (Empfohlen) Eine in konfigurierte Patch-Richtlinie Quick Setup — Basierend auf der Integration mit AWS Organizations können mit einer einzigen Patch-Richtlinie Patch-Zeitpläne und Patch-Baselines für eine gesamte Organisation definiert werden, einschließlich mehrerer AWS-Konten und all AWS-Regionen dieser Konten. Eine Patch-Richtlinie kann auch nur auf einige Organisationseinheiten (OUs) in einer Organisation ausgerichtet sein. Sie können eine einzige Patch-Richtlinie verwenden, um nach verschiedenen Zeitplänen zu scannen und zu installieren. Weitere Informationen finden Sie unter Patch Manager Patching-Konfiguration der Organisation und Verwenden von Quick Setup-Patch-Richtlinien.
- Eine in Quick Setup konfigurierte Host-Management-Option – Host-Management-Konfigurationen werden auch durch die Integration mit AWS Organizations unterstützt, wodurch die Ausführung eines Patching-Vorgangs für eine ganze Organisation möglich ist. Diese Option ist jedoch darauf beschränkt, anhand der aktuellen Standard-Patch-Baseline nach fehlenden Patches zu suchen und Ergebnisse in Compliance-Berichten bereitzustellen. Mit dieser Vorgangsmethode können keine Patches installiert werden. Weitere Informationen finden Sie unter Amazon-EC2-Host-Verwaltung.
- Ein Wartungsfenster zum Ausführen eines Patch-**`Scan`** oder einer **`Install`**-Aufgabe – Ein Wartungsfenster, das Sie in der Systems-Manager-Funktion mit dem Namen Maintenance Windows einrichten, kann so konfiguriert werden, dass es verschiedene Arten von Aufgaben nach einem von Ihnen definierten Zeitplan ausführt. Eine Aufgabe vom Run Command-Typ kann

verwendet werden, um `Scan` oder `Scan and install` Aufgaben auf einem Satz verwalteter Knoten Ihrer Wahl auszuführen. Jede Aufgabe im Wartungsfenster kann auf verwaltete Knoten in nur einem einzigen AWS-Konto Paar abzielen.AWS-Region Weitere Informationen finden Sie unter Walkthrough: Erstellen eines Wartungsfensters für das Einspielen von Patches (Konsole).

- Ein „Patch now“ (Jetzt patchen)-On-Demand-Vorgang in Patch Manager – Mit der Option Patch now (Jetzt patchen) können Sie Zeitplan-Einrichtungen umgehen, wenn Sie verwaltete Knoten so schnell wie möglich patchen müssen. Mit Patch now (Jetzt patchen) geben Sie an, ob der `Scan`- oder `Scan and install`-Vorgang ausgeführt werden soll und auf welchen verwalteten Knoten der Vorgang ausgeführt werden soll. Sie können auch festlegen, dass Systems Manager Manager-Dokumente (SSM-Dokumente) während des Patchvorgangs als Lifecycle-Hooks ausgeführt werden. Jeder Patch-Now-Vorgang kann auf verwaltete Knoten in nur einem einzigen Paar abzielen AWS-Konto.AWS-Region Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten.

Compliance-Meldung

Nach einer `Scan`-Operation können Sie die Systems-Manager-Konsole verwenden, um Informationen darüber anzuzeigen, welche Ihrer verwalteten Knoten die Patch-Compliance nicht erfüllen und welche Patches auf jedem dieser Knoten fehlen. Sie können auch Patch-Compliance-Berichte im CSV-Format generieren, die an einen Amazon Simple Storage Service (Amazon S3)-Bucket Ihrer Wahl gesendet werden. Sie können einmalige Berichte erstellen oder Berichte nach einem regelmäßigen Zeitplan erstellen. Für einen einzelnen verwalteten Knoten enthalten Berichte Details aller Patches für den Knoten. Für einen Bericht über alle verwaltete Knoten wird nur eine Zusammenfassung der fehlenden Patches bereitgestellt. Nachdem ein Bericht generiert wurde, können Sie ein Tool wie Amazon verwenden, QuickSight um die Daten zu importieren und zu analysieren. Weitere Informationen finden Sie unter Arbeiten mit Patch-Compliance-Berichten.

> **Note**
>
> Ein durch die Verwendung einer Patch-Richtlinie generiertes Compliance-Element hat den Ausführungstyp `PatchPolicy`. Ein Compliance-Element, das nicht in einem Patch-Richtlinienvorgang generiert wurde, hat den Ausführungstyp `Command`.

Integrationen

Patch Managerintegriert sich in die folgenden anderen AWS-Services:

- AWS Identity and Access Management (IAM) — Verwenden Sie IAM, um zu steuern, welche Benutzer, Gruppen und Rollen Zugriff Patch Manager auf Operationen haben. Weitere Informationen finden Sie unter Funktionsweise von AWS Systems Manager mit IAM und Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.
- AWS CloudTrail— Wird verwendet CloudTrail , um einen überprüfbaren Verlauf von Patch-Vorgängen aufzuzeichnen, die von Benutzern, Rollen oder Gruppen ausgelöst wurden. Weitere Informationen finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.
- AWS Security Hub— Daten zur Patch-Konformität von Patch Manager können an gesendet werden. AWS Security Hub Mit dem Security Hub erhalten Sie einen umfassenden Überblick über Ihre Sicherheitswarnungen und den Compliance-Status mit hoher Priorität. Er überwacht auch den Patching-Status Ihrer Flotte. Weitere Informationen finden Sie unter Integrieren Patch Manager mit AWS Security Hub.
- AWS Config— Richten Sie die Aufzeichnung ein AWS Config , um die Amazon EC2 EC2-Instance-Verwaltungsdaten im Patch Manager Dashboard anzuzeigen. Weitere Informationen finden Sie unter Patch-Dashboard-Zusammenfassungen anzeigen.

Themen

- Verwenden von Quick Setup-Patch-Richtlinien
- Patch Manager-Voraussetzungen
- So funktionieren Patch Manager-Operationen
- Über SSM-Dokumente für das Patchen von verwalteten Knoten
- Über Patch-Baselines
- Verwenden von Kernel Live Patching auf von Amazon Linux 2 verwalteten Knoten
- Arbeiten mit Patch Manager (Konsole)
- Arbeiten mit Patch Manager (AWS CLI)
- AWS Systems ManagerPatch ManagerTutorials
- Fehlerbehebung für Patch Manager

Verwenden von Quick Setup-Patch-Richtlinien

Ab dem 22. Dezember 2022 Patch Manager bietet eine neue, empfohlene Methode zum Konfigurieren von Patches für Ihre Organisation und AWS-Konten durch die Verwendung von Patch-Richtlinien.

Eine Patch-Richtlinie ist eine Konfiguration, die Sie mit Quick Setup, einer Funktion von AWS Systems Manager, einrichten. Patch-Richtlinien bieten eine umfassendere und zentralisiertere Kontrolle über Ihre Patching-Vorgänge, als dies mit früheren Methoden zum Konfigurieren von Patches möglich war. Patch-Richtlinien können mit allen von Patch Manager unterstützten Betriebssystemen verwendet werden, einschließlich unterstützter Versionen von Linux, macOS und Windows Server. Informationen zum Erstellen einer Patch-Richtlinie finden Sie unter Patch Manager Patching-Konfiguration der Organisation.

Hauptfeatures von Patch-Richtlinien

Anstatt andere Methoden zum Patchen Ihrer Knoten zu verwenden, verwenden Sie eine Patch-Richtlinie, um die Vorteile dieser Hauptfeatures zu nutzen:

- Einzelkonfiguration – Das Einrichten von Patching-Vorgängen mithilfe eines Wartungsfensters oder einer State Manager-Zuordnung kann mehrere Aufgaben in verschiedenen Teilen der Systems-Manager-Konsole erfordern. Mithilfe einer Patch-Richtlinie können alle Ihre Patching-Vorgänge in einem einzigen Assistenten eingerichtet werden.
- Unterstützung mehrerer Konten/Multi-Regionen – Mithilfe eines Wartungsfensters, einer State Manager Zuordnung oder der Patch now-Funktion in sind Sie darauf beschränktPatch Manager, verwaltete Knoten in einem einzigen AWS-Konto-AWS-Region Paar anzuvisieren. Wenn Sie mehrere Konten und mehrere Regionen verwenden, können Ihre Einrichtungs- und Wartungsaufgaben viel Zeit in Anspruch nehmen, da Sie Einrichtungsaufgaben in jedem Konto-Region-Paar durchführen müssen. Wenn Sie jedoch verwenden AWS Organizations, können Sie eine Patch-Richtlinie einrichten, die für alle Ihre verwalteten Knoten in allen AWS-Regionen in allen Ihren gilt AWS-Konten. Wenn Sie möchten, kann eine Patch-Richtlinie auch nur für bestimmte Organisationseinheiten (OEs) in den von Ihnen gewählten Konten und Regionen gelten. Eine Patch-Richtlinie kann auf Wunsch auch für ein einzelnes lokales Konto gelten.
- Installationsunterstützung auf Organisationsebene – Die vorhandene Host-Management-Konfigurationsoption in Quick Setup bietetUnterstützung für einen täglichen Scan Ihrer verwalteten Knoten auf Patch-Compliance. Dieser Scan wird jedoch zu einem vorher festgelegten Zeitpunkt durchgeführt und liefert nur Patch-Compliance-Informationen. Es werden keine Patch-Installationen durchgeführt. Mithilfe einer Patch-Richtlinie können Sie unterschiedliche Zeitpläne für das Scannen

und Installieren festlegen. Sie können auch die Häufigkeit und Zeit dieser Vorgänge auswählen, indem Sie benutzerdefinierte CRON- oder Rate-Ausdrücke verwenden. Sie könnten beispielsweise jeden Tag nach fehlenden Patches suchen, um regelmäßig aktualisierte Compliance-Informationen bereitzustellen. Ihr Installationsplan könnte jedoch nur einmal pro Woche sein, um unerwünschte Ausfallzeiten zu vermeiden.

- Vereinfachte Auswahl von Patch-Baselines – Patch-Richtlinien enthalten weiterhin Patch-Baselines, und es gibt keine Änderungen an der Art und Weise, wie Patch-Baselines konfiguriert werden. Wenn Sie jedoch eine Patch-Richtlinie erstellen oder aktualisieren, können Sie die AWS verwaltete oder benutzerdefinierte Baseline, die Sie für jeden Betriebssystemtyp (OS) verwenden möchten, in einer einzigen Liste auswählen. Es ist nicht erforderlich, die Standard-Baseline für jeden Betriebssystemtyp in separaten Aufgaben anzugeben.

> **Note**
>
> Wenn Patching-Vorgänge, die auf einer Patch-Richtlinie basieren, ausgeführt werden, verwenden diese das `AWS-RunPatchBaseline`-SSM-Dokument. Weitere Informationen finden Sie unter Informationen über das `AWS-RunPatchBaseline` SSM-Dokument.

Ähnliche Informationen

Zentrale Bereitstellung von Patching-Operationen in Ihrer gesamten AWS Organisation mithilfe von Systems Manager Quick Setup (Blog AWS zu Cloud-Operationen und Migrationen)

Weitere Unterschiede bei Patch-Richtlinien

Hier sind einige weitere Unterschiede, die bei der Verwendung von Patch-Richtlinien anstelle der vorherigen Methoden zum Konfigurieren von Patches zu beachten sind:

- Keine Patchgruppen erforderlich – Bei früheren Patching-Vorgängen konnten Sie mehrere Knoten so kennzeichnen, dass sie zu einer Patch-Gruppe gehören, und dann die Patch-Baseline angeben, die für diese Patch-Gruppe verwendet werden soll. Wenn keine Patch-Gruppe definiert wurde, patcht Patch Manager die Instances mit der aktuellen Standard-Patch-Baseline für den OS-Typ. Durch die Verwendung von Patch-Richtlinien ist es nicht mehr erforderlich, Patch-Gruppen einzurichten und zu verwalten.
- Seite „Patching konfigurieren“ entfernt – Vor der Veröffentlichung von Patch-Richtlinien konnten Sie auf der Seite Configure patching (Patching konfigurieren) Standardwerte für die zu patchenden

Knoten, einen Patch-Zeitplan und einen Patching-Vorgang angeben. Diese Seite wurde von Patch Manager entfernt. Diese Optionen werden jetzt in Patch-Richtlinien festgelegt.

- Keine „Patch now“-Unterstützung – Die Fähigkeit, Knoten bei Bedarf zu patchen, ist immer noch auf ein einzelnes AWS-Konto-AWS-Region Paar gleichzeitig beschränkt. Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten.
- Patch-Richtlinien und Compliance-Informationen – Wenn Ihre verwalteten Knoten gemäß einer Patching-Richtlinienkonfiguration auf Compliance gescannt werden, werden Ihnen Compliance-Daten zur Verfügung gestellt. Sie können die Daten auf die gleiche Weise wie bei anderen Methoden des Compliance-Scannens anzeigen und bearbeiten. Obwohl Sie eine Patch-Richtlinie für eine gesamte Organisation oder mehrere Organisationseinheiten einrichten können, werden Compliance-Informationen für jedes AWS-Konto-AWS-Region Paar einzeln gemeldet. Weitere Informationen finden Sie unter Arbeiten mit Patch-Compliance-Berichten.
- Zuordnungs-Compliance-Status und Patch-Richtlinien – Der Patching-Status für einen verwalteten Knoten, der einer Quick Setup Patch-Richtlinie unterliegt, entspricht dem Status der State Manager Zuordnungsausführung für diesen Knoten. Wenn der Status der Zuordnungsausführung lautet`Compliant`, wird der Patching-Status für den verwalteten Knoten ebenfalls als markiert`Compliant`. Wenn der Status der Zuordnungsausführung lautet`Non-Compliant`, wird der Patching-Status für den verwalteten Knoten ebenfalls als markiert`Non-Compliant`.

AWS-Regionen unterstützt für Patch-Richtlinien

Patch-Richtlinien-Konfigurationen in Quick Setup werden derzeit in den folgenden Regionen unterstützt:

- USA Ost (Ohio): (us-east-2)
- USA Ost (Nord-Virginia): (us-east-1)
- USA West (Nordkalifornien) (us-west-1)
- USA West (Oregon): (us-west-2)
- Asien-Pazifik (Mumbai): (ap-south-1)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Kanada (Zentral): (ca-central-1)

- Europa (Frankfurt) (eu-central-1)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Europa (Stockholm) (eu-north-1)
- Südamerika (São Paulo) (sa-east-1)

Patch Manager-Voraussetzungen

Stellen Sie sicher, dass Sie die erforderlichen Voraussetzungen erfüllt habenPatch Manager, bevor Sie eine Funktion von verwenden AWS Systems Manager.

Themen

- SSM Agent-Version
- Python-Version
- Konnektivität zur Patch-Quelle
- S3-Endpunkt-Zugriff
- Unterstützte Betriebssysteme für Patch Manager

SSM Agent-Version

Version 2.0.834.0 oder höher von SSM Agent muss auf dem verwalteten Knoten ausgeführt werden, den Sie mit Patch Manager verwalten möchten.

> **Note**
>
> Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite mit den SSM

AgentVersionshinweisen aufGitHub, um Benachrichtigungen über SSM Agent Updates zu erhalten.

Python-Version

Für macOS und die meisten Linux-Betriebssysteme (OS), unterstützt Patch Manager derzeit die Python-Versionen 2.6–3.10. Die Ubuntu Server Betriebssysteme AlmaLinuxDebian Server,Raspberry Pi OS, und benötigen eine unterstützte Version von Python 3 (3.0-3.10).

Konnektivität zur Patch-Quelle

Wenn Ihre verwalteten Knoten keine direkte Verbindung zum Internet haben und Sie eine Amazon Virtual Private Cloud (Amazon VPC) mit einem VPC-Endpunkt verwenden, müssen Sie sicherstellen, dass die Knoten Zugriff auf die Quell-Patch-Verzeichnisse (Repos) haben. Auf Linux-Knoten werden Patch-Updates normalerweise von den auf dem Knoten konfigurierten Remote-Repos heruntergeladen. Daher muss der Knoten eine Verbindung zu den Repos herstellen können, damit die Patches ausgeführt werden können. Weitere Informationen finden Sie unter Wie Sicherheitspatches ausgewählt werden.

Von Windows Server verwaltete Knoten müssen eine Verbindung zum Windows Update Catalog oder Windows Server Update Services (WSUS) herstellen können. Vergewissern Sie sich, dass Ihre Knoten über ein Internet-Gateway, ein NAT-Gateway oder eine NAT-Instance eine Verbindung zum Microsoft Update Catalog hergestellt haben. Wenn Sie WSUS verwenden, stellen Sie sicher, dass der Knoten eine Verbindung zum WSUS-Server in Ihrer Umgebung hat. Weitere Informationen finden Sie unter Problem: Der verwaltete Knoten hat keinen Zugriff auf Windows Update Catalog oder WSUS.

S3-Endpunkt-Zugriff

Unabhängig davon, ob Ihre verwalteten Knoten in einem privaten oder öffentlichen Netzwerk betrieben werden, ohne Zugriff auf die erforderlichen AWS verwalteten Amazon Simple Storage Service (Amazon S3) -Buckets schlagen Patch-Vorgänge fehl. Informationen zu den S3-Buckets, auf die Ihre verwalteten Knoten zugreifen können müssen, finden Sie unter SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets und Verbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

Unterstützte Betriebssysteme für Patch Manager

Die Patch Manager-Funktion unterstützt nicht dieselben Betriebssystemversionen, die von anderen Systems Manager-Funktionen unterstützt werden. Beispielsweise unterstützt Patch Manager nicht CentOS 6.3 oder Raspberry Pi OS 8 (Jessie). (Eine vollständige Liste der von Systems Manager unterstützten Betriebssysteme finden Sie unter Unterstützte Betriebssysteme für Systems Manager.) Stellen Sie daher sicher, dass die verwalteten Knoten, die Sie mit Patch Manager verwenden möchten, eines der Betriebssysteme ausführen, die in der folgenden Tabelle aufgeführt sind.

> **Note**
>
> Patch Managerstützt sich auf die Patch-Repositorys, die auf einem verwalteten Knoten konfiguriert sind, z. B. Windows Update Catalog und Windows Server Update Services für Windows, um verfügbare Patches für die Installation abzurufen. Daher können Betriebssystemversionen, die am Ende ihres Lebenszyklus (EOL) sind, Patch Manager möglicherweise nicht in der Lage, über die neuen Updates zu berichten, wenn keine neuen Updates verfügbar sind. Dies kann daran liegen, dass vom Linux-Distributionsbetreuer, Microsoft oder Apple keine neuen Updates veröffentlicht wurden oder dass der verwaltete Knoten nicht über die richtige Lizenz für den Zugriff auf die neuen Updates verfügt.
> Patch Managermeldet den Konformitätsstatus anhand der verfügbaren Patches auf dem verwalteten Knoten. Wenn auf einer Instanz ein EOL-Betriebssystem ausgeführt wird und keine Updates verfügbar sind, wird der Knoten daher Patch Manager möglicherweise als konform gemeldet, je nachdem, welche Patch-Baselines für den Patchvorgang konfiguriert wurden.

Betriebssystem	Details
Linux	• AlmaLinux 8.3—8.7, 9.0—9.2 • Amazon Linux 2012.03–2018.03 • Amazon Linux 2 Version 2.0 und alle späteren Versionen • Amazon Linux 2022 • Amazon Linux 2023 • CentOS 6.5–7.9, 8.0–8.5 • CentOS Stream8

Betriebssystem	Details
	• Debian Server8.x, 9.x, 10.x, 11.x und 12.x • Oracle Linux 7.5–8.7, 9.0–9.2 • Raspberry Pi OS (früher Raspbian) 9 (Stretch) • Red Hat Enterprise Linux() 6,5—8,9, 9,0—9,3 RHEL • Rocky Linux 8.4–8.7, 9.0–9.2 • SUSE Linux Enterprise Server() 12.0 SLES und später 12. x-Versionen; 15.0 - 15.5 • Ubuntu Server14.04 LTS, 16.04 LTS, 18.04 LTS, 20.04 LTS, 20.10 STR, 22.04 LTS und 23.04

Betriebssystem	Details
macOS	11.3.1, 11.4–11.7 (Big Sur) 12.0–12.6 (Monterey) 13.0–13.5 (Ventura) 14,0 (Sonoma) macOS Updates Patch Manager unterstützt keine Updates oder Upgrades für macOS, z. B. von 12.x auf 13.x oder 13.1 auf 13.2. Für die Aktualisierung der Betriebssystemversion von macOS empfehlen wir, die in Apple integrierten Mechanismen zu verwenden. Weitere Informationen finden Sie auf der Website mit Entwicklerdokumentation von Apple unter Device Management. Unterstützung für Homebrew Das Open-Source-Softwarepaketverwaltungssystem Homebrew hat die Unterstützung für macOS 10.14.x (Mojave) und 10.15.x (Catalina) eingestellt. Aus diesem Grund werden Patch-Operationen für diese Versionen derzeit nicht unterstützt. Regionsunterstützung macOSwird nicht in allen unterstützt. AWS-Regionen Weitere Informationen zur Amazon EC2-Unterstützung für macOS finden Sie unter Amazon EC2 Mac-Instances im Amazon EC2 EC2-Benutzerhandbuch. macOS-Edge-Geräte

Betriebssystem	Details
	SSM Agentfür AWS IoT Greengrass Core-Geräte wird auf nicht unterstützt. macOS Sie können Patch Manager nicht verwenden, um macOS-Edge-Geräte zu patchen.

Betriebssystem	Details
Windows	Windows Server 2008 bis Windows Server 2022, einschließlich R2-Versionen. **Note** SSM Agentfür AWS IoT Greengras s Kerngeräte wird unter Windows 10 nicht unterstützt. Sie können Patch Manager nicht verwenden, um Windows-10-Edge-Geräte zu patchen. Über die Unterstützung von Windows Server 2008 Ab 14. Januar 2020 wird Windows Server 2008 für Feature- oder Sicherheitsupdates von Microsoft nicht mehr unterstützt. Legacy Amazon Machine Images (AMIs) für Windows Server 2008 und 2008 R2 enthalten immer noch die Version 2 vom vorinstallierten SSM Agent, aber Systems Manager unterstützt offiziell nicht mehr die 2008-Versionen und aktualisiert den Agenten für diese Versionen von Windows Server. Darüber hinaus ist SSM Agent Version 3 möglicherweise nicht mit allen Operationen auf Windows Server 2008 und 2008 R2 kompatibel. Die endgültige offiziell unterstützte Version von SSM Agent für Windows Server 2008 Versionen ist 2.3.1644.0. Informationen zur R2-Unterstützung für Windows Server 2012 und 2012 Windows Server2012 und 2012 R2 haben am 10. Oktober 2023 das Ende der Unterstüt

Betriebssystem	Details
	zung erreicht. Für die Verwendung Patch Manager mit diesen Versionen empfehlen wir, auch Extended Security Updates (ESU) von Microsoft zu verwenden. Weitere Informati onen finden Sie auf der Microsoft-Website unter Windows Server2012 und 2012 R2 erreichen das Ende des Supports.

So funktionieren Patch Manager-Operationen

Dieser Abschnitt enthält technische Details, die erklären, wie Patch Manager, eine Funktion von AWS Systems Manager, bestimmt, welche Patches installiert werden und wie er sie auf dem jeweiligen unterstützten Betriebssystem installiert. Für Linux-Betriebssysteme enthält er auch Informationen zur Angabe einer Quell-Repository, in einer benutzerdefinierten Patch-Baseline, für andere Patches als diejenigen, die standardmäßig auf einem verwalteten Knoten konfiguriert sind. Dieser Abschnitt bietet außerdem Informationen darüber, wie Patch-Baseline-Regeln auf verschiedenen Verteilungen des Linux-Betriebssystems funktionieren.

> **Note**
>
> Die Informationen in den folgenden Themen gelten unabhängig davon, welche Methode oder Art der Konfiguration Sie für Ihre Patching-Vorgänge verwenden:
>
> - Eine in Quick Setup konfigurierte Patch-Richtlinie
> - Eine in Quick Setup konfigurierte Host-Management-Option
> - Ein Wartungsfenster zum Ausführen eines Patch-`Scans` oder einer `Install`-Aufgabe
> - Eine On-Demand Patch now-Operation (Jetzt patchen)

Themen

- So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet
- Wie Sicherheitspatches ausgewählt werden
- So geben Sie ein alternatives Patch-Quell-Repository an (Linux)
- Wie Patches installiert werden

- Funktionsweise von Patch-Baseline-Regeln auf Linux-basierten Systemen
- Wichtige Unterschiede zwischen Linux- und Windows-Patching

So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet

> **Important**
>
> Die Informationen auf dieser Seite gelten für die Betriebssysteme (OSs) von Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 und Amazon Linux 2023 für Amazon Elastic Compute Cloud (Amazon EC2)-Instances. Die Pakete für diese Betriebssystemtypen werden von Amazon Web Services erstellt und verwaltet. Wie die Hersteller anderer Betriebssysteme ihre Pakete und Repositorys verwalten, wirkt sich darauf aus, wie ihre Veröffentlichungs- und Aktualisierungsdaten berechnet werden. Für Betriebssysteme außer Amazon Linux, Amazon Linux 2, Amazon Linux 2022 und Amazon Linux 2023 wie Red Hat Enterprise Linux (RHEL) und SUSE Linux Enterprise Server (SLES), finden Sie in der Dokumentation des Herstellers Informationen darüber, wie ihre Pakete aktualisiert und gewartet werden.

In den Einstellungen für benutzerdefinierte Patch-Baselines, die Sie erstellen, können Sie für die meisten Betriebssystemtypen angeben, dass Patches nach einer bestimmten Anzahl von Tagen automatisch für die Installation genehmigt werden. AWS stellt mehrere vordefinierte Patch-Baselines bereit, die automatische Genehmigungsdaten von 7 Tagen enthalten.

Eine Verzögerung der automatischen Genehmigung ist die Anzahl an Tagen, die gewartet werde soll, nachdem die Patch veröffentlicht wurde, bevor der Patch automatisch genehmigt wird. Beispielsweise erstellen Sie eine Regel mit der `CriticalUpdates`-Klassifizierung und konfigurieren sie für eine Verzögerung der automatischen Genehmigung um sieben Tage. Infolgedessen wird ein neuer kritischer Patch mit einem Veröffentlichungsdatum oder dem letzten Aktualisierungsdatum vom 7. Juli automatisch am 14. Juli genehmigt.

Um unerwartete Ergebnisse mit Verzögerungen bei der automatischen Genehmigung auf Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 und Amazon Linux 2023 zu vermeiden, ist es wichtig zu verstehen, wie ihre Veröffentlichungs- und Aktualisierungsdaten berechnet werden.

In den meisten Fällen wird die Wartezeit für die automatische Genehmigung vor der Installation von Patches aus einem `Updated Date`-Wert in `updateinfo.xml` und nicht aus einem `Release Date`-Wert berechnet. Im Folgenden finden Sie wichtige Details zu diesen Datumsberechnungen:

- Das `Release Date` ist das Datum, an dem ein Hinweis veröffentlicht wird. Dies bedeutet nicht, dass das Paket bereits in den zugehörigen Repositorys verfügbar ist.
- Das `Update Date` ist das letzte Datum, an dem der Hinweis aktualisiert wurde. Eine Aktualisierung eines Hinweises kann etwas so Kleines wie eine Text- oder Beschreibungsaktualisierung darstellen. Dies bedeutet nicht, dass das Paket ab diesem Datum veröffentlicht wurde oder notwendigerweise in den zugehörigen Repositorys verfügbar ist.

  Dies bedeutet, dass ein Paket einen `Update Date`-Wert vom 7. Juli haben kann, aber erst (zum Beispiel) am 13. Juli für die Installation verfügbar sein kann. Angenommen, in diesem Fall wird am 14. Juli in einem `Install`-Vorgang eine Patch-Baseline ausgeführt, die eine 7-tägige automatische Genehmigungsverzögerung angibt. Da der `Update Date`-Wert sieben Tage vor dem Ausführungsdatum liegt, werden die Patches und Updates im Paket am 14. Juli installiert. Die Installation erfolgt, obwohl erst ein Tag vergangen ist, seit das Paket für die eigentliche Installation verfügbar ist.
- Ein Paket, das Betriebssystem- oder Anwendungs-Patches enthält, kann nach der ersten Veröffentlichung mehrmals aktualisiert werden.
- Ein Paket kann in den AWS verwalteten Repositorys freigegeben, aber dann zurückgesetzt werden, wenn später Probleme damit erkannt werden.

Bei einigen Patch-Vorgängen sind diese Faktoren möglicherweise nicht wichtig. Wenn beispielsweise eine Patch-Baseline so konfiguriert ist, dass ein Patch mit den Schweregraden `Low` und `Medium` und der Klassifizierung `Recommended` installiert wird, kann jede Verzögerung der automatischen Genehmigung nur geringe Auswirkungen auf Ihren Betrieb haben.

In Fällen, in denen das Timing kritischer Patches oder Patches mit hohem Schweregrad wichtiger ist, sollten Sie möglicherweise mehr Kontrolle darüber haben, wann Patches installiert werden. Die empfohlene Methode hierfür ist die Verwendung alternativer Patch-Quell-Repositorys anstelle der Standard-Repositorys für Patch-Vorgänge auf einem verwalteten Knoten.

Sie können beim Erstellen einer benutzerdefinierten Patch-Baseline alternative Patch-Quell-Repositorys angeben. Für jede benutzerdefinierte Patch-Baseline können Sie Patch-Quellkonfigurationen für bis zu 20 Versionen eines unterstützten Linux-Betriebssystems angeben. Weitere Informationen finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux).

Wie Sicherheitspatches ausgewählt werden

Der primäre Schwerpunkt von Patch Manager, einer Funktion von AWS Systems Manager, liegt auf der Installation von Sicherheits-Updates für Betriebssysteme auf verwalteten Knoten. Standardmäßig installiert Patch Manager daher nicht alle verfügbaren Patches, sondern eher eine kleinere Reihe von Patches mit Schwerpunkt auf der Sicherheit.

Für Linux-basierte Betriebssysteme, die einen Schweregrad für Patches melden, verwendet Patch Manager den vom Softwareherausgeber gemeldeten Schweregrad für den Update-Hinweis oder den einzelnen Patch. Patch Manager leitet keinen Schweregrad aus Drittquellen wie dem Common Vulnerability Scoring System (CVSS) oder aus Metriken ab, die von der National Vulnerability Database (NVD) veröffentlicht werden.

> **Note**
>
> Auf allen Linux-basierten Systemen, die von Patch Manager unterstützt werden, können Sie ein anderes für den verwalteten Knoten konfiguriertes Quell-Repository auswählen, typischerweise zur Installation von nicht-sicherheitsrelevanten Updates. Weitere Informationen finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux).

Der Rest dieses Abschnitts erläutert, wie Patch Manager-Sicherheits-Patches für die verschiedenen unterstützten Betriebssysteme auswählt.

Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, and Amazon Linux 2023

Vorkonfigurierte Repositorys werden auf Amazon Linux 1 und Amazon Linux 2 anders behandelt als auf Amazon Linux 2022 und Amazon Linux 2023.

Auf Amazon Linux 1 und Amazon Linux 2 verwendet der Systems Manager Patch Baseline Service vorkonfigurierte Repositorys auf dem verwalteten Knoten. Es gibt in der Regel zwei vorkonfigurierte Repositorys (Repos) auf einem Knoten:

Auf Amazon Linux 1

- Repo-ID: `amzn-main/latest`

  Repo-Name: `amzn-main-Base`
- Repo-ID: `amzn-updates/latest`

Repo-Name: `amzn-updates-Base`

Auf Amazon Linux 2

- Repo-ID: `amzn2-core/2/`*`architecture`*

  Repo-Name: `Amazon Linux 2 core repository`
- Repo-ID: `amzn2extra-docker/2/`*`architecture`*

  Repo-Name: `Amazon Extras repo for docker`

> **Note**
>
> *`Die Architektur`* kann x86_64 oder aarch64 sein.

Amazon Linux 2023 (AL2023)-Instances enthalten zunächst die Updates, die in der Version von AL2023 und der ausgewählten AMI verfügbar waren. Standardmäßig erhält Ihre AL203-Instance beim Start nicht automatisch zusätzliche kritische und wichtige Sicherheitsupdates. Mit dem Feature für deterministische Upgrades durch versionierte Repositorys in AL2023, die standardmäßig aktiviert ist, können Sie stattdessen Aktualisierungen nach einem Zeitplan durchführen, der Ihren spezifischen Anforderungen entspricht. Weitere Informationen finden Sie unter Deterministische Upgrades durch versionierte Repositories im Benutzerhandbuch von Amazon Linux 2023.

Unter Amazon Linux 2022 sind die vorkonfigurierten Repositorys an gesperrte Versionen von Paketaktualisierungen gebunden. Wenn neue Amazon Machine Images (AMIs) für Amazon Linux 2022 veröffentlicht werden, sind sie an eine bestimmte Version gebunden. Bei Patch-Aktualisierungen ruft Patch Manager die letzte gesperrte Version des Patch-Aktualisierungs-Repositorys ab und aktualisiert dann die Pakete auf dem verwalteten Knoten basierend auf dem Inhalt dieser gesperrten Version.

Auf AL2023 sieht das vorkonfigurierte Repository wie folgt aus:

- Repo-ID: `amazonlinux`

  Repository-Name: Amazon-Linux-2023-Repository

Bei Amazon Linux 2022 (Vorschauversion) sind die vorkonfigurierten Repositories an gesperrte Versionen von Paket-Updates gebunden. Wenn neue Amazon Machine Images (AMIs) für Amazon Linux 2022 veröffentlicht werden, sind sie an eine bestimmte Version gebunden. Bei Patch-Aktualisierungen ruft Patch Manager die letzte gesperrte Version des Patch-Aktualisierungs-Repositorys ab und aktualisiert dann die Pakete auf dem verwalteten Knoten basierend auf dem Inhalt dieser gesperrten Version.

Auf Amazon Linux 2022 sieht das vorkonfigurierte Repository wie folgt aus:

- Repo-ID: `amazonlinux`

  Repository-Name: Amazon-Linux-2022-Repository

> **Note**
>
> Alle Updates werden von den auf dem verwalteten Knoten konfigurierten Remote-Repos heruntergeladen. Daher muss der Knoten über einen ausgehenden Zugang zum Internet verfügen, um eine Verbindung zu den Repos herzustellen, damit das Patching durchgeführt werden kann.

Die verwalteten Knoten von Amazon Linux 1 und Amazon Linux 2 verwenden Yum als Paketmanager. Amazon Linux 2022 und Amazon Linux 2023 verwenden DNF als Paketmanager.

Beide Paket-Manager verwenden das Konzept einer Aktualisierungsbenachrichtigung in Form einer Datei mit dem Namen `updateinfo.xml`. Ein Update-Hinweis ist einfach eine Sammlung von Paketen, die bestimmte Probleme beheben. Alle Pakete in einem Update-Hinweis werden von Patch Manager als sicherheitsrelevant betrachtet. Einzelnen Paketen werden keine Klassifizierungen oder Schweregrade zugewiesen. Daher weist Patch Manager die Attribute eines Update-Hinweises den jeweiligen Paketen zu.

> **Note**
>
> Wenn Sie das Kontrollkästchen Funktionsupdates einschließen auf der Seite Patch-Baseline erstellen aktivieren, können Pakete, die nicht in einer `updateinfo.xml`-Datei klassifiziert sind (oder Pakete, die eine Datei ohne korrekt formatierte Klassifizierung, Schweregrad und Datenwerte), in die vorgefilterte Patchliste aufgenommen werden. Damit

ein Patch jedoch angewendet werden kann, muss er dennoch die benutzerspezifischen Patch-Baseline-Regeln erfüllen.

CentOS and CentOS Stream

Der Patch-Baseline-Service des Systems Managers auf CentOS und CentOS Stream verwendet vorkonfigurierte Repositorys (Repos) auf dem verwalteten Knoten. Die folgende Liste enthält Beispiele für ein fiktives CentOS 8.2 Amazon Machine Image (AMI):

- Repo-ID: `example-centos-8.2-base`

  Repo-Name: `Example CentOS-8.2 - Base`
- Repo-ID: `example-centos-8.2-extras`

  Repo-Name: `Example CentOS-8.2 - Extras`
- Repo-ID: `example-centos-8.2-updates`

  Repo-Name: `Example CentOS-8.2 - Updates`
- Repo-ID: `example-centos-8.x-examplerepo`

  Repo-Name: `Example CentOS-8.x – Example Repo Packages`

> **Note**
>
> Alle Updates werden von den auf dem verwalteten Knoten konfigurierten Remote-Repos heruntergeladen. Daher muss der Knoten über einen ausgehenden Zugang zum Internet verfügen, um eine Verbindung zu den Repos herzustellen, damit das Patching durchgeführt werden kann.

Von CentOS 6 und 7 verwaltete Knoten verwenden Yum als Paketmanager. CentOS-8- und CentOS Stream-Knoten verwenden DNF als Paketmanager. Beide Paketmanager verwenden das Konzept eines Update-Hinweises als einen aktualisierten Hinweis. Ein Update-Hinweis ist einfach eine Sammlung von Paketen, die bestimmte Probleme beheben.

CentOS und CentOS Stream Standard-Repos werden jedoch nicht mit einem Update-Hinweis konfiguriert. Das bedeutet, dass Patch Manager Pakete auf einem Standard-CentOS-

und CentOS Stream-Repos nicht erkennt. Um Patch Manager für die Verarbeitung von Paketen, die nicht in einem Update-Hinweis enthalten sind, zu aktivieren, müssen Sie die `EnableNonSecurity`-Markierung in den Patch-Baseline-Regeln aktivieren.

> **Note**
>
> CentOS und CentOS Stream-Update-Hinweise werden unterstützt. Repos mit Update-Hinweisen können nach dem Start heruntergeladen werden.

Debian Server and Raspberry Pi OS

Auf Debian Server und Raspberry Pi OS (früher Raspbian) verwendet der Patch-Baseline-Service des Systems Managers vorkonfigurierte Repositorys (Repos) auf der Instance. Diese vorkonfigurierten Repos werden verwendet, um eine aktualisierte Liste der verfügbaren Paket-Upgrades abzurufen. Dazu führt Systems Manager das Äquivalent eines `sudo apt-get update`-Befehls durch.

Pakete werden dann aus `debian-security` *`codename`*-Repos gefiltert. Das bedeutet, dass für jede Version von Patch Manager nur Upgrades identifiziert werdenDebian Server, die Teil des zugehörigen Repositorys für diese Version sind, und zwar wie folgt:

- Debian Server 8: `debian-security jessie`
- Debian Server9: `debian-security stretch`
- Debian Server10: `debian-security buster`
- Debian Server11: `debian-security bullseye`
- Debian Server12: `debian-security bookworm`

> **Note**
>
> Nur auf Debian Server 8: Da einige von Debian Server 8.* verwaltete Knoten auf ein überholtes Paket-Repository (`jessie-backports`) verweisen, führt Patch Manager zusätzliche Schritte aus, um sicherzustellen, dass Patch-Operationen erfolgreich ausgeführt werden. Weitere Informationen finden Sie unter Wie Patches installiert werden.

Oracle Linux

Der Patch-Baseline-Service des Systems Managers auf Oracle Linux verwendet vorkonfigurierte Repositorys (Repos) auf dem verwalteten Knoten. Es gibt in der Regel zwei vorkonfigurierte Repositorys (Repos) auf einem Knoten.

Oracle Linux 7:

- Repo-ID: `ol7_UEKR5/x86_64`

  Repo-Name: `Latest Unbreakable Enterprise Kernel Release 5 for Oracle Linux 7Server (x86_64)`
- Repo-ID: `ol7_latest/x86_64`

  Repo-Name: `Oracle Linux 7Server Latest (x86_64)`

Oracle Linux 8:

- Repo-ID: `ol8_baseos_latest`

  Repo-Name: `Oracle Linux 8 BaseOS Latest (x86_64)`
- Repo-ID: `ol8_appstream`

  Repo-Name: `Oracle Linux 8 Application Stream (x86_64)`
- Repo-ID: `ol8_UEKR6`

  Repo-Name: `Latest Unbreakable Enterprise Kernel Release 6 for Oracle Linux 8 (x86_64)`

Oracle Linux 9:

- Repo-ID: `ol9_baseos_latest`

  Repo-Name: `Oracle Linux 9 BaseOS Latest (x86_64)`
- Repo-ID: `ol9_appstream`

  Repo-Name: `Oracle Linux 9 Application Stream Packages(x86_64)`
- Repo-ID: `ol9_UEKR7`

Repo-Name: `Oracle Linux UEK Release 7 (x86_64)`

> **Note**
>
> Alle Updates werden von den auf dem verwalteten Knoten konfigurierten Remote-Repos heruntergeladen. Daher muss der Knoten über einen ausgehenden Zugang zum Internet verfügen, um eine Verbindung zu den Repos herzustellen, damit das Patching durchgeführt werden kann.

Von Oracle Linux verwaltete Knoten verwenden Yum als Paketmanager und Yum verwendet das Konzept eines Update-Hinweises in Form einer Datei namens `updateinfo.xml`. Ein Update-Hinweis ist einfach eine Sammlung von Paketen, die bestimmte Probleme beheben. Einzelnen Paketen werden keine Klassifizierungen oder Schweregrade zugewiesen. Aus diesem Grund weist Patch Manager die Attribute eines Update-Hinweises den jeweiligen Paketen zu und installiert Pakete basierend auf den Klassifizierungsfiltern, die in der Patch-Baseline angegeben sind.

> **Note**
>
> Wenn Sie das Kontrollkästchen Funktionsupdates einschließen auf der Seite Patch-Baseline erstellen aktivieren, können Pakete, die nicht in einer `updateinfo.xml`-Datei klassifiziert sind (oder Pakete, die eine Datei ohne korrekt formatierte Klassifizierung, Schweregrad und Datenwerte), in die vorgefilterte Patchliste aufgenommen werden. Damit ein Patch jedoch angewendet werden kann, muss er dennoch die benutzerspezifischen Patch-Baseline-Regeln erfüllen.

AlmaLinux, RHEL, and Rocky Linux

Ein AlmaLinuxRed Hat Enterprise Linux, und Rocky Linux der Systems Manager Patch Baseline Service verwendet vorkonfigurierte Repositorys (Repos) auf dem verwalteten Knoten. Es gibt in der Regel drei vorkonfigurierte Repositorys (Repos) auf einem Knoten.

Alle Updates werden von den auf dem verwalteten Knoten konfigurierten Remote-Repos heruntergeladen. Daher muss der Knoten über einen ausgehenden Zugang zum Internet

verfügen, um eine Verbindung zu den Repos herzustellen, damit das Patching durchgeführt werden kann.

> **Note**
>
> Wenn Sie das Kontrollkästchen Funktionsupdates einschließen auf der Seite Patch-Baseline erstellen aktivieren, können Pakete, die nicht in einer `updateinfo.xml`-Datei klassifiziert sind (oder Pakete, die eine Datei ohne korrekt formatierte Klassifizierung, Schweregrad und Datenwerte), in die vorgefilterte Patchliste aufgenommen werden. Damit ein Patch jedoch angewendet werden kann, muss er dennoch die benutzerspezifischen Patch-Baseline-Regeln erfüllen.

Red Hat Enterprise Linux7 verwaltete Knoten verwenden Yum als Paketmanager. AlmaLinux, Red Hat Enterprise Linux 8, und Rocky Linux verwaltete Knoten verwenden DNF als Paketmanager. Beide Paketmanager verwenden das Konzept eines Update-Hinweises als Datei namens `updateinfo.xml`. Ein Update-Hinweis ist einfach eine Sammlung von Paketen, die bestimmte Probleme beheben. Einzelnen Paketen werden keine Klassifizierungen oder Schweregrade zugewiesen. Aus diesem Grund weist Patch Manager die Attribute eines Update-Hinweises den jeweiligen Paketen zu und installiert Pakete basierend auf den Klassifizierungsfiltern, die in der Patch-Baseline angegeben sind.

RHEL 7

> **Note**
>
> Die folgenden Repository-IDs sind RHUI 2 zugeordnet. RHUI 3 wurde im Dezember 2019 veröffentlicht und führte ein anderes Benennungsschema für Yum-Repository-IDs ein. Abhängig von dem RHEL-7-AMI, aus dem Sie Ihre verwalteten Knoten erstellen, müssen Sie möglicherweise Ihre Befehle aktualisieren. Weitere Informationen finden Sie unter Repository-IDs für RHEL 7 in AWS Haben sich geändert im Red Hat Customer Portal.

- Repo-ID: `rhui-REGION-client-config-server-7/x86_64`

  Repo-Name: `Red Hat Update Infrastructure 2.0 Client Configuration Server 7`

- Repo-ID: `rhui-REGION-rhel-server-releases/7Server/x86_64`

  Repo-Name: `Red Hat Enterprise Linux Server 7 (RPMs)`
- Repo-ID: `rhui-REGION-rhel-server-rh-common/7Server/x86_64`

  Repo-Name: `Red Hat Enterprise Linux Server 7 RH Common (RPMs)`

AlmaLinux, 8, RHEL 8 und Rocky Linux 8

- Repo-ID: `rhel-8-appstream-rhui-rpms`

  Repo-Name: `Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (RPMs)`
- Repo-ID: `rhel-8-baseos-rhui-rpms`

  Repo-Name: `Red Hat Enterprise Linux 8 for x86_64 - BaseOS from RHUI (RPMs)`
- Repo-ID: `rhui-client-config-server-8`

  Repo-Name: `Red Hat Update Infrastructure 3 Client Configuration Server 8`

AlmaLinux 9, RHEL 9 und Rocky Linux 9

- Repo-ID: `rhel-9-appstream-rhui-rpms`

  Repo-Name: `Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMs)`
- Repo-ID: `rhel-9-baseos-rhui-rpms`

  Repo-Name: `Red Hat Enterprise Linux 9 for x86_64 - BaseOS from RHUI (RPMs)`
- Repo-ID: `rhui-client-config-server-9`

  Repo-Name: `Red Hat Enterprise Linux 9 Client Configuration`

SLES

Auf von SUSE Linux Enterprise Server (SLES) verwalteten Knoten erhält die ZYPP-Bibliothek die Liste der verfügbaren Patches (eine Sammlung von Paketen) von den folgenden Standorten:

- Liste der Repositorys: `etc/zypp/repos.d/*`

- Paketinformationen: `/var/cache/zypp/raw/*`

Von SLES verwaltete Knoten verwenden Zypper als Paketmanager und Zypper verwendet das Konzept eines Patches. Ein Patch ist einfach eine Sammlung von Paketen, die ein bestimmtes Problem beheben. Patch Manager behandelt alle Pakete, auf die in einem Patch verwiesen wird, als sicherheitsbezogen. Da einzelnen Paketen weder Klassifizierungen noch Schweregrade zugewiesen werden, weist Patch Manager den Paketen die Attribute des Patches zu, dem sie angehören.

Ubuntu Server

Der Patch-Baseline-Service des Systems Managers auf Ubuntu Server verwendet vorkonfigurierte Repositorys (Repos) auf dem verwalteten Knoten. Diese vorkonfigurierten Repos werden verwendet, um eine aktualisierte Liste der verfügbaren Paket-Upgrades abzurufen. Dazu führt Systems Manager das Äquivalent eines `sudo apt-get update`-Befehls durch.

Pakete werden dann aus *`codename`*`-security`-Repos gefiltert, wobei der Codename für die Release-Version eindeutig ist, z. B. `trusty` für Ubuntu Server 14. Patch Manager identifiziert nur Upgrades, die Teil dieser Repos sind:

- 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 Server23,04 () `lunar-security`

Windows Server

Unter Microsoft Windows-Betriebssystemen ruft Patch Manager eine Liste der verfügbaren, von Microsoft über Microsoft Update veröffentlichten und automatisch auf Windows Server Update Services (WSUS) verfügbaren Updates ab.

Patch Manager überwacht kontinuierlich neue Updates in allen AWS-Region. Die Liste der verfügbaren Updates wird in jeder Region mindestens einmal pro Tag aktualisiert. Wenn die Patch-Informationen von Microsoft verarbeitet werden, entfernt Patch Manager Updates, die durch

spätere Updates ersetzt wurden, aus der Patch-Liste. Daher werden nur die neuesten Updates angezeigt und zur Installation zur Verfügung gestellt. Beispiel: Wenn `KB4012214` `KB3135456` ersetzt, steht nur `KB4012214` als Update in Patch Manager zur Verfügung.

Patch Manager stellt nur dann Patches für Windows Server-Betriebssystemversionen bereit, wenn diese für Patch Manager unterstützt werden. Beispiel: Patch Manager kann nicht zum Patchen von Windows RT verwendet werden.

> **Note**
>
> In einigen Fällen veröffentlicht Microsoft Patches für Anwendungen, die kein Datum und keine Uhrzeit der Aktualisierung angeben. In diesen Fällen wird ein aktualisiertes Datum und eine Uhrzeit von `01/01/1970` standardmäßig angegeben.

So geben Sie ein alternatives Patch-Quell-Repository an (Linux)

Wenn Sie die Standard-Repositorys verwenden, die auf einem verwalteten Knoten für Patch-Operationen konfiguriert sind, sucht Patch Manager, eine Funktion von AWS Systems Manager, nach sicherheitsrelevanten Patches oder installiert sie. Dies ist das Standardverhalten für Patch Manager. Ausführliche Informationen darüber, wie Patch Manager Sicherheits-Patches auswählt und installiert, finden Sie unter Wie Sicherheitspatches ausgewählt werden.

Sie können auf Linux-Systemen jedoch auch mithilfe von Patch Manager Patches installieren, die nicht auf Sicherheit bezogen sind oder die sich in einem anderen als dem Standard-Quell-Repository befinden, das in dem verwalteten Knoten konfiguriert ist. Sie können beim Erstellen einer benutzerdefinierten Patch-Baseline alternative Patch-Quell-Repositorys angeben. Für jede benutzerdefinierte Patch-Baseline können Sie Patch-Quellkonfigurationen für bis zu 20 Versionen eines unterstützten Linux-Betriebssystems angeben.

Beispiel: Angenommen, Ihre Ubuntu Server-Flotte beinhaltet sowohl Ubuntu Server 14.04- als auch Ubuntu Server 16.04-verwaltete Knoten. In diesem Fall können Sie alternative Repositorys für jede Version in derselben benutzerdefinierten Patch-Baseline angeben. Geben Sie für jede Version einen Namen, die Version und den Typ des Betriebssystems (Produkt) und eine Repository-Konfiguration an. Sie können auch ein einziges alternatives Quell-Repository angeben, das für alle Versionen eines unterstützten Betriebssystems gilt.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Patch-Baseline ausführen, die alternative Patch-Repositorys für einen verwalteten Knoten angeben, werden diese Repositorys dadurch nicht zum neuen Standard-Repository auf dem Betriebssystem. Nach Abschluss der Patching-Operation bleiben die zuvor definierten Standard-Repositorys für das Betriebssystem des Knoten als Standard erhalten.

Eine Liste mit Beispielszenarien zur Verwendung dieser Option finden Sie weiter Anwendungsbeispiele für alternative Patch-Quell-Repositorys unten in diesem Thema.

Weitere Informationen zu Standard- und benutzerdefinierten Patch-Baselines finden Sie unter Info zu vordefinierten und benutzerdefinierten Patch-Baselines.

Beispiel: Verwenden der Konsole

Um alternative Patch-Quell-Repositorys anzugeben, wenn Sie in der Systems Manager-Konsole arbeiten, verwenden Sie den Abschnitt Patch sources auf der Seite Create patch baseline. Weitere Informationen zur Verwendung der Optionen in Patch sources (Patch-Quellen) finden Sie unter So erstellen Sie eine benutzerdefinierte Patch-Baseline (Linux).

Beispiel: Verwenden der AWS CLI

Ein Beispiel für die Verwendung der Option `--sources` mit der AWS Command Line Interface (AWS CLI) finden Sie in Erstellen einer Patch-Baseline mit benutzerdefinierten Repositorys für verschiedene Betriebssystemversionen.

Themen

- Wichtige Überlegungen für alternative Repositorys
- Anwendungsbeispiele für alternative Patch-Quell-Repositorys

Wichtige Überlegungen für alternative Repositorys

Beachten Sie die folgenden Punkte, wenn Sie planen, in Ihrer Patching-Strategie alternative Patch-Repositorys zu verwenden.

Nur angegebene Repositorys werden für das Einspielen von Patches verwendet.

Angeben von alternativen Repositorys bedeutet nicht das Angeben zusätzlicher Repositorys. Sie können wählen, andere Repositorys als die auf einem verwalteten Knoten als Standardwerte konfigurierten festzulegen. Sie müssen jedoch auch die Standard-Repositorys als Teil der alternativen Patch-Quell-Konfiguration angeben, wenn Sie möchten, dass deren Updates übernommen werden.

Beispielsweise sind die Standard-Repositorys auf von Amazon Linux 2 verwalteten Knoten `amzn2-core` und `amzn2extra-docker`. Wenn Sie das Repository "Extra Packages for Enterprise Linux (EPEL)" in Ihre Patching-Operationen einschließen möchten, müssen Sie alle drei Repositorys als alternative Repositorys angeben.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Patch-Baseline ausführen, die alternative Patch-Repositorys für einen verwalteten Knoten angeben, werden diese Repositorys dadurch nicht zum neuen Standard-Repository auf dem Betriebssystem. Nach Abschluss der Patching-Operation bleiben die zuvor definierten Standard-Repositorys für das Betriebssystem des Knoten als Standard erhalten.

Das Patching-Verhalten für YUM-basierte Distributionen hängt vom updateinfo.xml-Manifest ab

Wenn Sie alternative Patch-Repositorys für YUM-basierte Distributionen wie Amazon Linux 1 oder Amazon Linux 2, oder CentOS angeben, hängt das Patching-Verhalten davon abRed Hat Enterprise Linux, ob das Repository ein Update-Manifest in Form einer vollständigen und korrekt formatierten `updateinfo.xml` Datei enthält. Diese Datei gibt das Release-Datum, Klassifizierungen und Schweregrade der verschiedenen Pakete an. Jede der folgenden Optionen wirkt sich auf das Patching-Verhalten aus:

- Wenn Sie nach Klassifizierung und Schweregrad filtern, diese aber nicht in `updateinfo.xml` angegeben sind, wird das Paket nicht in Filter aufgenommen. Dies bedeutet auch, dass Pakete ohne eine `updateinfo.xml`-Datei werden nicht in das Patching eingeschlossen werden.
- Wenn Sie nach filternApprovalAfterDays, aber das Veröffentlichungsdatum des Pakets nicht im Unix-Epoch-Format vorliegt (oder kein Veröffentlichungsdatum angegeben ist), wird das Paket nicht in den Filter aufgenommen.
- Eine Ausnahme besteht, wenn Sie das Kontrollkästchen Genehmigte Patches umfassen nicht sicherheitsrelevante Updates auf der Seite Patch-Baseline erstellen aktivieren. In diesem Fall werden Pakete ohne eine `updateinfo.xml`-Datei (oder die diese Datei ohne ordnungsgemäß

formatierte Werte für Klassifizierung, Schweregrad und Datum enthalten) in die vorgefilterte Liste der Patches aufgenommen. (Diese müssen nach wie vor die übrigen Anforderungen Patch-Baseline Regel erfüllen, damit sie installiert werden.)

Anwendungsbeispiele für alternative Patch-Quell-Repositorys

Beispiel 1: Nicht sicherheitsrelevante Updates für Ubuntu Server

Sie verwenden bereits Patch Manager, um Sicherheitspatches auf einer Flotte Ubuntu Server verwalteter Knoten mithilfe der von bereitgestellten vordefinierten Patch AWS-Baseline zu installieren`AWS-UbuntuDefaultPatchBaseline`. Sie können eine neue Patch-Baseline erstellen, die auf diesem Standard basiert, aber in den Genehmigungsregeln angeben, dass nicht auf die Sicherheit bezogene Updates, die Teil der Standardverteilung sind, ebenfalls installiert werden sollen. Wenn diese Patch-Baseline für Ihre Knoten ausgeführt wird, werden sowohl sicherheitsbezogene als auch nicht-sicherheitsbezogene Patches angewendet. Sie können auch auswählen, dass nicht sicherheitsbezogene Patches in den Patch-Ausnahmen, die Sie für eine Baseline angeben, genehmigt werden.

Beispiel 2: Personal Package Archives (PPA) für Ubuntu Server

Auf Ihren von Ubuntu Server verwalteten Knoten wird Software ausgeführt, die über ein Personal Package Archives (PPA) für Ubuntu verteilt wird. In diesem Fall erstellen Sie eine Patch-Baseline, die ein PPA-Repository angibt, das Sie auf dem verwalteten Knoten als das Quell-Repository für die Patch-Operation konfiguriert haben. Führen Sie anschließend mithilfe von Run Command das Patch-Baseline-Dokument auf den Knoten aus.

Beispiel 3: Interne Firmenanwendungen auf Amazon Linux

Sie müssen auf Ihren von Amazon Linux verwalteten Knoten einige Anwendungen ausführen, die für die Compliance gesetzlicher Vorschriften und Branchenstandards erforderlich sind. Sie können ein Repository für diese Anwendungen auf den Knoten konfigurieren, mit YUM die Anwendungen erstmals installieren und eine neue Patch-Baseline aktualisieren oder erstellen, um dieses neue Unternehmens-Repository hinzuzufügen. Anschließend können Sie mit Run Command das Dokument `AWS-RunPatchBaseline` mit der Option `Scan` ausführen, um zu prüfen, ob das Unternehmenspaket unter den installierten Paketen aufgelistet und auf dem verwalteten Knoten aktuell ist. Wenn es nicht aktuell ist, können Sie das Dokument mithilfe der Option `Install` erneut ausführen, um die Anwendungen zu aktualisieren.

Wie Patches installiert werden

Patch Manager, eine Fähigkeit von AWS Systems Manager, verwendet den entsprechenden integrierten Mechanismus für einen Betriebssystemtyp, um Updates auf einem verwalteten Knoten zu installieren. Beispielsweise wird auf Windows Server die Windows Update API und auf Amazon Linux 2 der `yum` Paketmanager verwendet.

Der Rest dieses Abschnitts erläutert, wie Patch Manager Patches auf einem Betriebssystem installiert.

Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, and Amazon Linux 2023

Auf den verwalteten Knoten Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 und Amazon Linux 2023 sieht der Ablauf der Patch-Installation wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.
4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.

5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
7. Die YUM-Update-API (Amazon Linux 1, Amazon Linux 2) oder die DNF-Update-API (Amazon Linux 2022, Amazon Linux 2023) wird wie folgt auf genehmigte Patches angewendet:
   - Für vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, werden nur die in `updateinfo.xml` angegebenen Patches angewendet (nur Sicherheitsupdates). Dies liegt daran, dass das Kontrollkästchen Funktionsupdates einschließen nicht aktiviert ist. Die vordefinierten Baselines entsprechen einer benutzerdefinierten Baseline mit folgenden Eigenschaften:
     - Das Kontrollkästchen Funktionsupdates einschließen ist nicht aktiviert
     - Eine Liste des SCHWEREGRADE von `[Critical, Important]`
     - Eine Liste der KLASSIFIZIERUNGEN von `[Security, Bugfix]`

     Für Amazon Linux 1 und Amazon Linux 2 lautet der entsprechende yum-Befehl für diesen Workflow:

     ```
     sudo yum update-minimal --sec-severity=critical,important --bugfix -y
     ```

     Für Amazon Linux 2022 und Amazon Linux 2023 lautet der entsprechende dnf-Befehl für diesen Workflow:

     ```
     sudo dnf upgrade-minimal --sec-severity=critical --sec-severity=important --
     bugfix -y
     ```

     Wenn das Kontrollkästchen Funktionsupdates einschließen aktiviert ist, werden alle Patches angewendet (Sicherheits- und Funktionsupdates), die in `updateinfo.xml` und nicht in `updateinfo.xml` sind.

     Wenn für Amazon Linux 1 und Amazon Linux 2 eine Baseline mit Include Non-Security-Updates ausgewählt ist, die eine SCHWEREGRAD-Liste `[Critical, Important]` und eine KLASSIFIZIERUNGSLISTE von hat`[Security, Bugfix]`, lautet der entsprechende YUM-Befehl:

```
sudo yum update --security --sec-severity=critical,important --bugfix -y
```

Für Amazon Linux 2022 und Amazon Linux 2023 lautet der entsprechende dnf-Befehl:

```
sudo dnf upgrade --security --sec-severity=critical --sec-severity=important --
bugfix -y
```

> **Note**
>
> Für Amazon Linux 2022 und Amazon Linux 2023 entspricht ein Patch-Schweregrad von `Medium` einem Schweregrad von `Moderate`, der in einigen externen Repositories definiert sein könnte. Wenn Sie Patches mit dem `Medium`-Schweregrad in die Patch-Baseline aufnehmen, werden auch Patches mit dem `Moderate`-Schweregrad von externen Patches auf den Instances installiert.
> Wenn Sie Konformitätsdaten mit der API-Aktion DescribeInstancePatches abfragen, werden beim Filtern nach dem Schweregrad `Medium` Patches mit den Schweregraden `Medium` und `Moderate` gemeldet.
> Amazon Linux 2022 und Amazon Linux 2023 unterstützen auch den Patch-Schweregrad `None`, der vom DNF-Paketmanager erkannt wird.

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

CentOS and CentOS Stream

Auf von CentOS und CentOS Stream verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.

Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.

2. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.
3. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
4. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
5. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
6. Auf genehmigte Patches wird die YUM-Update-API (auf CentOS 6.x und 7.x Versionen) oder das DNF-Update (auf CentOS 8 und CentOS Stream) angewendet.
7. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Debian Server and Raspberry Pi OS

Auf Debian Server und Raspberry Pi OS (früher Rasbian) Instances sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenn eine Aktualisierung für `python3-apt` (eine Python-Bibliotheks-Schnittstelle zu `libapt`) verfügbar ist, wird es auf die neueste Version aktualisiert. (Dieses nicht sicherheitsrelevante Paket wird aktualisiert, auch wenn Sie die Option Mit nicht sicherheitsrelevanten Updates nicht ausgewählt haben.)

   > **Important**
   >
   > Nur auf Debian Server 8: Da einige von Debian Server 8.* verwaltete Knoten auf ein überholtes Paket-Repository (`jessie-backports`) verweisen, führt Patch Manager die folgenden zusätzlichen Schritte aus, um sicherzustellen, dass Patch-Operationen erfolgreich ausgeführt werden:
   >
   > a. Auf Ihrem verwalteten Knoten wird der Verweis auf das Repository `jessie-backports` aus der Liste der Quellspeicherorte (`/etc/apt/sources.list.d/jessie-backports`) auskommentiert. Daher wird nicht versucht, Patches von diesem Speicherort herunterzuladen.
   >
   > b. Ein Signaturschlüssel für Stretch-Sicherheitsupdates wird importiert. Dieser Schlüssel stellt die erforderlichen Berechtigungen für die Aktualisierungs- und Installationsoperationen auf Debian Server 8.*-Distributionen bereit.
   >
   > c. Zu diesem Zeitpunkt wird eine `apt-get`-Operation ausgeführt, um sicherzustellen, dass die neueste Version von `python3-apt` installiert ist, bevor der Patch-Prozess beginnt.
   >
   > d. Wenn die Installation abgeschlossen ist, wird der Verweis auf das Repository `jessie-backports` wiederhergestellt und der Signaturschlüssel wird aus dem Schlüsselbund von APT Sources entfernt. Dies erfolgt, damit die Systemkonfiguration so belassen wird, wie sie vor der Patch-Operation war.
   >
   > Wenn Patch Manager das nächste Mal das System aktualisiert, wird dieser Vorgang wiederholt.

3. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.

4. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   > **Note**
   >
   > Da es nicht möglich ist, die Veröffentlichungstermine von Update-Paketen für Debian Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

   > **Note**
   >
   > Für Debian Server und Raspberry Pi OS sind Patch-Kandidaten-Versionen auf Patches beschränkt, die in `debian-security` enthalten sind.

5. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
6. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
7. Die APT-Bibliothek wird verwendet, um Upgrades für Pakete durchzuführen.

> **Note**
>
> Patch Managerunterstützt nicht die Verwendung der `Pin-Priority` APT-Option, um Paketen Prioritäten zuzuweisen. Patch Manageraggregiert verfügbare Updates aus allen aktivierten Repositorys und wählt das neueste Update aus, das der Baseline für jedes installierte Paket entspricht.

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

macOS

Auf von macOS verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Die `/Library/Receipts/InstallHistory.plist`-Eigenschaft ist ein Datensatz der Software, die mit den `softwareupdate`- und `installer`-Paketmanagern installiert und aktualisiert wurde. Unter Verwendung der`pkgutil`-Befehlszeilen-Tool (für `installer`) und des `softwareupdate`-Paketmanagers werden CLI-Befehle ausgeführt, um diese Liste zu analysieren.

   Für `installer` umfasst die Antwort auf die CLI-Befehle Details zu `package name`, `version`, `volume`, `location` und `install-time`, aber nur der `package name` und die `version` werden von Patch Manager verwendet.

   Für `softwareupdate` umfasst die Antwort auf die CLI-Befehle den Paketnamen (`display name`),`version` und`date`, aber nur der Paketname und die Version werden vom Patch Manager verwendet.

   Für Brew and Brew Cask unterstützt Homebrew seine Befehle, die unter dem Root-Benutzer ausgeführt werden, nicht. Darum fragt Patch Manager entweder als Eigentümer des Homebrew-Verzeichnisses oder als gültiger Benutzer,der zur Eigentümergruppe des Homebrew-Verzeichnisses gehört, Homebrew–Befehle ab und führt sie aus. Die Befehle sind ähnlich wie `softwareupdate` und `installer` und werden durch einen Python-Subprozess ausgeführt, um Paketdaten zu sammeln. Die Ausgabe wird dann analysiert, um Paketnamen und -versionen zu identifizieren.

2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.
4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
7. Ruft die entsprechende Paket-CLI auf dem verwalteten Knoten auf, um genehmigte Patches wie folgt zu verarbeiten:

   > **Note**
   >
   > `installer` fehlt die Funktion, um nach Updates zu suchen und sie zu installieren. Daher meldet Patch Manager für `installer` nur, welche Pakete installiert sind. Das Ergebnis: `installer`-Pakete werden nie als `Missing` gemeldet.

   - Für vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und für benutzerdefinierte Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen nicht ausgewählt wurde, werden nur Sicherheitsupdates angewendet.
   - Bei benutzerdefinierten Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen aktiviert ist, werden sowohl Sicherheits- als auch Funktionsupdates angewendet.
8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Oracle Linux

Auf von Oracle Linux verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.
4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
7. Auf von Version 7 verwalteten Knoten wird die YUM-Update-API wie folgt auf genehmigte Patches angewendet:
   - Für vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und für benutzerdefinierte Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen

nicht ausgewählt wurde, werden nur Patches, die in `updateinfo.xml` angegeben sind (nur Sicherheitsupdates), angewendet.

Der entsprechende Yum-Befehl für diesen Workflow lautet:

```
sudo yum update-minimal --sec-severity=Important,Moderate --bugfix -y
```

- Bei benutzerdefinierten Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen aktiviert ist, werden sowohl Patches aus der Datei `updateinfo.xml` als auch solche, die nicht in `updateinfo.xml` enthalten sind, angewendet (sowohl Sicherheits- als auch Funktionsupdates).

  Der entsprechende Yum-Befehl für diesen Workflow lautet:

  ```
  sudo yum update --security --bugfix -y
  ```

Auf von Version 8 und 9 verwalteten Knoten wird die DNF-Update-API wie folgt auf genehmigte Patches angewendet:

- Für vordefinierte Standard-Patch-Baselines, die von bereitgestellt werden AWS, und für benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Nicht sicherheitsrelevante Updates einbeziehen nicht aktiviert ist, `updateinfo.xml` werden nur die in angegebenen Patches angewendet (nur Sicherheitsupdates).

  Der entsprechende Yum-Befehl für diesen Workflow lautet:

  ```
  sudo dnf upgrade-minimal --security --sec-severity=Moderate --sec-
  severity=Important
  ```

- Bei benutzerdefinierten Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen aktiviert ist, werden sowohl Patches aus der Datei `updateinfo.xml` als auch solche, die nicht in `updateinfo.xml` enthalten sind, angewendet (sowohl Sicherheits- als auch Funktionsupdates).

  Der entsprechende Yum-Befehl für diesen Workflow lautet:

  ```
  sudo dnf upgrade --security --bugfix
  ```

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline`

gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

AlmaLinux, RHEL, and Rocky Linux

Auf AlmaLinux und Rocky Linux verwalteten Knoten Red Hat Enterprise Linux sieht der Arbeitsablauf für die Installation von Patches wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.
4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.

7. Die YUM-Update-API (auf RHEL 7) oder die DNF-Update-API (auf AlmaLinux 8 und 9, RHEL 8 und 9 und Rocky Linux 8 und 9) wird wie folgt auf genehmigte Patches angewendet:

   - Für vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und für benutzerdefinierte Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen nicht ausgewählt wurde, werden nur Patches, die in `updateinfo.xml` angegeben sind (nur Sicherheitsupdates), angewendet.

     Für RHEL 7 lautet der entsprechende YUM-Befehl für diesen Workflow:

     ```
     sudo yum update-minimal --sec-severity=Critical,Important --bugfix -y
     ```

     Für AlmaLinux, RHEL 8 undRocky Linux, lauten die entsprechenden dnf-Befehle für diesen Workflow:

     ```
     sudo dnf update-minimal --sec-severity=Critical --bugfix -y ; \
     sudo dnf update-minimal --sec-severity=Important --bugfix -y
     ```

   - Bei benutzerdefinierten Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen aktiviert ist, werden sowohl Patches aus der Datei `updateinfo.xml` als auch solche, die nicht in `updateinfo.xml` enthalten sind, angewendet (sowohl Sicherheits- als auch Funktionsupdates).

     Für RHEL 7 lautet der entsprechende YUM-Befehl für diesen Workflow:

     ```
     sudo yum update --security --bugfix -y
     ```

     Für AlmaLinux 8 und 9, RHEL 8 und 9 sowie Rocky Linux 8 und 9 lautet der entsprechende dnf-Befehl für diesen Workflow:

     ```
     sudo dnf update --security --bugfix -y
     ```

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

SLES

Auf von SUSE Linux Enterprise Server (SLES) verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.
4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
7. Die Zypper-Update-API wird auf genehmigte Patches angewendet.
8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline`

gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Ubuntu Server

Auf von Ubuntu Server verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenn eine Aktualisierung für `python3-apt` (eine Python-Bibliotheks-Schnittstelle zu `libapt`) verfügbar ist, wird es auf die neueste Version aktualisiert. (Dieses nicht sicherheitsrelevante Paket wird aktualisiert, auch wenn Sie die Option Mit nicht sicherheitsrelevanten Updates nicht ausgewählt haben.)
3. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
4. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

> **Note**
>
> Da es nicht möglich ist, die Veröffentlichungsdaten von Updatepaketen für Ubuntu Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

> **Note**
>
> Für jede Version von Ubuntu Server sind die Patchkandidatenversionen wie folgt auf Patches beschränkt, die Teil des zugehörigen Repositorys für diese Version sind:
>
> - Ubuntu Server 14.04 LTS: `trusty-security`
> - Ubuntu Server 16.04 LTS: `xenial-security`
> - Ubuntu Server 18.04 LTS: `bionic-security`
> - Ubuntu Server20.04 LTS): `focal-security`
> - Ubuntu Server 20.10 STR: `groovy-security`
> - Ubuntu Server22,04 LTS: `jammy-security`
> - Ubuntu Server23,04: `lunar-lobster`

5. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
6. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
7. Die APT-Bibliothek wird verwendet, um Upgrades für Pakete durchzuführen.

> **Note**
>
> Patch Managerunterstützt nicht die Verwendung der `Pin-Priority` APT-Option, um Paketen Prioritäten zuzuweisen. Patch Manageraggregiert verfügbare Updates aus allen aktivierten Repositorys und wählt das neueste Update aus, das der Baseline für jedes installierte Paket entspricht.

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Windows Server

Wenn eine Patch-Operation auf einem von Windows Server verwalteten Knoten durchgeführt wird, fordert die Instance einen Snapshot der entsprechenden Patch-Baseline von Systems Manager an. Dieser Snapshot enthält die Liste aller in der Patch-Baseline verfügbaren Updates, die für die Bereitstellung genehmigt wurden. Diese Liste der Updates wird an die Windows-Update-API gesendet, die festlegt, welche Updates für den verwalteten Knoten zutreffen, und diese bei Bedarf installiert. Wenn Updates installiert sind, wird der verwaltete Knoten danach so oft wie nötig neu gestartet, bis alle erforderlichen Patches abgeschlossen sind. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.) Die Zusammenfassung des Patch-Vorgangs finden Sie in der Ausgabe der Run Command-Anfrage. Zusätzliche Protokolle finden Sie auf dem verwalteten Knoten im `%PROGRAMDATA%\Amazon\PatchBaselineOperations\Logs`-Ordner.

Da die Windows Update-API verwendet wird, um Patches herunterzuladen und zu installieren, müssen alle Gruppenrichtlinieneinstellungen für Windows Update beachtet werden. Für die Verwendung von Patch Manager sind keine Gruppenrichtlinien-Einstellungen erforderlich. Allerdings werden alle definierten Einstellungen angewendet, wie etwa die Weiterleitung von verwalteten Knoten an einen Windows Server Update Services (WSUS)-Server.

> **Note**
>
> Windows lädt standardmäßig alle Patches von der Windows Update-Website von Microsoft herunter, weil Patch Manager die Windows Update-API verwendet, um den Download und die Installation von Patches voranzutreiben. Der verwaltete Knoten muss daher die Website von Microsoft Windows Update erreichen können. Andernfalls tritt ein Fehler bei der Patch-Operation auf. Alternativ können Sie einen WSUS-Server als Patch-Repository konfigurieren und Ihre verwalteten Knoten so konfigurieren, dass sie den WSUS-Server anvisieren, anstatt Gruppenrichtlinien zu verwenden.

Funktionsweise von Patch-Baseline-Regeln auf Linux-basierten Systemen

Die Regeln in einer Patch-Baseline für Linux-Verteilungen funktionieren je nach Verteilungstyp unterschiedlich. Im Gegensatz zu Patch-Updates auf Windows Server verwalteten Knoten werden Regeln auf jedem Knoten ausgewertet, um die konfigurierten Repos auf der Instanz zu berücksichtigen. Patch Manager, eine Funktion von AWS Systems Manager, verwendet den systemeigenen Paketmanager, um die Installation von Patches voranzutreiben, die von der Patch-Baseline genehmigt wurden.

Für Linux-basierte Betriebssysteme, die einen Schweregrad für Patches melden, verwendet Patch Manager den vom Software-Publisher gemeldeten Schweregrad für den Update-Hinweis oder den einzelnen Patch. Patch Manager leitet keinen Schweregrad aus Drittquellen wie dem Common Vulnerability Scoring System (CVSS) oder aus Metriken ab, die von der National Vulnerability Database (NVD) veröffentlicht werden.

Themen

- So funktionieren Patch-Basisregeln auf Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 und Amazon Linux 2023
- Wie Patch-Baseline-Regeln auf CentOS und CentOS Stream funktionieren
- Funktionsweise von Patch-Baseline-Regeln auf Debian Server und Raspberry Pi OS
- Funktionsweise von Patch-Baseline-Regeln auf macOS
- Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux
- So funktionieren Patch-Basisregeln für AlmaLinuxRHEL, und Rocky Linux
- Funktionsweise von Patch-Baseline-Regeln auf SUSE Linux Enterprise Server
- Funktionsweise von Patch-Baseline-Regeln auf Ubuntu Server

So funktionieren Patch-Basisregeln auf Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 und Amazon Linux 2023

Auf Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 und Amazon Linux 2023 läuft der Prozess der Patch-Auswahl wie folgt ab:

1. Auf dem verwalteten Knoten greift die YUM-Bibliothek (Amazon Linux 1 und Amazon Linux 2) oder die DNF-Bibliothek (Amazon Linux 2022 und Amazon Linux 2023) auf die `updateinfo.xml` Datei für jedes konfigurierte Repository zu.

> **Note**
>
> Wenn keine `updateinfo.xml`-Datei gefunden wird, hängt es von den Einstellungen für Funktionsupdates einschließen und Automatische Genehmigung ab, ob Patches installiert werden. Wenn beispielsweise nicht sicherheitsrelevante Updates zulässig sind, werden sie installiert, wenn die automatische Genehmigung eintrifft.

2. Jeder Update-Hinweis in `updateinfo.xml` enthält mehrere Attribute, die die Eigenschaften der Pakete im Hinweis kennzeichnen, wie in der folgenden Tabelle beschrieben.

   Update-Hinweis-Attribute

Attribut	Beschreibung
Typ	Entspricht dem Wert des Klassifizierungsschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Pakets. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation DescribePatchProperties anzeigen. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
severity	Entspricht dem Wert des Schweregradschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der im Update-Hinweis enthaltenen Pakete. Gilt in der Regel nur für Update-Hinweise im Hinblick auf die Sicherheit. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation

Attribut	Beschreibung
	anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
update_id	Kennzeichnet die Advisory ID, wie etwa ALAS-2017-867. Die Advisory ID kann in der Patch-Baseline im ApprovedPatches oder RejectedPatches-Attribut verwendet werden.
Referenzen	Enthält weitere Informationen über den Update-Hinweis, wie etwa eine CVE ID (Format: CVE-2017-1234567). Die CVE ID kann in der Patch-Baseline im ApprovedPatches- oder im RejectedPatches-Attribut verwendet werden.
Aktualisiert	Entspricht ApproveAfterDays in der Patch-Baseline. Kennzeichnet das Veröffentlichungsdatum (Aktualisierungsdatum) der im Update-Hinweis enthaltenen Pakete. Ein Vergleich zwischen dem aktuellen Zeitstempel und dem Wert dieses Attributs plus `ApproveAfterDays` wird verwendet, um zu bestimmen, ob der Patch für die Bereitstellung genehmigt wurde.

> **Note**
>
> Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

3. Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
4. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl
Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerdefinierte Patch-Baselines, bei denen Funktionsupdates einschließen nicht ausgewählt wurde	Für jeden Update-Hinweis in `updateinfo.xml` wird die Patch-Baseline als Filter verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet. Für Amazon Linux 1 und Amazon Linux 2 lautet der entsprechende yum-Befehl für diesen Workflow: `sudo yum update-minimal --security --sec-severity=Critical,Important --bugfix -y` Für Amazon Linux 2022 und Amazon Linux 2023 lautet der entsprechende dnf-Befehl für diesen Workflow: `sudo dnf upgrade-minimal --security --sec-severity=Critical --sec-severity=Important --bugfix -y`
Benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschließen aktiviert ist, mit einer SCHWEREGRAD-Liste von `[Critical, Important]` und einer KLASSIFIZIERUNG-Liste von `[Security, Bugfix]`	Neben den aus `updateinfo.xml` ausgewählten Sicherheits-Updates wendet Patch Manager auch nicht sicherheitsrelevante Updates an, die ansonsten den Patch-Filterregeln entsprechen.

Sicherheitsoption	Patch-Auswahl
	Für Amazon Linux und Amazon Linux 2 lautet der entsprechende YUM-Befehl für diesen Workflow: `sudo yum update-minimal --security --sec-severity=Critical,Important --bugfix -y` Für Amazon Linux 2022 und Amazon Linux 2023 lautet der entsprechende dnf-Befehl für diesen Workflow: `sudo dnf upgrade-minimal --security --sec-severity=Critical --sec-severity=Imporant --bugfix -y`

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten.

Wie Patch-Baseline-Regeln auf CentOS und CentOS Stream funktionieren

Die CentOS- und CentOS Stream Standard-Repositorys enthalten keine Datei. `updateinfo.xml` Benutzerdefinierte Repositorys, die Sie erstellen oder verwenden, können diese Datei jedoch enthalten. In diesem Thema beziehen sich Verweise nur `updateinfo.xml` auf diese benutzerdefinierten Repositorys.

In CentOS und CentOS Stream erfolgt die Patch-Auswahl wie folgt:

1. Auf dem verwalteten Knoten greifen die YUM-Bibliothek (auf CentOS 6.x- und 7.x-Versionen) oder die DNF-Bibliothek (auf CentOS 8.x und CentOS Stream) für jedes konfigurierte Repository auf die `updateinfo.xml` Datei zu, sofern sie in einem benutzerdefinierten Repository vorhanden ist.

   Wenn keine Patches **`updateinfo.xml`** gefunden werden, was immer die Standardrepositorys einschließt, hängt es von den Einstellungen für Nicht sicherheitsrelevante Updates einbeziehen und Automatische Genehmigung ab, ob Patches installiert sind. Wenn beispielsweise nicht sicherheitsrelevante Updates zulässig sind, werden sie installiert, wenn die automatische Genehmigung eintrifft.

2. Falls `updateinfo.xml` vorhanden, enthält jeder Aktualisierungshinweis in der Datei mehrere Attribute, die die Eigenschaften der Pakete im Hinweis angeben, wie in der folgenden Tabelle beschrieben.

   Update-Hinweis-Attribute

Attribut	Beschreibung
Typ	Entspricht dem Wert des Klassifizierungsschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Pakets. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation DescribePatchProperties anzeigen. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
severity	Entspricht dem Wert des Schweregradschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der im Update-Hinweis enthaltenen Pakete. Gilt in der Regel nur für Update-Hinweise im Hinblick auf die Sicherheit. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.

Attribut	Beschreibung
update_id	Kennzeichnet die Advisory ID, wie beispielsweise CVE-2019-17055. Die Advisory ID kann in der Patch-Baseline im ApprovedPatches oder RejectedPatches-Attribut verwendet werden.
Referenzen	Enthält weitere Informationen über den Update-Hinweis, wie beispielsweise eine CVE-ID (Format: CVE-2019-17055) oder eine Bugzilla-ID (Format: 1463241). Die CVE ID und die Bugzilla ID können in der Patch-Baseline im ApprovedPatches- oder im RejectedPatches-Attribut verwendet werden.
Aktualisiert	Entspricht ApproveAfterDays in der Patch-Baseline. Kennzeichnet das Veröffentlichungsdatum (Aktualisierungsdatum) der im Update-Hinweis enthaltenen Pakete. Ein Vergleich zwischen dem aktuellen Zeitstempel und dem Wert dieses Attributs plus `ApproveAfterDays` wird verwendet, um zu bestimmen, ob der Patch für die Bereitstellung genehmigt wurde.

> **Note**
>
> Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

3. In allen Fällen wird das Produkt des verwalteten Knotens durch bestimmtSSM Agent. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
4. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl
Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerdefinierte Patch-Baselines, bei denen Funktionsupdates einschließen nicht ausgewählt wurde	Für jeden Aktualisierungshinweis in `updateinfo.xml` , sofern er in einem benutzerdefinierten Repository vorhanden ist, wird die Patch-Baseline als Filter verwendet , sodass nur die qualifizierten Pakete in das Update aufgenommen werden können. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet. Für CentOS 6 und 7, wo vorhanden `updateinfo.xml` ist, lautet der entsprech ende Yum-Befehl für diesen Workflow: `sudo yum update-minimal --sec-sev erity=Critical,Important --bugfix - y` Für CentOS 8 und CentOS Stream wo vorhanden `updateinfo.xml` ist, lautet der entsprechende dnf-Befehl für diesen Workflow: `sudo dnf upgrade-minimal --sec-sev erity=Critical --sec-severity=Imp ortant --bugfix -y`

Sicherheitsoption	Patch-Auswahl
Benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktions updates einschließen aktiviert ist, mit einer SCHWEREGRAD-Liste von `[Critical, Important]` und einer KLASSIFIZIERUNG-Liste von `[Security, Bugfix]`	Zusätzlich zur Anwendung der Sicherhei tsupdates, aus denen ausgewählt wurde, werden`updateinfo.xml` , sofern diese in einem benutzerdefinierten Repository vorhanden sind, Patch Manager auch nicht sicherheitsrelevante Updates angewendet, die ansonsten den Patch-Filterregeln entsprech en. Für CentOS 6 und 7, wo vorhanden `updateinfo.xml` ist, lautet der entsprech ende Yum-Befehl für diesen Workflow: `sudo yum update --sec-severity=Cri tical,Important --bugfix -y` Für CentOS 8 und CentOS Stream wo vorhanden `updateinfo.xml` ist, lautet der entsprechende dnf-Befehl für diesen Workflow: `sudo dnf upgrade --security --sec-sev erity=Critical --sec-severity=Imp ortant --bugfix -y` Für Standardrepositorys und benutzerd efinierte Repos ohne `updateinfo.xml` diese Option müssen Sie das Kontrollk ästchen Nicht sicherheitsrelevante Updates einbeziehen aktivieren, um Betriebss ystempakete (OS) zu aktualisieren.

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten.

Funktionsweise von Patch-Baseline-Regeln auf Debian Server und Raspberry Pi OS

Auf Debian Server und Raspberry Pi OS (früher Rasbian) bietet der Patch-Baseline-Service Filtern in den Feldern Priorität und Abschnitt an. Diese Felder sind normalerweise für alle Debian Server und Raspberry Pi OS-Pakete vorhanden. Um zu bestimmen, ob ein Patch von der Patch-Baseline ausgewählt wird, geht Patch Manager folgendermaßen vor:

1. Auf Debian Server und Raspberry Pi OS-Systemen wird das Äquivalent von `sudo apt-get update` ausgeführt, um die Liste der verfügbaren Pakete zu aktualisieren. Repos sind nicht konfiguriert und die Daten werden aus Repos abgerufen, die in einer `sources`-Liste konfiguriert sind.
2. Wenn eine Aktualisierung für `python3-apt` (eine Python-Bibliotheks-Schnittstelle zu `libapt`) verfügbar ist, wird es auf die neueste Version aktualisiert. (Dieses nicht sicherheitsrelevante Paket wird aktualisiert, auch wenn Sie die Option Mit nicht sicherheitsrelevanten Updates nicht ausgewählt haben.)

> **Important**
>
> Nur auf Debian Server 8: Da Debian Server 8.*-Betriebssysteme auf ein veraltetes Paket-Repository (`jessie-backports`) verweisen, führt Patch Manager die folgenden zusätzlichen Schritte aus, um sicherzustellen, dass Patch-Operationen erfolgreich ausgeführt werden:
>
> a. Auf Ihrem verwalteten Knoten wird der Verweis auf das Repository `jessie-backports` aus der Liste der Quellspeicherorte (`/etc/apt/sources.list.d/jessie-backports`) auskommentiert. Daher wird nicht versucht, Patches von diesem Speicherort herunterzuladen.
>
> b. Ein Signaturschlüssel für Stretch-Sicherheitsupdates wird importiert. Dieser Schlüssel stellt die erforderlichen Berechtigungen für die Aktualisierungs- und Installationsoperationen auf Debian Server 8.*-Distributionen bereit.
>
> c. Zu diesem Zeitpunkt wird eine `apt-get`-Operation ausgeführt, um sicherzustellen, dass die neueste Version von `python3-apt` installiert ist, bevor der Patch-Prozess beginnt.
>
> d. Wenn die Installation abgeschlossen ist, wird der Verweis auf das Repository `jessie-backports` wiederhergestellt und der Signaturschlüssel wird aus dem Schlüsselbund von APT Sources entfernt. Dies erfolgt, damit die Systemkonfiguration so belassen wird, wie sie vor der Patch-Operation war.

3. Als Nächstes werden die Listen GlobalFilters, ApprovalRules, ApprovedPatches und RejectedPatches angewendet.

> **Note**
>
> Da es nicht möglich ist, die Veröffentlichungstermine von Update-Paketen für Debian Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein. Für Debian Server sind Patch-Kandidaten-Versionen in diesem Fall auf Patches beschränkt, die in den folgenden Repos enthalten sind:

Diese Repos werden wie folgt benannt:

- Debian Server 8: `debian-security jessie`
- Debian Server und Raspberry Pi OS 9: `debian-security stretch`
- Debian Server10: `debian-security buster`
- Debian Server11: `debian-security bullseye`
- Debian Server12: `debian-security bookworm`

Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

> **Note**
>
> Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

Zum Anzeigen der Inhalte der Felder Priorität und Abschnitt führen Sie den folgenden `aptitude`-Befehl aus:

> **Note**
>
> Möglicherweise müssen Sie zuerst Aptitude auf Debian Server-Systemen installieren.

```
aptitude search -F '%p %P %s %t %V#' '~U'
```

In der Antwort auf diesen Befehl werden alle Pakete, für die ein Upgrade durchgeführt werden kann, in diesem Format gemeldet:

```
name, priority, section, archive, candidate version
```

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten.

Funktionsweise von Patch-Baseline-Regeln auf macOS

In macOS erfolgt die Patch-Auswahl folgendermaßen:

1. Auf dem verwalteten Knoten greift Patch Manager auf den geparsten Inhalt der `InstallHistory.plist`-Datei zu und identifiziert Paketnamen und -versionen.

   Details zum Parsing-Prozess finden Sie im Abschnitt macOS in Wie Patches installiert werden.
2. Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
3. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl
Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerdefinierte Patch-Baselines, bei denen Funktionsupdates einschließen nicht ausgewählt wurde	Für jedes verfügbare Paket-Update wird die Patch-Baseline als Filter verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach

Sicherheitsoption	Patch-Auswahl
	Anwenden der Patch-Baseline-Definition verwendet.
Benutzerdefinierte Patch-Baselines, bei denen Funktionsupdates einschließen aktiviert ist	Neben den unter Verwendung von `InstallHistory.plist` identifizierten Sicherheits-Updates wendet Patchmanager auch nicht sicherheitsrelevante Updates an, die ansonsten den Patch-Filterregeln entsprechen.

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten.

Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux

In Oracle Linux erfolgt die Patch-Auswahl folgendermaßen:

1. Auf dem verwalteten Knoten ruft die YUM-Bibliothek die `updateinfo.xml`-Datei für jedes konfigurierte Repo auf.

   > **Note**
   >
   > Die `updateinfo.xml`-Datei ist möglicherweise nicht verfügbar, wenn das Repo nicht von Oracle verwaltet wird. Wenn keine `updateinfo.xml`-Datei gefunden wird, hängt es von den Einstellungen für Funktionsupdates einschließen und Automatische Genehmigung ab, ob Patches installiert werden. Wenn beispielsweise nicht sicherheitsrelevante Updates zulässig sind, werden sie installiert, wenn die automatische Genehmigung eintrifft.

2. Jeder Update-Hinweis in `updateinfo.xml` enthält mehrere Attribute, die die Eigenschaften der Pakete im Hinweis kennzeichnen, wie in der folgenden Tabelle beschrieben.

   Update-Hinweis-Attribute

Attribut	Beschreibung
Typ	Entspricht dem Wert des Klassifizierungsschlüsselattributs im PatchFilter-Datentyp der

Attribut	Beschreibung
	Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Pakets. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
severity	Entspricht dem Wert des Schweregradschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der im Update-Hinweis enthaltenen Pakete. Gilt in der Regel nur für Update-Hinweise im Hinblick auf die Sicherheit. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
update_id	Kennzeichnet die Advisory ID, wie beispielsweise CVE-2019-17055. Die Advisory ID kann in der Patch-Baseline im ApprovedPatches oder RejectedPatches-Attribut verwendet werden.

Attribut	Beschreibung
Referenzen	Enthält weitere Informationen über den Update-Hinweis, wie beispielsweise eine CVE-ID (Format: CVE-2019-17055) oder eine Bugzilla-ID (Format: 1463241). Die CVE ID und die Bugzilla ID können in der Patch-Baseline im ApprovedPatches- oder im RejectedPatches-Attribut verwendet werden.
Aktualisiert	Entspricht ApproveAfterDays in der Patch-Baseline. Kennzeichnet das Veröffentlichungsdatum (Aktualisierungsdatum) der im Update-Hinweis enthaltenen Pakete. Ein Vergleich zwischen dem aktuellen Zeitstempel und dem Wert dieses Attributs plus `ApproveAfterDays` wird verwendet, um zu bestimmen, ob der Patch für die Bereitstellung genehmigt wurde.

> **Note**
>
> Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

3. Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
4. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl
Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerdefinierte Patch-Baselines, bei	Für jeden Update-Hinweis in `updateinfo.xml` wird die Patch-Baseline als Filter verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn

Sicherheitsoption	Patch-Auswahl
denen Funktionsupdates einschließen nicht ausgewählt wurde	mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet. Für von Version 7 verwaltete Knoten lautet der entsprechende Yum-Befehl für diesen Workflow: `sudo yum update-minimal --sec-severity=Important,Moderate --bugfix -y` Für von Version 8 und 9 verwaltete Knoten lautet der entsprechende DNF-Befehl für diesen Workflow: `sudo dnf upgrade-minimal --security --sec-severity=Moderate --sec-severity=Important`

Sicherheitsoption	Patch-Auswahl
Benutzerdefinierte Patch-Baselines, bei denen Funktionsupdates einschließen aktiviert ist, mit einer SCHWEREGRAD-Liste von `[Critical, Important]` und einer KLASSIFIZIERUNG-Liste von `[Security, Bugfix]`	Neben den aus `updateinfo.xml` ausgewählten Sicherheits-Updates wendet Patch Manager auch nicht sicherheitsrelevante Updates an, die ansonsten den Patch-Filterregeln entsprechen. Für von Version 7 verwaltete Knoten lautet der entsprechende Yum-Befehl für diesen Workflow: `sudo yum update --security --sec-severity=Critical,Important --bugfix -y` Für von Version 8 und 9 verwaltete Knoten lautet der entsprechende DNF-Befehl für diesen Workflow: `sudo dnf upgrade --security --sec-severity=Critical, --sec-severity=Important --bugfix y`

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten.

So funktionieren Patch-Basisregeln für AlmaLinuxRHEL, und Rocky Linux

Bei AlmaLinux, Red Hat Enterprise Linux (RHEL) und läuft Rocky Linux der Prozess der Patch-Auswahl wie folgt ab:

1. Auf dem verwalteten Knoten greift die YUM-Bibliothek (RHEL7) oder die DNF-Bibliothek (AlmaLinux 8 und 9, RHEL 8 und 9 und Rocky Linux 8 und 9) auf die `updateinfo.xml` Datei für jedes konfigurierte Repository zu.

> **Note**
>
> Die `updateinfo.xml`-Datei ist möglicherweise nicht verfügbar, wenn das Repo nicht von Red Hat verwaltet wird. Falls keine `updateinfo.xml` gefunden werden, wird kein Patch angewendet.

2. Jeder Update-Hinweis in `updateinfo.xml` enthält mehrere Attribute, die die Eigenschaften der Pakete im Hinweis kennzeichnen, wie in der folgenden Tabelle beschrieben.

   Update-Hinweis-Attribute

Attribut	Beschreibung
Typ	Entspricht dem Wert des Klassifizierungsschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Pakets. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigen. DescribePatchProperties Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
severity	Entspricht dem Wert des Schweregradschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der im Update-Hinweis enthaltenen Pakete. Gilt in der Regel nur für Update-Hinweise im Hinblick auf die Sicherheit. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie

Attribut	Beschreibung
	können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
update_id	Kennzeichnet die Advisory ID, wie etwa RHSA-2017:0864. Die Advisory ID kann in der Patch-Baseline im ApprovedPatches oder RejectedPatches-Attribut verwendet werden.
Referenzen	Enthält weitere Informationen über den Update-Hinweis, wie etwa eine CVE ID (Format: CVE-2017-1000371) oder eine Bugzilla ID (Format: 1463241). Die CVE ID und die Bugzilla ID können in der Patch-Baseline im ApprovedPatches- oder im RejectedPatches-Attribut verwendet werden.
Aktualisiert	Entspricht ApproveAfterDays in der Patch-Baseline. Kennzeichnet das Veröffentlichungsdatum (Aktualisierungsdatum) der im Update-Hinweis enthaltenen Pakete. Ein Vergleich zwischen dem aktuellen Zeitstempel und dem Wert dieses Attributs plus `ApproveAfterDays` wird verwendet, um zu bestimmen, ob der Patch für die Bereitstellung genehmigt wurde.

> **Note**
>
> Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

3. Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
4. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl
Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschließen nicht ausgewählt wurde	Für jeden Update-Hinweis in `updateinfo.xml` wird die Patch-Baseline als Filter verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet. Für RHEL 7 lautet der entsprechende YUM-Befehl für diesen Workflow: `sudo yum update-minimal --security --sec-severity=Critical,Important --bugfix -y` Für AlmaLinux 8 und 9, RHEL 8 und 9 sowie Rocky Linux 8 und 9 lautet der entsprechende dnf-Befehl für diesen Workflow: `sudo dnf upgrade-minimal --security --sec-severity=Critical --sec-severity=Important --bugfix -y`
Benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschließen aktiviert ist, mit einer SCHWEREGRAD-Liste von `[Critical, Important]` und einer KLASSIFIZIERUNG-Liste von `[Security, Bugfix]`	Neben den aus `updateinfo.xml` ausgewählten Sicherheits-Updates wendet Patch Manager auch nicht sicherheitsrelevante Updates an, die ansonsten den Patch-Filterregeln entsprechen. Für RHEL 7 lautet der entsprechende YUM-Befehl für diesen Workflow:

Sicherheitsoption	Patch-Auswahl
	`sudo yum update --security --sec-severity=Critical,Important --bugfix -y` Für AlmaLinux 8 und 9, RHEL 8 und 9 sowie Rocky Linux 8 und 9 lautet der entsprechende dnf-Befehl für diesen Workflow: `sudo dnf upgrade --sec-severity=Critical --sec-severity=Important --bugfix -y`

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten.

Funktionsweise von Patch-Baseline-Regeln auf SUSE Linux Enterprise Server

Auf SLES enthält jeder Patch die folgenden Attribute, mit denen die Eigenschaften der Pakete im Patch gekennzeichnet werden:

- Category: Entspricht dem Wert des Klassifizierungs-Schlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Patches.

  Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation DescribePatchProperties anzeigen. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
- Schweregrad: Entspricht dem Wert des Schweregrads-Schlüsselattributs in dem PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der Patches.

  Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.

Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produkt-Schlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.

Für jeden Patch wird die Patch-Baseline als Filter verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet.

> **Note**
>
> Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

Funktionsweise von Patch-Baseline-Regeln auf Ubuntu Server

Auf Ubuntu Server bietet der Patch-Baseline-Service Filtern in den Feldern Priorität und Abschnitt. Diese Felder sind normalerweise für alle Ubuntu Server-Pakete vorhanden. Um zu bestimmen, ob ein Patch von der Patch-Baseline ausgewählt wird, geht Patch Manager folgendermaßen vor:

1. Auf Ubuntu Server-Systemen wird das Äquivalent von `sudo apt-get update` ausgeführt, um die Liste der verfügbaren Pakete zu aktualisieren. Repos sind nicht konfiguriert und die Daten werden aus Repos abgerufen, die in einer `sources`-Liste konfiguriert sind.
2. Wenn eine Aktualisierung für `python3-apt` (eine Python-Bibliotheks-Schnittstelle zu `libapt`) verfügbar ist, wird es auf die neueste Version aktualisiert. (Dieses nicht sicherheitsrelevante Paket wird aktualisiert, auch wenn Sie die Option Mit nicht sicherheitsrelevanten Updates nicht ausgewählt haben.)
3. Als Nächstes werden die Listen GlobalFilters, ApprovalRules, ApprovedPatches und RejectedPatches angewendet.

   > **Note**
   >
   > Da es nicht möglich ist, die Veröffentlichungsdaten von Updatepaketen für Ubuntu Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein. Für Ubuntu Server sind Patch-Kandidaten-Versionen in diesem Fall auf Patches beschränkt, die in den folgenden Repos enthalten sind:

- 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 Server23,04 () `lunar-security`

Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

> **Note**
>
> Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

Zum Anzeigen der Inhalte der Felder Priorität und Abschnitt führen Sie den folgenden `aptitude`-Befehl aus:

> **Note**
>
> Möglicherweise müssen Sie zuerst Aptitude auf Ubuntu Server 16-Systemen installieren.

```
aptitude search -F '%p %P %s %t %V#' '~U'
```

In der Antwort auf diesen Befehl werden alle Pakete, für die ein Upgrade durchgeführt werden kann, in diesem Format gemeldet:

```
name, priority, section, archive, candidate version
```

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten.

Wichtige Unterschiede zwischen Linux- und Windows-Patching

In diesem Thema werden wichtige Unterschiede zwischen Linux- und Windows-Patching in Patch Manager, einer Funktion von , beschrieben AWS Systems Manager.

> **Note**
>
> Um von Linux verwaltete Knoten zu patchen, müssen Ihre Knoten SSM Agent der Version 2.0.834.0 oder höher ausführen.
> Wenn Systems Manager neue Funktionen hinzugefügt oder Aktualisierungen an den vorhandenen Funktionen vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent auf Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die Seite SSM Agent Versionshinweise auf , GitHub um Benachrichtigungen über SSM Agent Updates zu erhalten.

Unterschied 1: Patch-Bewertung

Linux

Bei Linux-Patches wertet Systems Manager auf jedem verwalteten Knoten einzeln zuerst die Patch-Baseline-Regeln und dann die Liste der genehmigten bzw. abgelehnten Patches aus. Systems Manager muss die Patches auf jedem Knoten gesondert auswerten, weil der Service die Liste der bekannten Patches und Updates von den Repositorys abruft, die auf dem verwalteten Knoten konfiguriert sind.

Windows

Patch Manager verwendet auf Windows-verwalteten Knoten und Linux-verwalteten Knoten jeweils andere Prozesse, um zu ermitteln, welche Patches installiert sein sollten. Für Windows-Patches wertet Systems Manager direkt im Service zuerst die Patch-Baseline-Regeln und dann die Liste der genehmigten bzw. abgelehnten Patches aus. Dies ist möglich, weil Windows-Patches aus einem einzigen Repository (Windows Update) abgerufen werden.

Unterschied 2: **`Not Applicable`**-Patches

Aufgrund der großen Anzahl der verfügbaren Pakete für Linux-Betriebssysteme, meldet Systems Manager keine Details zu Patches mit dem Status Nicht anwendbar. Ein `Not Applicable`-Patch ist beispielsweise ein Patch für Apache-Software, wenn auf der Instance Apache nicht installiert ist. Systems Manager meldet die Anzahl der `Not Applicable` Patches in der Zusammenfassung, aber wenn Sie die DescribeInstancePatches API für einen verwalteten Knoten aufrufen, enthalten die zurückgegebenen Daten keine Patches mit dem Status `Not Applicable`. Dieses Verhalten unterscheidet sich von dem bei Windows.

Unterschied 3: Unterstützung von SSM-Dokumenten

Das Systems-Manager-Dokument (SSM-Dokument) `AWS-ApplyPatchBaseline` unterstützt keine Linux-verwalteten Knoten. Um Patch-Baselines auf Linux-, macOS- und Windows Server-verwalteten Knoten anzuwenden, wird das SSM-Dokument `AWS-RunPatchBaseline` empfohlen. Weitere Informationen finden Sie unter Über SSM-Dokumente für das Patchen von verwalteten Knoten und Informationen über das `AWS-RunPatchBaseline` SSM-Dokument.

Unterschied 4: Anwendungspatches

Der Hauptfokus von Patch Manager liegt auf dem Patchen von Betriebssystemen. Sie können mit Patch Manager jedoch auch Patches für bestimmte Anwendungen auf Ihren verwalteten Knoten anwenden.

Linux

Auf Linux-Betriebssystemen verwendet Patch Manager die konfigurierten Repositorys für Updates und unterscheidet dabei nicht zwischen Betriebssystem- und Anwendungs-Patches. Sie können mit Patch Manager festlegen, aus welchen Repositorys Updates abgerufen werden. Weitere Informationen finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux).

Windows

Auf von Windows Server verwaltete Knoten können Sie für Microsoft-Anwendungen wie Microsoft Word 2016 und Microsoft Exchange Server 2016 Genehmigungsregeln sowie die Patch-

Ausnahmen Approved (Freigegeben) und Rejected (Abgelehnt) anwenden. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines.

Über SSM-Dokumente für das Patchen von verwalteten Knoten

In diesem Thema werden die neun derzeit verfügbaren Systems-Manager-Dokumente (SSM-Dokumente) beschrieben, die Ihnen dabei helfen, Ihre verwalteten Knoten mit den neuesten sicherheitsrelevanten Updates zu patchen.

Wir empfehlen Ihnen, nur fünf dieser Dokumente für Ihre Patches zu verwenden. Zusammen bieten Ihnen diese fünf SSM-Dokumente eine breite Palette an Patch-Optionen mit AWS Systems Manager. Vier dieser Dokumente wurden später veröffentlicht als die vier alten SSM-Dokumente, die sie ersetzen und Erweiterungen oder Konsolidierungen der Funktion darstellen.

Empfohlene SSM-Dokumente für das Patchen

Wir empfehlen, bei Ihren Patch-Vorgängen die folgenden fünf SSM-Dokumente zu verwenden.

- `AWS-ConfigureWindowsUpdate`
- `AWS-InstallWindowsUpdates`
- `AWS-RunPatchBaseline`
- `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineWithHooks`

Ältere SSM-Dokumente zum Patchen

Die folgenden vier älteren SSM-Dokumente können in einigen weiterhin verwendet werden, werden AWS-Regionen jedoch nicht mehr aktualisiert, es kann nicht garantiert werden, dass sie in allen Szenarien funktionieren, und sie werden möglicherweise in future nicht mehr unterstützt. Wir empfehlen, sie nicht bei Ihren Patching-Vorgängen zu verwenden.

- `AWS-ApplyPatchBaseline`
- `AWS-FindWindowsUpdates`
- `AWS-InstallMissingWindowsUpdates`
- `AWS-InstallSpecificWindowsUpdates`

In den folgenden Abschnitten finden Sie weitere Informationen zur Verwendung dieser SSM-Dokumente bei Ihren Patching-Vorgängen.

Themen

- Empfohlene SSM-Dokumente für das Patchen von verwalteten Knoten
- Legacy-SSM-Dokumente für das Patchen von verwalteten Knoten
- Informationen über das AWS-RunPatchBaseline SSM-Dokument
- Informationen über das AWS-RunPatchBaselineAssociation SSM-Dokument
- Informationen über das AWS-RunPatchBaselineWithHooks SSM-Dokument
- Beispielszenario für die Verwendung des Parameters „InstallOverrideList“ in AWS-RunPatchBaseline oder AWS-RunPatchBaselineAssociation
- Verwenden des BaselineOverride Parameters

Empfohlene SSM-Dokumente für das Patchen von verwalteten Knoten

Die folgenden fünf SSM-Dokumente werden für die Verwendung bei Ihren Patching-Operationen für Ihre verwalteten Knoten empfohlen.

Empfohlene SSM-Dokumente

- AWS-ConfigureWindowsUpdate
- AWS-InstallWindowsUpdates
- AWS-RunPatchBaseline
- AWS-RunPatchBaselineAssociation
- AWS-RunPatchBaselineWithHooks

`AWS-ConfigureWindowsUpdate`

Unterstützt die Konfiguration grundlegender Funktionen für Windows Update und deren Verwendung zum automatischen Installieren von Updates (oder zum Deaktivieren automatischer Updates). In allen AWS-Regionen verfügbar.

Mit diesem SSM-Dokument wird Windows Update aufgefordert, die angegebenen Updates herunterzuladen und zu installieren und die verwalteten Knoten bei Bedarf neu zu starten. Verwenden Sie dieses Dokument mit State Manager einer Funktion von AWS Systems Manager, um

sicherzustellen, dass Windows Update seine Konfiguration beibehält. Sie können es auch manuell mit Run Command, einer Funktion von AWS Systems Manager, ausführen, um die Konfiguration von Windows Update zu ändern.

Die in diesem Dokument verfügbaren Parameter unterstützen die Angabe einer Kategorie von Updates, die installiert werden sollen (oder ob automatische Updates deaktiviert werden sollen), sowie die Angabe des Wochentages und der Tageszeit für die Ausführung von Patch-Vorgängen. Dieses SSM-Dokument ist besonders dann von Vorteil, wenn Sie keine strenge Kontrolle über Windows Updates benötigen und keine Compliance-Informationen sammeln müssen.

Replaces legacy SSM documents:

- Keine

`AWS-InstallWindowsUpdates`

Installiert Updates auf einem von Windows Server verwalteten Knoten. In allen AWS-Regionen verfügbar.

Dieses SSM-Dokument bietet grundlegende Patch-Funktion für den Fall, dass Sie entweder ein bestimmtes Update (mit Hilfe des `Include Kbs`-Parameters) installieren möchten oder Patches mit bestimmten Klassifizierungen oder Kategorien installieren möchten, aber keine Informationen zur Patch-Compliance benötigen.

Replaces legacy SSM documents:

- `AWS-FindWindowsUpdates`
- `AWS-InstallMissingWindowsUpdates`
- `AWS-InstallSpecificWindowsUpdates`

Die drei alten Dokumente erfüllen zwar unterschiedliche Funktionen, aber Sie können die gleichen Ergebnisse erzielen, indem Sie unterschiedliche Parametereinstellungen mit dem neueren SSM-Dokument `AWS-InstallWindowsUpdates` verwenden. Diese Parametereinstellungen werden in Legacy-SSM-Dokumente für das Patchen von verwalteten Knoten beschrieben.

`AWS-RunPatchBaseline`

Installiert Patches auf Ihren verwalteten Knoten oder scannt Knoten, um festzustellen, ob qualifizierte Patches fehlen. In allen AWS-Regionen verfügbar.

Mit AWS-RunPatchBaseline können Sie Patch-Genehmigungen mithilfe der Patch-Baseline steuern, die als „Standard“ für einen Betriebssystemtyp angegeben ist. Stellt Informationen zur Patch-Compliance dar, die Sie mit den Systems Manager-Compliance-Tools einsehen können. Mit diesen Tools erhalten Sie Erkenntnisse in den Zustand der Patch-Compliance Ihrer verwalteten Knoten, z. B. bei welchen Knoten Patches fehlen und was diese Patches sind. Wenn Sie AWS-RunPatchBaseline verwenden, werden Patch-Compliance-Informationen mit dem API-Befehl PutInventory aufgezeichnet. Für Linux-Betriebssysteme werden Compliance-Informationen für Patches sowohl über das in einem verwalteten Knoten konfigurierte Standard-Quell-Repository bereitgestellt, als auch von einem beliebigen alternativen Quell-Repository aus, das Sie in einer benutzerdefinierten Patch-Baseline angeben. Weitere Informationen über alternative Quell-Repositorys finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux). Weitere Informationen zu den Systems Manager-Compliance-Tools finden Sie unter AWS Systems Manager-Compliance.

Ersetzt alte Dokumente:

- AWS-ApplyPatchBaseline

Das Legacy-Dokument AWS-ApplyPatchBaseline gilt nur für von Windows Server verwaltete Knoten und bietet keinen Support für Anwendungs-Patches. Das neue Dokument AWS-RunPatchBaseline bietet die gleiche Unterstützung für sowohl Windows- als auch Linux-Systeme. Version 2.0.834.0 oder höher von SSM Agent ist erforderlich, um das Dokument AWS-RunPatchBaseline zu verwenden.

Weitere Informationen über das SSM-Dokument AWS-RunPatchBaseline finden Sie unter Informationen über das AWS-RunPatchBaseline SSM-Dokument.

AWS-RunPatchBaselineAssociation

Installiert Patches auf Ihren Instances oder scannt Instances, um festzustellen, ob qualifizierte Patches fehlen. In allen kommerziellen AWS-Regionen verfügbar.

AWS-RunPatchBaselineAssociation unterscheidet sich in verschiedenen wichtigen Punkten von AWS-RunPatchBaseline:

- AWS-RunPatchBaselineAssociationist in erster Linie für die Verwendung mit State Manager Zuordnungen vorgesehenQuick Setup, die mit einer Funktion von erstellt wurden AWS Systems Manager. Insbesondere, wenn Sie den Quick Setup-Host-Management-Konfigurationstyp verwenden und die Option Scan instances for missing patches daily (Instances täglich auf fehlende

Patches scannen) auswählen, verwendet das System `AWS-RunPatchBaselineAssociation` für diese Operation.

In den meisten Fällen sollten Sie jedoch beim Einrichten eigener Patching-Vorgänge `AWS-RunPatchBaseline` oder `AWS-RunPatchBaselineWithHooks` anstelle von `AWS-RunPatchBaselineAssociation` auswählen.

Weitere Informationen finden Sie unter den folgenden Themen:

- AWS Systems Manager Quick Setup
- Informationen über das `AWS-RunPatchBaselineAssociation` SSM-Dokument

- `AWS-RunPatchBaselineAssociation` unterstützt die Verwendung von Tags, um zu identifizieren, welche Patch-Baseline bei der Ausführung mit einer Reihe von Zielen verwendet werden soll.
- Für Patching-Vorgänge, die `AWS-RunPatchBaselineAssociation` verwenden, werden Patch-Compliance-Daten in Bezug auf eine bestimmte State Manager-Zuordnung gesammelt. Die Patch-Compliance-Daten, die gesammelt werden, wenn `AWS-RunPatchBaselineAssociation` ausgeführt wird, werden mit dem API-Befehl `PutComplianceItems` anstelle des Befehls `PutInventory` aufgezeichnet. Dies verhindert, dass Compliance-Daten, die nicht mit dieser bestimmten Zuordnung verknüpft sind, überschrieben werden.

  Für Linux-Betriebssysteme werden Compliance-Informationen für Patches sowohl über das in einer Instance konfigurierte Standard-Quell-Repository bereitgestellt, als auch von einem beliebigen alternativen Quell-Repository aus, das Sie in einer benutzerdefinierten Patch-Baseline angeben. Weitere Informationen über alternative Quell-Repositorys finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux). Weitere Informationen zu den Systems Manager-Compliance-Tools finden Sie unter AWS Systems Manager-Compliance.

Ersetzt alte Dokumente:

- Keine

Weitere Informationen über das SSM-Dokument `AWS-RunPatchBaselineAssociation` finden Sie unter Informationen über das `AWS-RunPatchBaselineAssociation` SSM-Dokument.

`AWS-RunPatchBaselineWithHooks`

Installiert Patches auf Ihren verwalteten Knoten oder scannt Knoten, um festzustellen, ob qualifizierte Patches fehlen. Mit optionalen Hooks können Sie SSM-Dokumente an drei Punkten während des Patch-Zyklus ausführen. In allen kommerziellen AWS-Regionen verfügbar.

`AWS-RunPatchBaselineWithHooks` unterscheidet sich von `AWS-RunPatchBaseline` in seiner `Install`-Operation.

`AWS-RunPatchBaselineWithHooks` unterstützt Lebenszyklus-Hooks, die während dem Patching von verwalteten Knoten an festgelegten Punkten ausgeführt werden. Da Patch-Installationen manchmal den Neustart von verwalteten Knoten erfordern, ist die Patch-Operation in zwei Ereignisse unterteilt, wobei insgesamt drei Hooks enthalten sind, die benutzerdefinierte Funktion unterstützen. Der erste Hook ist vor der `Install with NoReboot`-Operation. Der zweite Hook ist nach der `Install with NoReboot`-Operation. Der dritte Hook ist nach dem Neustart des Knoten verfügbar.

Ersetzt alte Dokumente:

- Keine

Weitere Informationen über das SSM-Dokument `AWS-RunPatchBaselineWithHooks` finden Sie unter Informationen über das `AWS-RunPatchBaselineWithHooks` SSM-Dokument.

Legacy-SSM-Dokumente für das Patchen von verwalteten Knoten

Die folgenden vier SSM-Dokumente sind in einigen AWS-Regionen noch verfügbar. Sie werden jedoch nicht mehr aktualisiert und werden möglicherweise in future nicht mehr unterstützt, weshalb wir ihre Verwendung nicht empfehlen. Stattdessen verwenden Sie bitte die unter Empfohlene SSM-Dokumente für das Patchen von verwalteten Knoten beschriebenen Dokumente.

Alte SSM-Dokumente

- AWS-ApplyPatchBaseline
- AWS-FindWindowsUpdates
- AWS-InstallMissingWindowsUpdates
- AWS-InstallSpecificWindowsUpdates

AWS-ApplyPatchBaseline

Unterstützt nur von Windows Server verwaltete Knoten, enthält jedoch, anders als der Nachfolger `AWS-RunPatchBaseline`, keinen Support für Anwendungs-Patches. Nicht verfügbar, wenn sie nach August 2017 AWS-Regionen veröffentlicht wurden.

> **Note**
>
> Um dieses SSM-Dokument, `AWS-RunPatchBaseline`, zu ersetzen, wird die Version 2.0.834.0 oder eine neuere Version von SSM Agent benötigt. Sie können das Dokument `AWS-UpdateSSMAgent` verwenden, um Ihre verwalteten Knoten auf die neueste Version des Agenten zu aktualisieren.

AWS-FindWindowsUpdates

Ersetzt durch `AWS-InstallWindowsUpdates`, die alle die gleichen Aktionen ausführen können. Nicht verfügbar bei AWS-Regionen Markteinführungen nach April 2017.

Um das gleiche Ergebnis wie bei diesem alten SSM-Dokument zu erzielen, verwenden Sie die folgende Parameterkonfiguration mit dem empfohlenen Ersatzdokument, `AWS-InstallWindowsUpdates`:

- `Action` = `Scan`
- `Allow Reboot` = `False`

AWS-InstallMissingWindowsUpdates

Ersetzt durch `AWS-InstallWindowsUpdates`, die alle die gleichen Aktionen ausführen können. Nicht verfügbar bei Produkten, die nach April 2017 auf den AWS-Regionen Markt gebracht wurden.

Um das gleiche Ergebnis wie bei diesem alten SSM-Dokument zu erzielen, verwenden Sie die folgende Parameterkonfiguration mit dem empfohlenen Ersatzdokument, `AWS-InstallWindowsUpdates`:

- `Action` = `Install`
- `Allow Reboot` = `True`

`AWS-InstallSpecificWindowsUpdates`

Ersetzt durch `AWS-InstallWindowsUpdates`, die alle die gleichen Aktionen ausführen können. Nicht verfügbar bei Produkten, die nach April 2017 auf den AWS-Regionen Markt gebracht wurden.

Um das gleiche Ergebnis wie bei diesem alten SSM-Dokument zu erzielen, verwenden Sie die folgende Parameterkonfiguration mit dem empfohlenen Ersatzdokument, `AWS-InstallWindowsUpdates`:

- `Action` = `Install`
- `Allow Reboot` = `True`
- `Include Kbs` = *`durch Komma getrennte Liste der KB-Artikel`*

Informationen über das **`AWS-RunPatchBaseline`** SSM-Dokument

AWS Systems Manager unterstütztAWS-RunPatchBaseline, ein Systems Manager Manager-Dokument (SSM-Dokument) fürPatch Manager, eine Fähigkeit von AWS Systems Manager. Dieses SSM-Dokument führt Patch-Operationen auf verwaltete Knoten sowohl für sicherheitsrelevante als auch für andere Arten von Updates durch. Wenn das Dokument ausgeführt wird, verwendet es die Patch-Baseline, die der „Standard“ für einen Betriebssystemtyp ist, wenn keine Patch-Gruppe angegeben ist. Andernfalls wird die Patch-Baseline verwendet, die der Patch-Gruppe zugeordnet ist. Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

Sie können das Dokument `AWS-RunPatchBaseline` verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows Server ist der Anwendungssupport auf Updates für Microsoft-Anwendungen beschränkt.)

Dieses Dokument unterstützt von Linux, macOS und Windows Server verwaltete Knoten. Das Dokument führt die entsprechenden Aktionen für jede Plattform durch.

> **Note**
>
> Patch Manager unterstützt auch das veraltete SSM-Dokument `AWS-ApplyPatchBaseline`. Dieses Dokument unterstützt jedoch nur das Patchen von Windows-verwalteten Knoten. Wir empfehlen Ihnen, stattdessen `AWS-RunPatchBaseline` zu verwenden, da es das Patchen auf Linux-, macOS- und Windows Server-verwaltete Knoten unterstützt. Version 2.0.834.0 oder höher von SSM Agent ist erforderlich, um das Dokument `AWS-RunPatchBaseline` zu verwenden.

Windows Server

Auf Windows Server verwalteten Knoten lädt das `AWS-RunPatchBaseline` Dokument ein PowerShell Modul herunter und ruft es auf, das wiederum einen Snapshot der Patch-Baseline herunterlädt, die für den verwalteten Knoten gilt. Dieser Patch-Baseline-Snapshot enthält eine Liste genehmigter Patches, die kompiliert werden, indem die Patch-Baseline auf einem WSUS-Server (Windows Server Update Services) abgefragt wird. Diese Liste wird an die Windows Update-API weitergeleitet, die das Herunterladen und Installieren des genehmigten Patches entsprechend steuert.

Linux

Auf Linux-verwalteten Knoten ruft das Dokument `AWS-RunPatchBaseline` ein Python-Modul auf, das wiederum einen entsprechenden Snapshot der Patch-Baseline für den verwalteten Knoten herunterlädt. Dieser Patch-Baseline-Snapshot verwendet die definierten Regeln und Listen der genehmigten und gesperrten Patches, um den entsprechenden Paketmanager für jeden Knoten-Typ anzutreiben:

- Die verwalteten Knoten Amazon Linux 1, Amazon Linux 2Oracle Linux, CentOS und RHEL 7 verwenden YUM. Für YUM-Vorgänge erfordert Patch Manager `Python 2.6` oder eine höhere unterstützte Version (2.6–3.10).
- Von RHEL 8 verwaltete Knoten verwenden DNF. Für DNF-Vorgänge erfordert Patch Manager eine unterstützte Version von `Python 2` oder `Python 3` (2.6–3.10). (Keine der beiden Versionen ist standardmäßig auf RHEL 8 installiert. Sie müssen die eine oder andere Version manuell installieren.)
- Debian Server, Raspberry Pi OS und Ubuntu Server-Instances verwenden APT. Für APT-Vorgänge erfordert Patch Manager eine unterstützte Version von `Python 3` (3.0–3.10).
- Von SUSE Linux Enterprise Server verwaltete Knoten verwenden Zypper. Für Zypper-Vorgänge erfordert Patch Manager `Python 2.6` oder eine höhere unterstützte Version (2.6–3.10).

macOS

Auf macOS-verwalteten Knoten ruft das Dokument `AWS-RunPatchBaseline` ein Python-Modul auf, das wiederum einen entsprechenden Snapshot der Patch-Baseline für den verwalteten Knoten herunterlädt. Als Nächstes ruft ein Python-Unterprozess die AWS Command Line Interface (AWS CLI) auf dem Knoten auf, um die Installations- und Aktualisierungsinformationen für die angegebenen Paketmanager abzurufen und den entsprechenden Paketmanager für jedes Aktualisierungspaket zu steuern.

Jeder Snapshot ist spezifisch für eine Patchgruppe AWS-Konto, ein Betriebssystem und eine Snapshot-ID. Der Snapshot wird über eine vorsignierte Amazon Simple Storage Service (Amazon S3)-URL bereitgestellt, die 24 Stunden nach Erstellung des Snapshots abläuft. Wenn Sie jedoch denselben Snapshot-Inhalt auf andere verwaltete Knoten anwenden möchten, können Sie nach Ablauf der URL bis zu drei Tage nach Erstellung des Snapshots eine neue vorsignierte Amazon-S3-URL generieren. Verwenden Sie dazu den Befehl get-deployable-patch-snapshot-for-instance.

Nachdem alle genehmigten und zutreffenden Updates installiert und je nach Bedarf Neustarts durchgeführt wurden, werden Patch-Compliance-Informationen auf einem verwalteten Knoten generiert und wieder an Patch Manager gemeldet.

> **Note**
>
> Wenn der Parameter `RebootOption` im Dokument `AWS-RunPatchBaseline` auf `NoReboot` gesetzt ist, wird der verwaltete Knoten nach dem Ausführen von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.

Weitere Informationen zum Anzeigen von Patch-Compliance-Daten finden Sie unter Info zu Patch Compliance.

`AWS-RunPatchBaseline` parameters

`AWS-RunPatchBaseline` unterstützt fünf Parameter. Der Parameter `Operation` muss angegeben werden. Die Parameter `InstallOverrideList`, `BaselineOverride` und `RebootOption` sind optional. `Snapshot-ID` ist technisch optional, wir empfehlen allerdings, dass Sie einen benutzerdefinierten Wert dafür angeben, wenn Sie `AWS-RunPatchBaseline` außerhalb von einem Wartungsfenster ausführen, und Patch Manager den Wert benutzerdefinierten automatisch angeben lassen, wenn das Dokument als Teil eines Wartungsfenstervorgangs ausgeführt wird.

Parameter

- Parametername: Operation
- Parametername: AssociationId
- Parametername: Snapshot ID
- Parametername: InstallOverrideList
- Parametername: RebootOption
- Parametername: BaselineOverride

Parametername: **`Operation`**

Nutzung: erforderlich.

Optionen: `Scan` | `Install`.

Scan

Wenn Sie die Option `Scan` wählen, bestimmt `AWS-RunPatchBaseline` den Patch-Compliance-Status des verwalteten Knoten und meldet diese Informationen an Patch Manager. `Scan` fordert nicht zum Installieren von Updates oder zum Neustarten von verwalteten Knoten auf. Stattdessen erkennt die Operation, wo für den Knoten genehmigte und geeignete Updates fehlen.

Installieren

Bei Auswahl der Option `Install` versucht `AWS-RunPatchBaseline`, die genehmigten und geeigneten Updates zu installieren, die auf dem verwalteten Knoten fehlen. Patch-Compliance-Informationen, die als Teil eines `Install`-Vorgangs generiert wurden, enthalten keine fehlenden Updates, melden allerdings möglicherweise Updates im Fehlerzustand, wenn die Installation des Updates aus einem beliebigen Grund nicht erfolgreich war. Immer wenn ein Update auf einem verwalteten Knoten installiert wird, wird der Knoten neu gestartet, um sicherzustellen, dass das Update installiert und aktiviert ist. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

> **Note**
>
> Wenn ein von den Basisregeln festgelegter Patch installiert wird, bevor der Patch Manager den verwalteten Knoten aktualisiert, wird das System möglicherweise nicht wie erwartet neu gestartet. Dies kann passieren, wenn ein Patch manuell von einem Benutzer oder automatisch von einem anderen Programm, z. B. dem `unattended-upgrades`-Paket auf Ubuntu Server, installiert wird.

Parametername: **`AssociationId`**

Nutzung: optional.

`AssociationId` ist die ID einer vorhandenen Zuordnung in State Manager, einer Funktion von AWS Systems Manager. Es wird von Patch Manager verwendet, um einer angegebenen Zuordnung

Compliance-Daten hinzuzufügen. Diese Zuordnung bezieht sich auf einen Patching-Vorgang, der in einer Patch-Richtlinie in Quick Setup eingerichtet ist.

> **Note**
>
> Wenn mit der `AWS-RunPatchBaseline` ein `AssociationId`-Wert zusammen mit einer Baseline-Überschreibung der Patch-Richtlinie bereitgestellt wird, wird das Patchen als eine `PatchPolicy`-Operation durchgeführt und der `ExecutionType`-Wert, der in `AWS:ComplianceItem` gemeldet wird, ist ebenfalls `PatchPolicy`. Wenn kein `AssociationId`-Wert angegeben wird, wird das Patchen als eine `Command`-Operation durchgeführt, und der `ExecutionType`-Wert, der in `AWS:ComplianceItem` übermittelt wird, ist ebenfalls `Command`.

Wenn Sie noch keine Zuordnung erstellt haben, die Sie verwenden möchten, können Sie eine erstellen, indem Sie den Befehl create-association ausführen.

Parametername: **`Snapshot ID`**

Nutzung: optional.

`Snapshot ID` ist eine eindeutige ID (GUID), die von Patch Manager verwendet wird, um sicherzustellen, dass ein Satz von verwalteten Knoten, für die in einer einzelnen Operation Patches durchgeführt werden, den genau gleichen Satz genehmigter Patches aufweist. Auch wenn der Parameter als optional definiert ist, hängen unsere Empfehlungen für bewährte Methoden davon ab, ob Sie `AWS-RunPatchBaseline` in einem Wartungsfenster, wie in der folgenden Tabelle beschrieben, ausführen.

Bewährte Methoden für **`AWS-RunPatchBaseline`**

Mode	Bewährte Methode	Details
Ausführen von `AWS-RunPatchBaseline` innerhalb eines Wartungsfensters	Geben Sie keine Snapshot ID an. Patch Manager wird sie für Sie bereitstellen.	Falls Sie ein Wartungsfenster zum Ausführen von `AWS-RunPatchBaseline` verwenden, dürfen Sie Ihre eigene generierte Snapshot ID nicht angeben. In diesem Szenario stellt

Mode	Bewährte Methode	Details
		Systems Manager einen GUID-Wert auf Grundlage der Wartungsfensterausführungs-ID bereit. Auf diese Weise wird sichergestellt, dass eine richtige ID für alle Aufrufe von `AWS-RunPatchBaseline` in diesem Wartungsfenster verwendet wird. Wenn Sie einen Wert in diesem Szenario angeben, beachten Sie, dass der Snapshot der Patch-Baseline möglicherweise nicht länger als drei Tagen erhalten bleibt. Danach wird ein neuer Snapshot erstellt, auch wenn Sie dieselbe ID angeben, nachdem der Snapshot abgelaufen ist.

Mode	Bewährte Methode	Details
Ausführen von `AWS-RunPatchBaseline` außerhalb eines Wartungsfensters	Generieren Sie einen benutzerdefinierten GUID-Wert für die Snapshot-ID und geben Sie ihn an.[1]	Wenn Sie kein Wartungsfenster zum Ausführen von `AWS-RunPatchBaseline` verwenden, empfehlen wir, dass Sie eine eindeutige Snapshot-ID für jede Patch-Baseline generieren und angeben, insbesondere wenn Sie das Dokument `AWS-RunPatchBaseline` auf mehreren verwaltete Knoten in derselben Operation ausführen. Wenn Sie keine ID in diesem Szenario angeben, generiert Systems Manager eine andere Snapshot-ID für jeden verwalteten Knoten, an den der Befehl gesendet wird. Dies kann zu unterschiedlichen Sätzen von Patches führen, die auf den jeweiligen verwalteten Knoten angegeben sind. Zum Beispiel: Angenommen, Sie führen das Dokument `AWS-RunPatchBaseline` direkt über Run Command, eine Funktion von AWS Systems Manager, aus und richten es auf eine Gruppe von 50 verwalteten Knoten aus. Das Angeben einer benutzerdefinierten Snapshot-ID führt zur Erstellung eines

Mode	Bewährte Methode	Details
		einzelnen Baseline-Snapshots , der verwendet wird, um alle Knoten zu bewerten und zu patchen. Dadurch wird gewährleistet, dass sie letztendlich einen konsistenten Zustand aufweisen.

[1] Sie können jedes beliebige Tool zum Generieren eines Werts für den Snapshot-ID-Parameter verwenden, das eine GUID generieren kann. In können Sie PowerShell beispielsweise das `New-Guid` Cmdlet verwenden, um eine GUID im Format von zu generieren. `12345699-9405-4f69-bc5e-9315aEXAMPLE`

Parametername: **`InstallOverrideList`**

Nutzung: optional.

Mit `InstallOverrideList` können Sie eine https-URL oder eine Amazon S3-PathStyle-URL zu einer Liste mit zu installierenden Patches angeben. Diese im YAML-Format geführte Patch-Installationsliste überschreibt die von der Standard-Patch-Baseline angegebenen Patches. Dies bietet Ihnen eine detalliertere Kontrolle darüber, welche Patches auf Ihren verwalteten Knoten installiert sind.

Das Verhalten des Patchvorgangs bei Verwendung des `InstallOverrideList` Parameters unterscheidet sich zwischen Linux- und macOS verwalteten Knoten und verwalteten Knoten. Windows Server Unter Linux & Patch Manager versuchtmacOS, in der `InstallOverrideList` Patch-Liste enthaltene Patches anzuwenden, die in einem beliebigen Repository vorhanden sind, das auf dem Knoten aktiviert ist, unabhängig davon, ob die Patches den Patch-Basisregeln entsprechen oder nicht. Auf Windows Server Knoten werden Patches in der `InstallOverrideList` Patch-Liste jedoch nur angewendet, wenn sie auch den Patch-Baseline-Regeln entsprechen.

Beachten Sie, dass Compliance-Berichte Patch-Status entsprechend den Angaben in der Patch-Baseline wiedergeben, nicht entsprechend Ihren Angaben in einer `InstallOverrideList`-Liste von Patches. Mit anderen Worten: Scan-Operationen ignorieren den Parameter `InstallOverrideList`. Auf diese Weise wird sichergestellt, dass Compliance-Berichte den Patch-

Status konsistent entsprechend der Richtlinie wiedergeben und nicht danach, was für eine bestimmte Patching-Operation genehmigt wurde.

Eine Beschreibung, wie Sie den Parameter `InstallOverrideList` verwenden können, um verschiedene Patch-Typen in verschiedenen Wartungsfenster-Zeitplänen auf eine Zielgruppe anzuwenden und gleichzeitig eine einzelne Patch-Baseline zu verwenden, finden Sie unter Beispielszenario für die Verwendung des Parameters „InstallOverrideList“ in `AWS-RunPatchBaseline` oder `AWS-RunPatchBaselineAssociation`.

Gültige URL-Formate

> **Note**
>
> Wenn Ihre Datei in einem öffentlich zugänglichen Bucket gespeichert ist, können Sie entweder ein HTTPS-URL-Format oder eine URL im Amazon S3-Pfadstil angeben. Wenn Ihre Datei in einem privaten Bucket gespeichert ist, müssen Sie eine URL im Amazon S3-Pfadstil angeben.

- HTTPS-URL-Format:

```
https://s3.aws-api-domain/DOC-EXAMPLE-BUCKET/my-windows-override-list.yaml
```

- URL im Amazon S3-Pfadstil:

```
s3://DOC-EXAMPLE-BUCKET/my-windows-override-list.yaml
```

Gültige YAML-Inhaltsformate

Die Formate, die Sie verwenden, um Patches in Ihrer Liste anzugeben, hängen von dem Betriebssystem Ihres verwalteten Knoten ab. Das allgemeine Format lautet jedoch folgendermaßen:

```
patches:
    -
        id: '{patch-d}'
        title: '{patch-title}'
        {additional-fields}:{values}
```

Sie können zwar zusätzliche Felder in der YAML-Datei bereitstellen, diese werden jedoch während der Patch-Operationen ignoriert.

Darüber hinaus empfehlen wir zu überprüfen, ob das Format Ihrer YAML-Datei gültig ist, bevor Sie die Liste in Ihrem S3-Bucket hinzufügen oder aktualisieren. Weitere Informationen zum YAML-Format finden Sie unter yaml.org. Für Validierungstool-Optionen suchen Sie im Internet nach "yaml format validators" durch.

Linux

id

Das Feld id ist ein Pflichtfeld. Verwenden Sie es, um Patches mit Paketnamen und Architektur anzugeben. Zum Beispiel: `'dhclient.x86_64'`. Sie können Platzhalter in der ID verwenden, um mehrere Pakete anzugeben. Zum Beispiel `'dhcp*'` und `'dhcp*1.*'`.

Title

Das Feld Titel ist optional, es bietet jedoch auf Linux-Systemen zusätzliche Filterfunktionen. Wenn Sie Titel verwenden, sollte er die Versionsinformationen des Pakets in einem der folgenden Formate enthalten:

YUM/SUSE Linux Enterprise Server (SLES):

```
{name}.{architecture}:{epoch}:{version}-{release}
```

APT

```
{name}.{architecture}:{version}
```

Für Linux-Patch-Titel können Sie einen oder mehrere Platzhalter in beliebigen Positionen verwenden, um die Anzahl der Paketzuordnungen zu erhöhen. Zum Beispiel: `'*32:9.8.2-0.*.rc1.57.amzn1'`.

Zum Beispiel:

- apt-Paketversion 1.2.25 ist derzeit auf Ihrem verwalteten Knoten installiert, aber Version 1.2.27 ist jetzt verfügbar.
- Fügen Sie die apt.amd64-Version 1.2.27 der Liste hinzu. Sie ist abhängig von apt utils.amd64 Version 1.2.27, aber apt-utils.amd64 Version 1.2.25 ist in der Liste angegeben.

In diesem Fall wird die Installation der APT-Version 1.2.27 blockiert und als „Fehlgeschlagen-NonCompliant“ gemeldet.

Windows Server

id

Das Feld id ist ein Pflichtfeld. Verwenden Sie es, um Patches mit Microsoft Knowledge Base-IDs (z. B. KB2736693) und Microsoft Security Bulletins-IDs (z. B. MS17-023) festzulegen.

Alle anderen Felder, die Sie in einer Patch-Liste für Windows bereitstellen möchten, sind optional und dienen nur zu Ihrer eigenen Information. Sie können zusätzlichen Felder verwenden, wie z. B. Titel, Klassifizierung, Schweregrad oder andere Angaben für detailliertere Informationen über die angegebenen Patches.

macOS

id

Das Feld id ist ein Pflichtfeld. Der Wert für das Feld id kann entweder mit einem `{package-name}.{package-version}`-Format oder einem {package_name}-Format bereitgestellt werden.

Patch-Beispiellisten

- 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'
```

Parametername: **`RebootOption`**

Nutzung: optional.

Optionen: `RebootIfNeeded` | `NoReboot`

Standardwert: `RebootIfNeeded`

> **Warning**
>
> Die Standardoption ist `RebootIfNeeded`. Stellen Sie sicher, dass Sie die richtige Option für Ihren Anwendungsfall auswählen. Wenn Ihre verwalteten Knoten beispielsweise sofort neu gestartet werden müssen, um einen Konfigurationsprozess abzuschließen, wählen Sie `RebootIfNeeded` aus. Oder wenn Sie die Verfügbarkeit von verwalteten Knoten bis zu einer geplanten Neustartzeit beibehalten müssen, wählen Sie `NoReboot` aus.

> **Important**
>
> Wir empfehlen nicht, Cluster-Instances in Amazon EMR (früher Amazon Elastic MapReduce genannt) zum Patchen zu verwendenPatch Manager. Wählen Sie insbesondere nicht die Option `RebootIfNeeded` für den Parameter `RebootOption` aus. (Diese Option ist in den SSM-Befehlsdokumenten für das Patchen von `AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation` und `AWS-RunPatchBaselineWithHooks` verfügbar.) Die zugrunde liegenden Befehle für das Patchen mithilfe von Patch Manager verwenden `yum`- und `dnf`-Befehle. Daher führen die Operationen aufgrund der Art und Weise, wie Pakete installiert werden, zu Inkompatibilitäten. Informationen zu den bevorzugten Methoden für die Aktualisierung von Software auf Amazon-EMR-Clustern finden Sie unter Verwendung des Standard-AMI für Amazon EMR im Amazon EMR Management Guide.

RebootIfNeeded

Wenn Sie die Option `RebootIfNeeded` auswählen, wird der verwaltete Knoten in einem der folgenden Fälle neu gestartet:

- Patch Manager ist auf einem oder mehreren Patches installiert.

  Patch Manager wertet nicht aus, ob ein Neustart vom Patch erfordert wird. Das System wird neu gestartet, auch wenn der Patch keinen Neustart erfordert.

- Patch Manager erkennt ein oder mehrere Patches mit dem Status `INSTALLED_PENDING_REBOOT` während der `Install`-Operation.

  Der `INSTALLED_PENDING_REBOOT` Status kann bedeuten, dass die Option ausgewählt `NoReboot` wurde, als der `Install` Vorgang das letzte Mal ausgeführt wurde, oder dass ein Patch Patch Manager seit dem letzten Neustart des verwalteten Knotens außerhalb installiert wurde.

Durch den Neustart von verwalteten Knoten wird in diesen beiden Fällen sichergestellt, dass aktualisierte Pakete aus dem Speicher gelöscht werden und das Patch- und Neustartverhalten über alle Betriebssysteme hinweg konsistent bleibt.

NoReboot

Wenn Sie die Option `NoReboot` auswählen, startet Patch Manager einen verwalteten Knoten nicht neu, selbst wenn über ihn während der `Install`-Operation Patches installiert wurden. Diese Option ist nützlich, wenn Sie wissen, dass für Ihre verwalteten Knoten nach dem Anwenden von Patches kein Neustart erforderlich ist oder Anwendungen bzw. Prozesse auf einem Knoten ausgeführt werden, die nicht durch einen Neustart beim Patchen unterbrochen werden sollten. Sie ist auch nützlich, wenn Sie mehr Kontrolle über das Timing von Neustarts von verwalteten Knoten wünschen, z. B. durch die Verwendung eines Wartungsfensters.

> **Note**
>
> Wenn Sie die Option `NoReboot` auswählen und ein Patch installiert ist, wird dem Patch der Status `InstalledPendingReboot` zugewiesen. Der verwaltete Knoten selbst wird jedoch als `Non-Compliant` gekennzeichnet. Nach einem Neustart und Ausführung einer `Scan`-Operation wird der Status des verwalteten Knoten auf `Compliant` aktualisiert.

Datei zum Nachverfolgen der Patch-Installation: Um die Patch-Installation nachzuverfolgen, insbesondere von Patches, die seit dem letzten Neustart des Systems installiert wurden, erstellt Systems Manager eine Datei auf dem verwalteten Knoten.

> **Important**
>
> Löschen oder ändern Sie die Tracking-Datei nicht. Wenn diese Datei gelöscht oder beschädigt wird, ist der Patch-Compliance-Bericht für den verwalteten Knoten ungenau. Starten Sie in diesem Fall den Knoten neu und führen Sie eine Patch-Scan-Operation aus, um die Datei wiederherzustellen.

Diese Tracking-Datei wird an den folgenden Speicherorten auf Ihren verwalteten Knoten gespeichert:

- Linux-Betriebssysteme:
  - `/var/log/amazon/ssm/patch-configuration/patch-states-configuration.json`
  - `/var/log/amazon/ssm/patch-configuration/patch-inventory-from-last-operation.json`
- Windows Server-Betriebssystem:
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchStatesConfiguration.json`
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchInventoryFromLastOperation.json`

Parametername: **`BaselineOverride`**

Nutzung: optional.

Sie können Patching-Voreinstellungen zur Laufzeit mit dem `BaselineOverride`-Parameter definieren. Diese Baseline-Überschreibung wird als JSON-Objekt in einem S3-Bucket beibehalten. Sie stellt sicher, dass Patchvorgänge die bereitgestellten Baselines verwenden, die dem Host-Betriebssystem entsprechen, anstatt die Regeln aus der Standard-Patch-Baseline anzuwenden

Weitere Informationen zur Verwendung des Parameters `BaselineOverride` finden Sie unter Verwenden des BaselineOverride Parameters.

Informationen über das **AWS-RunPatchBaselineAssociation** SSM-Dokument

Wie das AWS-RunPatchBaseline-Dokument führt auch AWS-RunPatchBaselineAssociation Patching-Operationen auf Instances für sicherheitsrelevante und andere Arten von Updates aus. Sie können das Dokument AWS-RunPatchBaselineAssociation auch verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows Server ist der Anwendungssupport auf Updates für Microsoft-Anwendungen beschränkt.)

Dieses Dokument unterstützt Amazon Elastic Compute Cloud (Amazon EC2)-Instances für Linux, macOS und Windows Server. Nicht-EC2-Knoten in einer Hybrid- und Multi-Cloud-Umgebung werden nicht unterstützt. Das Dokument führt die entsprechenden Aktionen für jede Plattform durch und ruft ein Python-Modul auf Linux und macOS Instanzen und ein PowerShell Modul auf Windows-Instanzen auf.

AWS-RunPatchBaselineAssociation unterscheidet sich jedoch auf folgende Weise von AWS-RunPatchBaseline:

- AWS-RunPatchBaselineAssociation ist hauptsächlich für die Verwendung mit State Manager-Zuordnungen vorgesehen, die mithilfe von Quick Setup, einer Fähigkeit von AWS Systems Manager, erstellt wurden. Insbesondere, wenn Sie den Quick Setup-Host-Management-Konfigurationstyp verwenden und die Option Scan instances for missing patches daily (Instances täglich auf fehlende Patches scannen) auswählen, verwendet das System AWS-RunPatchBaselineAssociation für diese Operation.

  In den meisten Fällen sollten Sie jedoch beim Einrichten eigener Patching-Vorgänge AWS-RunPatchBaseline oder AWS-RunPatchBaselineWithHooks anstelle von AWS-RunPatchBaselineAssociation auswählen.
- Wenn Sie das AWS-RunPatchBaselineAssociation-Dokument verwenden, können Sie ein Tag-Schlüssel-Paar im BaselineTags-Parameterfeld des Dokuments angeben. Wenn eine benutzerdefinierte Patch-Baseline in Ihrem AWS-Konto System diese Tags verwendetPatch Manager, verwendet eine Funktion von diese markierte Baseline AWS Systems Manager, wenn sie auf den Ziel-Instances ausgeführt wird, und nicht die aktuell angegebene „Standard“ -Patch-Baseline für den Betriebssystemtyp.

> **Important**
>
> Wenn Sie AWS-RunPatchBaselineAssociation in anderen Patching-Operationen als den mithilfe von Quick Setup eingerichteten verwenden und Sie den optionalen

`BaselineTags`-Parameter verwenden möchten, müssen Sie einige zusätzliche Berechtigungen für das Instance-Profil für Amazon Elastic Compute Cloud (Amazon EC2)-Instances bereitstellen. Weitere Informationen finden Sie unter Parametername: `BaselineTags`.

Die beiden folgenden Formate sind gültig für Ihre `BaselineTags`-Parameter:

`Key=tag-key,Values=tag-value`

`Key=tag-key,Values=tag-value1,tag-value2,tag-value3`

- Wenn `AWS-RunPatchBaselineAssociation` ausgeführt wird, werden die Patch-Compliance-Daten, die es sammelt, mit dem API-Befehl `PutComplianceItems` anstelle des Befehls `PutInventory`, der von `AWS-RunPatchBaseline` verwendet wird, aufgezeichnet. Dieser Unterschied bedeutet, dass die Patch-Compliance-Informationen gemäß einer bestimmten Zuordnung gespeichert und gemeldet werden. Patch-Compliance-Daten, die außerhalb dieser Zuordnung generiert wurden, werden nicht überschrieben.
- Die Patch-Compliance-Informationen, die nach der Ausführung von `AWS-RunPatchBaselineAssociation` gemeldet werden, geben an, ob eine Instance konform ist oder nicht. Sie enthält keine Details auf Patch-Ebene, wie die Ausgabe des folgenden AWS Command Line Interface (AWS CLI) -Befehls zeigt. Der Befehl filtert auf `Association` als Compliance-Typ:

```
aws ssm list-compliance-items \
    --resource-ids "i-02573cafcfEXAMPLE" \
    --resource-types "ManagedInstance" \
    --filters "Key=ComplianceType,Values=Association,Type=EQUAL" \
    --region us-east-2
```

Das System gibt unter anderem folgende Informationen zurück

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

Wenn ein Tag-Schlüssel-Paar-Wert als Parameter für das `AWS-RunPatchBaselineAssociation`-Dokument angegeben wurde, sucht Patch Manager nach einer benutzerdefinierten Patch-Baseline, die mit dem Betriebssystemtyp übereinstimmt und mit demselben Tag-Schlüssel-Paar gekennzeichnet wurde. Diese Suche ist nicht auf die aktuell angegebene Standard-Patch-Baseline oder die Baseline beschränkt, die einer Patch-Gruppe zugewiesen ist. Wenn keine Baseline mit den angegebenen Tags gefunden wird, sucht Patch Manager als Nächstes nach einer Patch-Gruppe, wenn eine in dem Befehl angegeben wurde, der `AWS-RunPatchBaselineAssociation` ausführt. Wenn keine Patch-Gruppe übereinstimmt, wird Patch Manager auf die aktuelle Standard-Patch-Baselines für das Betriebssystemkonto zurückgesetzt.

Wenn mehr als eine Patch-Baseline mit den Tags gefunden wird, die im `AWS-RunPatchBaselineAssociation`-Dokument angegeben sind, gibt Patch Manager eine Fehlermeldung zurück, die angibt, dass nur eine Patch-Baseline mit diesem Schlüssel-Wert-Paar gekennzeichnet werden kann, damit der Vorgang fortgesetzt wird.

> **Note**
>
> Auf Linux-Instances wird der entsprechende Paketmanager für jeden Instance-Typ verwendet, um Pakete zu installieren:
>
> - Amazon Linux 1, Amazon Linux 2, CentOS und RHEL Instances verwenden YUM. Oracle Linux Für YUM-Vorgänge erfordert Patch Manager `Python 2.6` oder eine höhere unterstützte Version (2.6–3.10).
> - Debian Server, Raspberry Pi OS und Ubuntu Server-Instances verwenden APT. Für APT-Vorgänge erfordert Patch Manager eine unterstützte Version von `Python 3` (3.0–3.10).

- SUSE Linux Enterprise Server-Instances verwenden Zypper. Für Zypper-Vorgänge erfordert Patch Manager `Python 2.6` oder eine höhere unterstützte Version (2.6–3.10).

Nachdem der Scan abgeschlossen wurde oder alle genehmigten und zutreffenden Updates installiert und je nach Bedarf Neustarts durchgeführt wurden, werden Patch-Compliance-Informationen auf einer Instance generiert und wieder an den Patchmanager-Service gemeldet.

> **Note**
>
> Wenn der Parameter `RebootOption` im Dokument `AWS-RunPatchBaselineAssociation` auf `NoReboot` gesetzt ist, wird die Instance nach dem Ausführen von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.

Weitere Informationen zum Anzeigen von Patch-Compliance-Daten finden Sie unter Info zu Patch Compliance.

`AWS-RunPatchBaselineAssociation` parameters

`AWS-RunPatchBaselineAssociation` unterstützt vier Parameter. Die Parameter `Operation` und `AssociationId` müssen angegeben werden. Die Parameter `InstallOverrideList`, `RebootOption` und `BaselineTags` sind optional.

Parameter

- Parametername: Operation
- Parametername: BaselineTags
- Parametername: AssociationId
- Parametername: InstallOverrideList
- Parametername: RebootOption

Parametername: **`Operation`**

Nutzung: erforderlich.

Optionen: `Scan` | `Install`.

Scan

Wenn Sie die Option `Scan` wählen, bestimmt `AWS-RunPatchBaselineAssociation` den Patch-Compliance-Status der Instance und meldet diese Informationen an Patch Manager. `Scan` fordert nicht zum Installieren von Updates oder zum Neustarten von Instances auf. Stattdessen erkennt der Vorgang, wo für die Instance genehmigte und geeignete Updates fehlen.

Installieren

Bei Auswahl der Option `Install` versucht `AWS-RunPatchBaselineAssociation`, die genehmigten und geeigneten Updates zu installieren, die auf der Instance fehlen. Patch-Compliance-Informationen, die als Teil eines `Install`-Vorgangs generiert wurden, enthalten keine fehlenden Updates, melden allerdings möglicherweise Updates im Fehlerzustand, wenn die Installation des Updates aus einem beliebigen Grund nicht erfolgreich war. Immer wenn ein Update auf einer Instance installiert wird, wird die Instance neu gestartet, um sicherzustellen, dass es installiert und aktiviert ist. (Ausnahme: Wenn der `RebootOption`-Parameter im `AWS-RunPatchBaselineAssociation`-Dokument auf `NoReboot` gesetzt ist, wird die Instance nach der Ausführung von Patch Manager nicht neugestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

> **Note**
>
> Wenn ein von den Basisregeln festgelegter Patch installiert wird, bevor der Patch Manager die Instance aktualisiert, wird das System möglicherweise nicht wie erwartet neu gestartet. Dies kann passieren, wenn ein Patch manuell von einem Benutzer oder automatisch von einem anderen Programm, z. B. dem `unattended-upgrades`-Paket auf Ubuntu Server, installiert wird.

Parametername: **`BaselineTags`**

Nutzung: optional.

`BaselineTags` ist ein eindeutiges Tag-Schlüssel-Wert-Paar, das Sie auswählen und einer individuellen benutzerdefinierten Patch-Baseline zuweisen. Sie können einen oder mehrere Werte für diesen Parameter angeben. Beider der folgenden Formate sind gültig:

`Key=`*`tag-key`*`,Values=`*`tag-value`*

`Key=`*`tag-key`*`,Values=`*`tag-value1`*`,`*`tag-value2`*`,`*`tag-value3`*

Der `BaselineTags`-Wert wird von Patch Manager verwendet, um sicherzustellen, dass eine Gruppe von Instances, für die in einem Vorgang Patches durchgeführt werden, den genau gleichen Satz genehmigter Patches aufweist. Wenn der Patching-Vorgang ausgeführt wird, überprüft Patch Manager, ob eine Patch-Baseline für den Betriebssystemtyp mit demselben Schlüssel-Wert-Paar versehen ist, das Sie für `BaselineTags` angeben. Wenn eine Übereinstimmung vorliegt, wird diese benutzerdefinierte Patch-Baseline verwendet. Wenn keine Übereinstimmung vorliegt, wird eine Patch-Baseline anhand einer beliebigen Patchgruppe identifiziert, die für die Patching-Operation angegeben wurde. Wenn keine vorhanden ist, wird die AWS verwaltete vordefinierte Patch-Baseline für dieses Betriebssystem verwendet.

Zusätzliche Berechtigungsanforderungen

Wenn Sie `AWS-RunPatchBaselineAssociation` in anderen Patching-Operationen als den mithilfe von Quick Setup eingerichteten verwenden und Sie den optionalen `BaselineTags`-Parameter verwenden möchten, müssen Sie die folgenden Berechtigungen für das Instance-Profil für Amazon Elastic Compute Cloud (Amazon EC2)-Instances bereitstellen.

> **Note**
>
> Quick Setup und `AWS-RunPatchBaselineAssociation` unterstützen keine On-Premises-Server und virtuellen Maschinen (VMs).

```
{
    "Effect": "Allow",
    "Action": [
        "ssm:DescribePatchBaselines",
        "tag:GetResources"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ssm:GetPatchBaseline",
        "ssm:DescribeEffectivePatchesForPatchBaseline"
    ],
    "Resource": "patch-baseline-arn"
}
```

*patch-baseline-arn*Ersetzen Sie es durch den Amazon-Ressourcennamen (ARN) der Patch-Baseline, auf die Sie Zugriff gewähren möchten, im folgenden Format`arn:aws:ssm:us-east-2:123456789012:patchbaseline/pb-0c10e65780EXAMPLE`.

Parametername: **`AssociationId`**

Nutzung: erforderlich.

`AssociationId` ist die ID einer vorhandenen Zuordnung in State Manager, einer Funktion von AWS Systems Manager. Es wird von Patch Manager verwendet, um einer angegebenen Zuordnung Compliance-Daten hinzuzufügen. Diese Zuordnung bezieht sich auf einen `Scan`-Patching-Vorgang, der in einer in Quick Setup erstellten Host-Management-Konfiguration aktiviert ist. Durch das Senden von Patching-Ergebnissen als Zuordnungs-Compliance-Daten anstelle von Inventar-Compliance-Daten werden vorhandene Inventar-Compliance-Informationen für Ihre Instances weder nach einer Patching-Operation noch für andere Zuordnungs-IDs überschrieben. Wenn Sie noch keine Zuordnung erstellt haben, die Sie verwenden möchten, können Sie eine erstellen, indem Sie den Befehl create-association ausführen. Beispielsweise:

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
```

Parametername: **`InstallOverrideList`**

Nutzung: optional.

Mit `InstallOverrideList` können Sie eine https-URL oder eine Amazon Simple Storage Service (Amazon S3)-URL im Pfadstil zu einer Liste mit zu installierenden Patches angeben. Diese im YAML-Format geführte Patch-Installationsliste überschreibt die von der Standard-Patch-Baseline angegebenen Patches. Dies bietet Ihnen eine differenziertere Kontrolle darüber, welche Patches auf Ihren Instances installiert sind.

Das Verhalten des Patch-Vorgangs bei Verwendung des `InstallOverrideList` Parameters unterscheidet sich zwischen Linux- und macOS verwalteten Knoten und Windows Server verwalteten Knoten. Unter Linux & Patch Manager versuchtmacOS, in der `InstallOverrideList` Patch-Liste enthaltene Patches anzuwenden, die in einem beliebigen Repository vorhanden sind, das auf dem Knoten aktiviert ist, unabhängig davon, ob die Patches den Patch-Basisregeln entsprechen oder nicht. Auf Windows Server Knoten werden Patches in der `InstallOverrideList` Patch-Liste jedoch nur angewendet, wenn sie auch den Patch-Baseline-Regeln entsprechen.

Beachten Sie, dass Compliance-Berichte Patch-Status entsprechend den Angaben in der Patch-Baseline wiedergeben, nicht entsprechend Ihren Angaben in einer `InstallOverrideList`-Liste von Patches. Mit anderen Worten: Scan-Operationen ignorieren den Parameter `InstallOverrideList`. Auf diese Weise wird sichergestellt, dass Compliance-Berichte den Patch-Status konsistent entsprechend der Richtlinie wiedergeben und nicht danach, was für eine bestimmte Patching-Operation genehmigt wurde.

Gültige URL-Formate

> **Note**
>
> Wenn Ihre Datei in einem öffentlich zugänglichen Bucket gespeichert ist, können Sie entweder ein HTTPS-URL-Format oder eine URL im Amazon S3-Pfadstil angeben. Wenn Ihre Datei in einem privaten Bucket gespeichert ist, müssen Sie eine URL im Amazon S3-Pfadstil angeben.

- Beispiel des HTTPS-URL-Formats:

```
https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/my-windows-override-list.yaml
```

- Beispiel-URL im Amazon-S3-Pfadstil:

```
s3://DOC-EXAMPLE-BUCKET/my-windows-override-list.yaml
```

Gültige YAML-Inhaltsformate

Die Formate, die Sie verwenden, um Patches in Ihrer Liste anzugeben, hängen von dem Betriebssystem Ihrer Instance ab. Das allgemeine Format lautet jedoch folgendermaßen:

```
patches:
    -
        id: '{patch-d}'
        title: '{patch-title}'
        {additional-fields}:{values}
```

Sie können zwar zusätzliche Felder in der YAML-Datei bereitstellen, diese werden jedoch während der Patch-Operationen ignoriert.

Darüber hinaus empfehlen wir zu überprüfen, ob das Format Ihrer YAML-Datei gültig ist, bevor Sie die Liste in Ihrem S3-Bucket hinzufügen oder aktualisieren. Weitere Informationen zum YAML-Format finden Sie unter yaml.org. Für Validierungstool-Optionen suchen Sie im Internet nach "yaml format validators" durch.

- Microsoft Windows

  id

  Das Feld id ist ein Pflichtfeld. Verwenden Sie es, um Patches mit Microsoft Knowledge Base-IDs (z. B. KB2736693) und Microsoft Security Bulletins-IDs (z. B. MS17-023) festzulegen.

  Alle anderen Felder, die Sie in einer Patch-Liste für Windows bereitstellen möchten, sind optional und dienen nur zu Ihrer eigenen Information. Sie können zusätzlichen Felder verwenden, wie z. B. Titel, Klassifizierung, Schweregrad oder andere Angaben für detailliertere Informationen über die angegebenen Patches.
- Linux

  id

Das Feld id ist ein Pflichtfeld. Verwenden Sie es, um Patches mit Paketnamen und Architektur anzugeben. Zum Beispiel: `'dhclient.x86_64'`. Sie können Platzhalter in der ID verwenden, um mehrere Pakete anzugeben. Zum Beispiel `'dhcp*'` und `'dhcp*1.*'`.

Titel

Das Feld Titel ist optional, es bietet jedoch auf Linux-Systemen zusätzliche Filterfunktionen. Wenn Sie Titel verwenden, sollte er die Versionsinformationen des Pakets in einem der folgenden Formate enthalten:

YUM/SUSE Linux Enterprise Server (SLES):

```
{name}.{architecture}:{epoch}:{version}-{release}
```

APT

```
{name}.{architecture}:{version}
```

Für Linux-Patch-Titel können Sie einen oder mehrere Platzhalter in beliebigen Positionen verwenden, um die Anzahl der Paketzuordnungen zu erhöhen. Zum Beispiel: `'*32:9.8.2-0.*.rc1.57.amzn1'`.

Zum Beispiel:

- apt-Paketversion 1.2.25 ist derzeit auf Ihrer Instance installiert, aber Version 1.2.27 ist jetzt verfügbar.
- Fügen Sie die apt.amd64-Version 1.2.27 der Liste hinzu. Sie ist abhängig von apt utils.amd64 Version 1.2.27, aber apt-utils.amd64 Version 1.2.25 ist in der Liste angegeben.

In diesem Fall wird die Installation der APT-Version 1.2.27 blockiert und als „Fehlgeschlagen-NonCompliant“ gemeldet.

Andere Felder

Alle anderen Felder, die Sie in einer Patch-Liste für Linux bereitstellen möchten, sind optional und dienen nur zu Ihrer eigenen Information. Sie können zusätzlichen Felder verwenden, wie z. B. Klassifizierung, Schweregrad oder andere Angaben für detailliertere Informationen über die angegebenen Patches.

Patch-Beispiellisten

- 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'
```

Parametername: **`RebootOption`**

Nutzung: optional.

Optionen: `RebootIfNeeded` | `NoReboot`

Standardwert: `RebootIfNeeded`

> **Warning**
>
> Die Standardoption ist `RebootIfNeeded`. Stellen Sie sicher, dass Sie die richtige Option für Ihren Anwendungsfall auswählen. Wenn Ihre Instances beispielsweise sofort neu starten müssen, um einen Konfigurationsprozess abzuschließen, wählen Sie `RebootIfNeeded` aus. Oder wenn Sie die Verfügbarkeit von Instances bis zu einer geplanten Neustartzeit beibehalten müssen, wählen Sie `NoReboot` aus.

> **Important**
>
> Wir empfehlen nicht, Cluster-Instances in Amazon EMR (früher Amazon Elastic MapReduce genannt) zum Patchen zu verwendenPatch Manager. Wählen Sie insbesondere nicht die Option `RebootIfNeeded` für den Parameter `RebootOption` aus. (Diese Option ist in den SSM-Befehlsdokumenten für das Patchen von `AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation` und `AWS-RunPatchBaselineWithHooks` verfügbar.) Die zugrunde liegenden Befehle für das Patchen mithilfe von Patch Manager verwenden `yum`- und `dnf`-Befehle. Daher führen die Operationen aufgrund der Art und Weise, wie Pakete installiert werden, zu Inkompatibilitäten. Informationen zu den bevorzugten Methoden für die Aktualisierung von Software auf Amazon-EMR-Clustern finden Sie unter Verwendung des Standard-AMI für Amazon EMR im Amazon EMR Management Guide.

RebootIfNeeded

Wenn Sie die Option `RebootIfNeeded` auswählen, wird die Instance in einem der folgenden Fälle neu gestartet:

- Patch Manager ist auf einem oder mehreren Patches installiert.

Patch Manager wertet nicht aus, ob ein Neustart vom Patch erfordert wird. Das System wird neu gestartet, auch wenn der Patch keinen Neustart erfordert.

- Patch Manager erkennt ein oder mehrere Patches mit dem Status `INSTALLED_PENDING_REBOOT` während der `Install`-Operation.

  Der `INSTALLED_PENDING_REBOOT` Status kann bedeuten, dass die Option ausgewählt `NoReboot` wurde, als der `Install` Vorgang das letzte Mal ausgeführt wurde, oder dass ein Patch Patch Manager seit dem letzten Neustart des verwalteten Knotens außerhalb installiert wurde.

Durch den Neustart von Instances wird in diesen beiden Fällen sichergestellt, dass aktualisierte Pakete aus dem Speicher gelöscht werden und das Patch- und Neustartverhalten über alle Betriebssysteme hinweg konsistent bleibt.

NoReboot

Wenn Sie die Option `NoReboot` auswählen, startet Patch Manager eine Instance nicht neu, selbst wenn über ihn während der `Install`-Operation Patches installiert wurden. Diese Option ist nützlich, wenn Sie wissen, dass für Ihre Instances nach dem Anwenden von Patches kein Neustart erforderlich ist oder Anwendungen bzw. Prozesse auf einer Instance ausgeführt werden, die nicht durch einen Neustart des Patches unterbrochen werden sollten. Sie ist auch nützlich, wenn Sie mehr Kontrolle über das Timing von Instance-Neustarts wünschen, z. B. durch die Verwendung eines Wartungsfensters.

Datei zum Nachverfolgen der Patch-Installation (Tracking-Datei): Um die Patch-Installation nachzuverfolgen, insbesondere von Patches, die seit dem letzten Neustart des Systems installiert wurden, erstellt Systems Manager eine Datei auf der verwalteten Instance.

> **Important**
>
> Löschen oder ändern Sie die Tracking-Datei nicht. Wenn diese Datei gelöscht oder beschädigt wird, ist der Patch-Compliance-Bericht für die Instance ungenau. Starten Sie in diesem Fall die Instance neu und führen Sie einen Patch-Scan-Vorgang aus, um die Datei wiederherzustellen.

Diese Tracking-Datei wird an den folgenden Speicherorten auf Ihren verwalteten Instances gespeichert:

- Linux-Betriebssysteme:
  - `/var/log/amazon/ssm/patch-configuration/patch-states-configuration.json`
  - `/var/log/amazon/ssm/patch-configuration/patch-inventory-from-last-operation.json`
- Windows Server-Betriebssystem:
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchStatesConfiguration.json`
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchInventoryFromLastOperation.json`

Informationen über das **`AWS-RunPatchBaselineWithHooks`** SSM-Dokument

AWS Systems Manager unterstütztAWS-RunPatchBaselineWithHooks, ein Systems Manager Manager-Dokument (SSM-Dokument) fürPatch Manager, eine Fähigkeit von AWS Systems Manager. Dieses SSM-Dokument führt Patch-Operationen auf verwaltete Knoten sowohl für sicherheitsrelevante als auch für andere Arten von Updates durch.

`AWS-RunPatchBaselineWithHooks` unterscheidet sich auf folgende Weise von `AWS-RunPatchBaseline`:

- Ein Wrapper-Dokument – `AWS-RunPatchBaselineWithHooks` ist ein Wrapper für `AWS-RunPatchBaseline` und setzt für einige seiner Operationen auf`AWS-RunPatchBaseline`.
- Die **`Install`**-Operation – `AWS-RunPatchBaselineWithHooks` unterstützt Lebenszyklus-Hooks, die während dem Patchen von verwalteten Knoten an festgelegten Punkten ausgeführt werden. Da Patch-Installationen manchmal den Neustart von verwalteten Knoten erfordern, ist die Patch-Operation in zwei Ereignisse unterteilt, wobei insgesamt drei Hooks enthalten sind, die benutzerdefinierte Funktion unterstützen. Der erste Hook ist vor der `Install with NoReboot`-Operation. Der zweite Hook ist nach der `Install with NoReboot`-Operation. Der dritte Hook ist nach dem Neustart des verwalteten Knoten verfügbar.
- Keine Unterstützung für benutzerdefinierte Patchlisten – `AWS-RunPatchBaselineWithHooks` unterstützt den `InstallOverrideList`-Parameter nicht.
- SSM Agent-Support – `AWS-RunPatchBaselineWithHooks` erfordert die Installation von SSM Agent 3.0.502 oder höher auf dem zu patchenden verwalteten Knoten.

Wenn das Dokument ausgeführt wird, verwendet es die Patch-Baseline, die aktuell der „Standard“ für einen Betriebssystemtyp ist, wenn keine Patch-Gruppe angegeben ist. Andernfalls werden die Patch-Baselines verwendet, die der Patch-Gruppe zugeordnet sind. Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

Sie können das Dokument `AWS-RunPatchBaselineWithHooks` verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows ist der Anwendungssupport auf Updates für von Microsoft veröffentlichte Anwendungen beschränkt.)

Dieses Dokument unterstützt von Linux, macOS und Windows Server verwaltete Knoten. Das Dokument führt die entsprechenden Aktionen für jede Plattform durch.

Linux

Auf Linux-verwalteten Knoten ruft das Dokument `AWS-RunPatchBaselineWithHooks` ein Python-Modul auf, das wiederum einen entsprechenden Snapshot der Patch-Baseline für den verwalteten Knoten herunterlädt. Dieser Patch-Baseline-Snapshot verwendet die definierten Regeln und Listen der genehmigten und gesperrten Patches, um den entsprechenden Paketmanager für jeden Knoten-Typ anzutreiben:

- Die verwalteten Knoten Amazon Linux 1, Amazon Linux 2Oracle Linux, CentOS und RHEL 7 verwenden YUM. Für YUM-Vorgänge erfordert Patch Manager `Python 2.6` oder eine höhere unterstützte Version (2.6–3.10).
- Von RHEL 8 verwaltete Knoten verwenden DNF. Für DNF-Vorgänge erfordert Patch Manager eine unterstützte Version von `Python 2` oder `Python 3` (2.6–3.10). (Keine der beiden Versionen ist standardmäßig auf RHEL 8 installiert. Sie müssen die eine oder andere Version manuell installieren.)
- Debian Server, Raspberry Pi OS und Ubuntu Server-Instances verwenden APT. Für APT-Vorgänge erfordert Patch Manager eine unterstützte Version von `Python 3` (3.0–3.10).
- Von SUSE Linux Enterprise Server verwaltete Knoten verwenden Zypper. Für Zypper-Vorgänge erfordert Patch Manager `Python 2.6` oder eine höhere unterstützte Version (2.6–3.10).

macOS

Auf macOS-verwalteten Knoten ruft das Dokument `AWS-RunPatchBaselineWithHooks` ein Python-Modul auf, das wiederum einen entsprechenden Snapshot der Patch-Baseline für den verwalteten Knoten herunterlädt. Als nächstes ruft ein Python-Subprozess die CLI auf dem Knoten

auf, um die Installations- und Updateinformationen für die angegebenen Paketmanager abzurufen und den entsprechenden Paketmanager für jedes Updatepaket zu steuern.

Windows Server

Auf Windows Server verwalteten Knoten lädt das `AWS-RunPatchBaselineWithHooks` Dokument ein PowerShell Modul herunter und ruft es auf, das wiederum einen Snapshot der Patch-Baseline herunterlädt, die für den verwalteten Knoten gilt. Dieser Patch-Baseline-Snapshot enthält eine Liste genehmigter Patches, die kompiliert werden, indem die Patch-Baseline auf einem WSUS-Server (Windows Server Update Services) abgefragt wird. Diese Liste wird an die Windows Update-API weitergeleitet, die das Herunterladen und Installieren des genehmigten Patches entsprechend steuert.

Jeder Snapshot ist spezifisch für eine AWS-Konto Patch-Gruppe, ein Betriebssystem und eine Snapshot-ID. Der Snapshot wird über eine vorsignierte Amazon Simple Storage Service (Amazon S3)-URL bereitgestellt, die 24 Stunden nach Erstellung des Snapshots abläuft. Wenn Sie jedoch denselben Snapshot-Inhalt auf andere verwaltete Knoten anwenden möchten, können Sie nach Ablauf der URL bis zu drei Tage nach Erstellung des Snapshots eine neue vorsignierte Amazon-S3-URL generieren. Verwenden Sie dazu den Befehl get-deployable-patch-snapshot-for-instance.

Nachdem alle genehmigten und zutreffenden Updates installiert und je nach Bedarf Neustarts durchgeführt wurden, werden Patch-Compliance-Informationen auf einem verwalteten Knoten generiert und wieder an Patch Manager gemeldet.

> **Note**
>
> Wenn der Parameter `RebootOption` im Dokument `AWS-RunPatchBaselineWithHooks` auf `NoReboot` gesetzt ist, wird der verwaltete Knoten nach dem Ausführen von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.

Weitere Informationen zum Anzeigen von Patch-Compliance-Daten finden Sie unter Info zu Patch Compliance.

`AWS-RunPatchBaselineWithHooks`-Betriebsschritte

Wenn `AWS-RunPatchBaselineWithHooks` ausgeführt wird, werden die folgenden Schritte durchgeführt:

1. Scan – Eine `Scan`-Operation mit `AWS-RunPatchBaseline` wird auf dem verwalteten Knoten ausgeführt und ein Compliance-Bericht wird generiert und hochgeladen.
2. Überprüfen der lokalen Patch-Zustände – Ein Skript wird ausgeführt, um zu bestimmen, welche Schritte auf der Grundlage der ausgewählten Operation und dem `Scan`-Ergebnis aus Schritt 1 ausgeführt werden.
   a. Wenn die ausgewählte Operation `Scan` ist, wird die Operation als abgeschlossen markiert. Die Operation ist abgeschlossen.
   b. Wenn die ausgewählte Operation `Install` ist, werte Patch Manager das `Scan`-Ergebnis aus Schritt 1, um zu bestimmen, was als nächstes ausgeführt werden soll:
      i. Wenn keine fehlenden Patches erkannt werden und keine ausstehenden Neustarts erforderlich sind, fährt die Operation direkt mit dem letzten Schritt (Schritt 8) fort, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen.
      ii. Wenn keine fehlenden Patches erkannt werden, aber ausstehende Neustarts erforderlich sind und die Neustartoption `NoReboot` ist, fährt die Operation direkt mit dem letzten Schritt (Schritt 8) fort, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen.
      iii. Andernfalls fährt die Operation mit dem nächsten Schritt fort.
3. Hook-Operation vor dem Patchen – Das SSM-Dokument, das Sie für den ersten Lebenszyklus-Hook bereitgestellt haben, `PreInstallHookDocName` wird auf dem verwalteten Knoten ausgeführt.
4. Installation mit NoReboot — Auf dem verwalteten Knoten `AWS-RunPatchBaseline` wird ein `Install` Vorgang `NoReboot` mit der Neustartoption ausgeführt, und es wird ein Konformitätsbericht generiert und hochgeladen.
5. Hook-Operation nach der Installation – Das SSM-Dokument, das Sie für den zweiten Lebenszyklus-Hook bereitgestellt haben, `PostInstallHookDocName` wird auf dem verwalteten Knoten ausgeführt.
6. Überprüfen des Neustarts – Ein Skript wird ausgeführt, um festzustellen, ob ein Neustart für den verwalteten Knoten erforderlich ist und welche Schritte ausgeführt werden sollen:
   a. Wenn die ausgewählte Neustartoption `NoReboot` ist, geht die Operation direkt zum letzten Schritt (Schritt 8) über, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen.
   b. Wenn die ausgewählte Neustartoption `RebootIfNeeded` ist, prüft Patch Manager, ob ausstehende Neustarts aus dem in Schritt 4 erfassten Bestand erforderlich sind. Dies bedeutet,

dass der Vorgang mit Schritt 7 fortgesetzt wird und der verwaltete Knoten in einem der folgenden Fälle neu gestartet wird:

i. Patch Manager hat einen oder mehrere Patches installiert. (Patch Manager wertet nicht aus, ob für den Patch ein Neustart erforderlich ist. Das System wird auch dann neu gestartet, wenn der Patch keinen Neustart erfordert.)

ii. Patch Manager erkennt ein oder mehrere Patches mit dem Status `INSTALLED_PENDING_REBOOT` während des Installationsvorgangs. Der `INSTALLED_PENDING_REBOOT` Status kann bedeuten, dass die Option ausgewählt `NoReboot` wurde, als der Installationsvorgang das letzte Mal ausgeführt wurde, oder dass ein Patch außerhalb des Zeitraums Patch Manager seit dem letzten Neustart des verwalteten Knotens installiert wurde.

Wenn keine Patches gefunden werden, die diese Kriterien erfüllen, ist der Patch-Vorgang für verwaltete Knoten abgeschlossen, und der Vorgang fährt direkt mit dem letzten Schritt (Schritt 8) fort, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen.

7. Neustart und Bericht – Eine Installations-Operation mit der Neustart-Option `RebootIfNeeded` wird auf dem verwalteten Knoten unter Verwendung von `AWS-RunPatchBaseline` ausgeführt und ein Compliance-Bericht wird generiert und hochgeladen.
8. Hook-Operation nach Neustart – Das SSM-Dokument, das Sie für den dritten Lebenszyklus-Hook bereitgestellt haben, `OnExitHookDocName` wird auf dem verwalteten Knoten ausgeführt.

Bei einer `Scan`-Operation wird der Prozess der Ausführung des Dokuments beendet, wenn Schritt 1 fehlschlägt, und der Schritt wird als fehlgeschlagen gemeldet, obwohl nachfolgende Schritte als erfolgreich gemeldet werden.

Wenn bei einem `Install`-Vorgang einer der `aws:runDocument`-Schritte während des Vorgangs fehlschlagen, werden diese Schritte als fehlgeschlagen gemeldet, und der Vorgang fährt direkt mit dem letzten Schritt (Schritt 8) fort, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen. Dieser Schritt wird als fehlgeschlagen gemeldet, der letzte Schritt meldet den Status des Vorgangsergebnisses, und alle dazwischen liegenden Schritte werden als erfolgreich gemeldet.

`AWS-RunPatchBaselineWithHooks` parameters

`AWS-RunPatchBaselineWithHooks` unterstützt sechs Parameter.

Der Parameter `Operation` muss angegeben werden.

Die Parameter `RebootOption`, `PreInstallHookDocName`, `PostInstallHookDocName` und `OnExitHookDocName` sind optional.

`Snapshot-ID` ist eigentlich optional, wir empfehlen jedoch, einen benutzerdefinierten Wert dafür anzugeben, wenn Sie `AWS-RunPatchBaselineWithHooks` außerhalb eines Wartungsfensters ausführen. Lassen Sie Patch Manager den Wert automatisch angeben, wenn das Dokument als Teil einer Wartungsfenster-Operation ausgeführt wird.

Parameter

- Parametername: Operation
- Parametername: Snapshot ID
- Parametername: RebootOption
- Parametername: PreInstallHookDocName
- Parametername: PostInstallHookDocName
- Parametername: OnExitHookDocName

Parametername: **`Operation`**

Nutzung: erforderlich.

Optionen: `Scan` | `Install`.

Scan

Wenn Sie die Option `Scan` wählen, verwende das System das Dokument `AWS-RunPatchBaseline`, um den Patch-Compliance-Zustand des verwalteten Knoten zu bestimmen und diese Informationen an Patch Manager zu melden. `Scan` fordert nicht zum Installieren von Updates oder zum Neustarten von verwalteten Knoten auf. Stattdessen erkennt die Operation, wo für den Knoten genehmigte und geeignete Updates fehlen.

Installieren

Bei Auswahl der Option `Install` versucht `AWS-RunPatchBaselineWithHooks`, die genehmigten und geeigneten Updates zu installieren, die auf dem verwalteten Knoten fehlen. Patch-Compliance-Informationen, die als Teil eines `Install`-Vorgangs generiert wurden, enthalten keine fehlenden Updates, melden allerdings möglicherweise Updates im Fehlerzustand, wenn die Installation des Updates aus einem beliebigen Grund nicht erfolgreich war. Immer

wenn ein Update auf einem verwalteten Knoten installiert wird, wird der Knoten neu gestartet, um sicherzustellen, dass das Update installiert und aktiviert ist. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaselineWithHooks` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

> **Note**
>
> Wenn ein von den Basisregeln festgelegter Patch installiert wird, bevor der Patch Manager den verwalteten Knoten aktualisiert, wird das System möglicherweise nicht wie erwartet neu gestartet. Dies kann passieren, wenn ein Patch manuell von einem Benutzer oder automatisch von einem anderen Programm, z. B. dem `unattended-upgrades`-Paket auf Ubuntu Server, installiert wird.

Parametername: **`Snapshot ID`**

Nutzung: optional.

`Snapshot ID` ist eine eindeutige ID (GUID), die von Patch Manager verwendet wird, um sicherzustellen, dass ein Satz von verwalteten Knoten, für die in einer einzelnen Operation Patches durchgeführt werden, den genau gleichen Satz genehmigter Patches aufweist. Auch wenn der Parameter als optional definiert ist, hängen unsere Empfehlungen für bewährte Methoden davon ab, ob Sie `AWS-RunPatchBaselineWithHooks` in einem Wartungsfenster, wie in der folgenden Tabelle beschrieben, ausführen.

Bewährte Methoden für **`AWS-RunPatchBaselineWithHooks`**

Mode	Bewährte Methode	Details
Ausführen von `AWS-RunPatchBaselineWithHooks` innerhalb eines Wartungsfensters	Geben Sie keine Snapshot ID an. Patch Manager wird sie für Sie bereitstellen.	Falls Sie ein Wartungsfenster zum Ausführen von `AWS-RunPatchBaselineWithHooks` verwenden, dürfen Sie Ihre eigene generierte Snapshot ID nicht angeben. In diesem Szenario stellt Systems Manager einen GUID-Wert auf Grundlage der

Mode	Bewährte Methode	Details
		Wartungsfensterausführungs-ID bereit. Auf diese Weise wird sichergestellt, dass eine richtige ID für alle Aufrufe von `AWS-RunPatchBaseli neWithHooks` in diesem Wartungsfenster verwendet wird. Wenn Sie einen Wert in diesem Szenario angeben, beachten Sie, dass der Snapshot der Patch-Bas eline möglicherweise nicht länger als drei Tagen erhalten bleibt. Danach wird ein neuer Snapshot erstellt, auch wenn Sie dieselbe ID angeben, nachdem der Snapshot abgelaufen ist.

Mode	Bewährte Methode	Details
Ausführen von `AWS-RunPatchBaselineWithHooks` außerhalb eines Wartungsfensters	Generieren Sie einen benutzerdefinierten GUID-Wert für die Snapshot-ID und geben Sie ihn an.[1]	Wenn Sie kein Wartungsfenster zum Ausführen von `AWS-RunPatchBaselineWithHooks` verwenden, empfehlen wir, dass Sie eine eindeutige Snapshot-ID für jede Patch-Baseline generieren und angeben, insbesondere wenn Sie das Dokument `AWS-RunPatchBaselineWithHooks` auf mehreren verwaltete Knoten in derselben Operation ausführen. Wenn Sie keine ID in diesem Szenario angeben, generiert Systems Manager eine andere Snapshot-ID für jeden verwalteten Knoten, an den der Befehl gesendet wird. Dies kann zu unterschiedlichen Sätzen von Patches führen, die auf den Knoten angegeben sind. Zum Beispiel: Angenommen, Sie führen das Dokument `AWS-RunPatchBaselineWithHooks` direkt über Run Command, eine Funktion von AWS Systems Manager, aus und richten es auf eine Gruppe von 50 verwalteten Knoten aus. Das Angeben einer benutzerdefinierten Snapshot-ID führt zur Erstellun

Mode	Bewährte Methode	Details
		g eines einzelnen Baseline-Snapshots, der verwendet wird, um alle verwaltete Knoten zu bewerten und zu patchen. Dadurch wird gewährleistet, dass sie letztendlich einen konsistenten Zustand aufweisen.

[1] Sie können jedes beliebige Tool zum Generieren eines Werts für den Snapshot-ID-Parame ter verwenden, das eine GUID generieren kann. Beispielsweise können Sie mit dem `New-Guid` Cmdlet eine GUID im PowerShell Format von generieren. `12345699-9405-4f69-bc5e-931 5aEXAMPLE`

Parametername: **`RebootOption`**

Nutzung: optional.

Optionen: `RebootIfNeeded` | `NoReboot`

Standardwert: `RebootIfNeeded`

> **Warning**
>
> Die Standardoption ist `RebootIfNeeded`. Stellen Sie sicher, dass Sie die richtige Option für Ihren Anwendungsfall auswählen. Wenn Ihre verwalteten Knoten beispielsweise sofort neu gestartet werden müssen, um einen Konfigurationsprozess abzuschließen, wählen Sie `RebootIfNeeded` aus. Oder wenn Sie die Verfügbarkeit von verwalteten Knoten bis zu einer geplanten Neustartzeit beibehalten müssen, wählen Sie `NoReboot` aus.

> **Important**
>
> Wir empfehlen nicht, Cluster-Instances in Amazon EMR (früher Amazon Elastic MapReduce genannt) zum Patchen zu verwendenPatch Manager. Wählen Sie insbesondere nicht die Option `RebootIfNeeded` für den Parameter `RebootOption` aus. (Diese Option ist

in den SSM-Befehlsdokumenten für das Patchen von `AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation` und `AWS-RunPatchBaselineWithHooks` verfügbar.) Die zugrunde liegenden Befehle für das Patchen mithilfe von Patch Manager verwenden `yum`- und `dnf`-Befehle. Daher führen die Operationen aufgrund der Art und Weise, wie Pakete installiert werden, zu Inkompatibilitäten. Informationen zu den bevorzugten Methoden für die Aktualisierung von Software auf Amazon-EMR-Clustern finden Sie unter Verwendung des Standard-AMI für Amazon EMR im Amazon EMR Management Guide.

RebootIfNeeded

Wenn Sie die Option `RebootIfNeeded` auswählen, wird der verwaltete Knoten in einem der folgenden Fälle neu gestartet:

- Patch Manager ist auf einem oder mehreren Patches installiert.

  Patch Manager wertet nicht aus, ob ein Neustart vom Patch erfordert wird. Das System wird neu gestartet, auch wenn der Patch keinen Neustart erfordert.

- Patch Manager erkennt ein oder mehrere Patches mit dem Status `INSTALLED_PENDING_REBOOT` während der `Install`-Operation.

  Der `INSTALLED_PENDING_REBOOT` Status kann bedeuten, dass die Option ausgewählt `NoReboot` wurde, als der `Install` Vorgang das letzte Mal ausgeführt wurde, oder dass ein Patch Patch Manager seit dem letzten Neustart des verwalteten Knotens außerhalb installiert wurde.

Durch den Neustart von verwalteten Knoten wird in diesen beiden Fällen sichergestellt, dass aktualisierte Pakete aus dem Speicher gelöscht werden und das Patch- und Neustartverhalten über alle Betriebssysteme hinweg konsistent bleibt.

NoReboot

Wenn Sie die Option `NoReboot` auswählen, startet Patch Manager einen verwalteten Knoten nicht neu, selbst wenn über ihn während der `Install`-Operation Patches installiert wurden. Diese Option ist nützlich, wenn Sie wissen, dass für Ihre verwalteten Knoten nach dem Anwenden von Patches kein Neustart erforderlich ist oder Anwendungen bzw. Prozesse auf einem Knoten ausgeführt werden, die nicht durch einen Neustart beim Patchen unterbrochen werden sollten. Sie ist auch nützlich, wenn Sie mehr Kontrolle über das Timing von Neustarts von verwalteten Knoten wünschen, z. B. durch die Verwendung eines Wartungsfensters.

> **Note**
>
> Wenn Sie die Option `NoReboot` auswählen und ein Patch installiert ist, wird dem Patch der Status `InstalledPendingReboot` zugewiesen. Der verwaltete Knoten selbst wird jedoch als `Non-Compliant` gekennzeichnet. Nach einem Neustart und Ausführung einer `Scan`-Operation wird der Knoten-Status in `Compliant` aktualisiert.

Datei zum Nachverfolgen der Patch-Installation: Um die Patch-Installation nachzuverfolgen, insbesondere von Patches, die seit dem letzten Neustart des Systems installiert wurden, erstellt Systems Manager eine Datei auf dem verwalteten Knoten.

> **Important**
>
> Löschen oder ändern Sie die Tracking-Datei nicht. Wenn diese Datei gelöscht oder beschädigt wird, ist der Patch-Compliance-Bericht für den verwalteten Knoten ungenau. Starten Sie in diesem Fall den Knoten neu und führen Sie eine Patch-Scan-Operation aus, um die Datei wiederherzustellen.

Diese Tracking-Datei wird an den folgenden Speicherorten auf Ihren verwalteten Knoten gespeichert:

- Linux-Betriebssysteme:
  - `/var/log/amazon/ssm/patch-configuration/patch-states-configuration.json`
  - `/var/log/amazon/ssm/patch-configuration/patch-inventory-from-last-operation.json`
- Windows Server-Betriebssystem:
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchStatesConfiguration.json`
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchInventoryFromLastOperation.json`

Parametername: **`PreInstallHookDocName`**

Nutzung: optional.

Standard: `AWS-Noop`.

Der Wert, der für den `PreInstallHookDocName`-Parameter anzugeben ist, ist der Name oder der Amazon-Ressourcenname (ARN) eines SSM-Dokuments Ihrer Wahl. Sie können den Namen eines AWS verwalteten Dokuments oder den Namen oder ARN eines benutzerdefinierten SSM-Dokuments angeben, das Sie erstellt haben oder das für Sie freigegeben wurde. (Für ein SSM-Dokument, das von einem anderen für Sie freigegeben wurde AWS-Konto, müssen Sie den vollständigen Ressourcen-ARN angeben, z. `arn:aws:ssm:us-east-2:123456789012:document/MySharedDocument` B.)

Das von Ihnen angegebene SSM-Dokument wird vor der `Install`-Operation ausgeführt und führt alle Aktionen aus, die von SSM Agent unterstützt werden, z. B. ein Shell-Skript, um die Zustandsprüfung der Anwendung zu überprüfen, bevor der verwaltete Knoten gepatcht wird. (Eine Liste der Aktionen finden Sie unter Referenz für Befehlsdokument-Plug-ins). Der SSM-Dokumentname ist standardmäßig `AWS-Noop`, was keine Operation für den verwalteten Knoten ausführt.

Informationen zum Erstellen eines benutzerdefinierten SSM-Dokuments finden Sie unter Erstellen von SSM-Dokumentinhalten.

Parametername: **`PostInstallHookDocName`**

Nutzung: optional.

Standard: `AWS-Noop`.

Der Wert, der für den `PostInstallHookDocName`-Parameter anzugeben ist, ist der Name oder der Amazon-Ressourcenname (ARN) eines SSM-Dokuments Ihrer Wahl. Sie können den Namen eines AWS verwalteten Dokuments oder den Namen oder ARN eines benutzerdefinierten SSM-Dokuments angeben, das Sie erstellt haben oder das für Sie freigegeben wurde. (Für ein SSM-Dokument, das von einem anderen für Sie freigegeben wurde AWS-Konto, müssen Sie den vollständigen Ressourcen-ARN angeben, z. `arn:aws:ssm:us-east-2:123456789012:document/MySharedDocument` B.)

Das von Ihnen angegebene SSM-Dokument wird nach der `Install with NoReboot`-Operation ausgeführt und führt alle Aktionen aus, die von SSM Agent unterstützt werden, z. B. ein Shell-Skript zum Installieren von Updates von Drittanbietern vor dem Neustart. (Eine Liste der Aktionen finden Sie unter Referenz für Befehlsdokument-Plug-ins). Der SSM-Dokumentname ist standardmäßig `AWS-Noop`, was keine Operation für den verwalteten Knoten ausführt.

Informationen zum Erstellen eines benutzerdefinierten SSM-Dokuments finden Sie unter Erstellen von SSM-Dokumentinhalten.

Parametername: **`OnExitHookDocName`**

Nutzung: optional.

Standard: `AWS-Noop`.

Der Wert, der für den `OnExitHookDocName`-Parameter anzugeben ist, ist der Name oder der Amazon-Ressourcenname (ARN) eines SSM-Dokuments Ihrer Wahl. Sie können den Namen eines AWS verwalteten Dokuments oder den Namen oder ARN eines benutzerdefinierten SSM-Dokuments angeben, das Sie erstellt haben oder das für Sie freigegeben wurde. (Für ein SSM-Dokument, das aus einem anderen AWS-Konto freigegeben wurde, müssen Sie den vollständigen Ressourcen-ARN angeben, z. B. `arn:aws:ssm:us-east-2:123456789012:document/MySharedDocument`.)

Das von Ihnen angegebene SSM-Dokument wird nach dem Neustart des verwalteten Knoten ausgeführt und führt alle Aktionen aus, die von SSM Agent unterstützt werden, z. B. ein Shell-Skript, um den Knoten-Zustand nach Abschluss des Patchvorgangs zu überprüfen. (Eine Liste der Aktionen finden Sie unter Referenz für Befehlsdokument-Plug-ins). Der SSM-Dokumentname ist standardmäßig `AWS-Noop`, was keine Operation für den verwalteten Knoten ausführt.

Informationen zum Erstellen eines benutzerdefinierten SSM-Dokuments finden Sie unter Erstellen von SSM-Dokumentinhalten.

Beispielszenario für die Verwendung des Parameters „InstallOverrideList“ in **`AWS-RunPatchBaseline`** oder **`AWS-RunPatchBaselineAssociation`**

Sie können den Parameter `InstallOverrideList` verwenden, wenn Sie die von der aktuellen Standard-Patch-Baseline in Patch Manager, einer Funktion von AWS Systems Manager, angegebenen Patches überschreiben möchten. In diesem Thema finden Sie Beispiele, die zeigen, wie Sie diesen Parameter verwenden, um Folgendes zu erreichen:

- Anwendung verschiedener Sätzen von Patches auf eine Zielgruppe von verwalteten Knoten.
- Anwendung dieser Patch-Sets auf verschiedene Häufigkeiten
- Verwendung derselben Patch-Baseline für beide Operationen

Angenommen, Sie möchten zwei verschiedene Kategorien von Patches auf Ihren von Amazon Linux 2 verwalteten Knoten installieren. Sie möchten diese Patches mithilfe von Wartungsfenstern nach verschiedenen Zeitplänen installieren. Sie möchten, dass jede Woche ein Wartungsfenster ausgeführt wird und alle `Security`-Patches installiert werden. Sie möchten, dass einmal im Monat

ein weiteres Wartungsfenster ausgeführt wird und dabei alle verfügbaren Patches oder Kategorien von Patches außer `Security` installiert werden.

Es kann jedoch nur jeweils eine Patch-Baseline als Standard für ein Betriebssystem definiert werden. Diese Anforderung hilft, Situationen zu vermeiden, in denen eine Patch-Baseline einen Patch genehmigt, während eine andere ihn blockiert, was zu Problemen zwischen in Konflikt stehenden Versionen führen kann.

Mit der folgenden Strategie können Sie den Parameter `InstallOverrideList` verwenden, um verschiedene Patch-Typen nach verschiedenen Zeitplänen auf eine Zielgruppe anzuwenden und dabei dennoch dieselbe Patch-Baseline zu verwenden:

1. Stellen Sie in der Standard-Patch-Baseline sicher, dass nur `Security`-Updates angegeben sind.
2. Erstellen Sie ein Wartungsfenster, das `AWS-RunPatchBaseline` oder `AWS-RunPatchBaselineAssociation` jede Woche ausführt. Geben Sie keine Überschreibungsliste an.
3. Erstellen Sie eine Überschreibungsliste der Patches aller Typen, die Sie monatlich anwenden möchten, und speichern Sie sie in einem Amazon Simple Storage Service (Amazon S3)-Bucket.
4. Erstellen Sie ein zweites Wartungsfenster, das einmal im Monat ausgeführt wird. Geben Sie für die Run Command-Aufgabe, die Sie für dieses Wartungsfenster registrieren, jedoch den Speicherort Ihrer Überschreibungsliste an.

Das Ergebnis: Jede Woche werden nur `Security`-Patches installiert, wie in Ihrer Standard-Patch-Baseline definiert. Die Installation aller verfügbaren Patches oder einer von Ihnen definierten Teilmenge von Patches erfolgt jeden Monat.

Weitere Informationen und Beispiellisten finden Sie unter Parametername: `InstallOverrideList`.

Verwenden des BaselineOverride Parameters

Sie können zur Laufzeit Einstellungen für Patches definieren, indem Sie die Funktion zum Überschreiben von Baselines in verwendenPatch Manager, eine Funktion von AWS Systems Manager. Geben Sie dazu einen Amazon Simple Storage Service (Amazon S3)-Bucket an, der ein JSON-Objekt mit einer Liste mit Patch-Baselines enthält. Beim Patchvorgang werden die im JSON-Objekt bereitgestellten Baselines verwendet, die mit dem Hostbetriebssystem übereinstimmen, anstatt die Regeln aus der Standard-Patch-Baseline anzuwenden.

> **Note**
>
> Außer wenn bei einem Patchvorgang eine Patch-Richtlinie verwendet wird, wird durch die Verwendung des `BaselineOverride` Parameters die Patch-Konformität der im Parameter angegebenen Baseline nicht überschrieben. Die Ausgabeergebnisse werden in den Stdout-Protokollen von aufgezeichnetRun Command, eine Fähigkeit von. AWS Systems Manager Die Ergebnisse drucken nur Pakete aus, die als `NON_COMPLIANT` gekennzeichnet sind. Das bedeutet, dass das Paket als `Missing`, `Failed`, `InstalledRejected` oder `InstalledPendingReboot` gekennzeichnet ist.
> Wenn ein Patch-Vorgang jedoch eine Patch-Richtlinie verwendet, übergibt das System den Override-Parameter aus dem zugehörigen S3-Bucket, und der Compliance-Wert wird für den verwalteten Knoten aktualisiert. Weitere Informationen zum Verhalten von Patch-Richtlinien finden Sie unterVerwenden von Quick Setup-Patch-Richtlinien.

Verwenden der Patch-Baseline-Überschreibung mit den Parametern „Snapshot-ID“ oder „Install Override List“

Es gibt zwei Fälle, in denen die Patch-Baseline-Überschreibung ein bemerkenswertes Verhalten aufweist.

Gleichzeitiges Verwenden von Baseline-Überschreiben und Snapshot-ID

Snapshot-IDs stellen sicher, dass alle verwalteten Knoten in einem bestimmten Patching-Befehl dasselbe anwenden. Wenn Sie beispielsweise 1 000 Knoten gleichzeitig patchen, sind die Patches identisch.

Wenn Sie sowohl eine Snapshot-ID als auch eine Patch-Baseline-Überschreibung verwenden, hat die Snapshot-ID Vorrang vor der Patch-Baseline-Überschreibung. Die Baseline-Überschreibungsregeln werden weiterhin verwendet, aber sie werden nur einmal ausgewertet. Im vorangegangenen Beispiel sind die Patches für Ihre 1 000 verwaltete Knoten immer gleich. Wenn Sie in der Mitte des Patching-Vorgangs die JSON-Datei im referenzierten S3-Bucket auf etwas anderes geändert haben, sind die angewendeten Patches immer noch identisch. Dies liegt daran, dass die Snapshot-ID bereitgestellt wurde.

Gleichzeitiges Verwenden der Baseline-Überschreibung und des Parameters „Override List“

Sie können diese beiden Parameter nicht gleichzeitig verwenden. Das Patching-Dokument schlägt fehl, wenn beide Parameter angegeben sind, und es führt keine Scans oder Installationen auf dem verwalteten Knoten durch.

Codebeispiele

Das folgende Codebeispiel für Python zeigt, wie die Patch-Baseline-Überschreibung generiert wird.

```
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)
```

So wird eine Patch-Baseline-Überschreibung wie die folgende erstellt.

```
[
    {
        "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": []
    }
]
```

Über Patch-Baselines

In den Themen in diesem Abschnitt finden Sie Informationen zur Funktionsweise von Patch-Baselines in Patch Manager, einer Funktion von AWS Systems Manager, wenn Sie eine `Scan`- oder `Install`-Operation auf Ihren verwalteten Knoten ausführen.

Themen

- Info zu vordefinierten und benutzerdefinierten Patch-Baselines

- Paketnamen-Formate für Listen genehmigter und abgelehnter Patches
- Patch-Gruppen
- Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden

Info zu vordefinierten und benutzerdefinierten Patch-Baselines

Patch Manager, eine Funktion von AWS Systems Manager, bietet vordefinierte Patch-Baselines für jedes der von unterstützten BetriebssystemePatch Manager. Sie können diese Baselines in ihrer aktuellen Konfiguration verwenden (eine Anpassung ist nicht möglich) oder eigene benutzerdefinierte Patch-Baselines erstellen. Benutzerdefinierte Patch-Baselines ermöglichen Ihnen eine bessere Kontrolle darüber, welche Patches für Ihre Umgebung genehmigt oder abgelehnt werden. Außerdem weisen die vordefinierten Baselines allen Patches, die mit diesen Baselines installiert wurden, die Compliance-Ebene `Unspecified` zu. Für die Zuweisung von Compliance-Werten können Sie eine Kopie einer vordefinierten Baseline erstellen und die Compliance-Werte angeben, die Patches zugewiesen werden sollen. Weitere Informationen finden Sie unter Info zu benutzerdefinierten Baselines und Arbeiten mit benutzerdefinierten Patch-Baselines.

> **Note**
>
> Die Informationen in diesem Thema gelten unabhängig davon, welche Methode oder Art der Konfiguration Sie für Ihren Patching-Vorgang verwenden:
>
> - Eine in Quick Setup konfigurierte Patch-Richtlinie
> - Eine in Quick Setup konfigurierte Host-Management-Option
> - Ein Wartungsfenster zum Ausführen eines Patch-`Scans` oder einer `Install`-Aufgabe
> - Eine On-Demand Patch now (Jetzt patchen)-Operation

Themen

- Info zu vordefinierten Baselines
- Info zu benutzerdefinierten Baselines

Info zu vordefinierten Baselines

Die folgende Tabelle beschreibt die mit Patch Manager bereitgestellten vordefinierten Patch-Baselines.

Informationen dazu, welche Versionen der einzelnen Betriebssysteme von Patch Manager unterstützt werden, finden Sie unter Patch Manager-Voraussetzungen.

Name	Unterstütztes Betriebssystem	Details
`AWS-AlmaLinuxDefaultPatchBaseline`	AlmaLinux	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Genehmigt außerdem alle Patches mit der Klassifizierung „Bugfix“. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]
`AWS-AmazonLinuxDefaultPatchBaseline`	Amazon Linux 1	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Genehmigt außerdem automatisch alle Patches mit der Klassifizierung „Bugfix“. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]
`AWS-AmazonLinux2DefaultPatchBaseline`	Amazon Linux 2	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und

Name	Unterstütztes Betriebssystem	Details
		dem Schweregrad „Critical“ oder „Important“. Genehmigt außerdem alle Patches mit der Klassifizierung „Bugfix“. Patches werden automatisch sieben Tage nach der Veröffentlichung genehmigt.[1]
`AWS-AmazonLinux2022DefaultPatchBaseline`	Amazon Linux 2022	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Patches werden automatisch sieben Tage nach der Veröffentlichung genehmigt. Genehmigt außerdem alle Patches mit einer Klassifizierung „Bugfix“ sieben Tage nach der Veröffentlichung.
`AWS-AmazonLinux2023DefaultPatchBaseline`	Amazon Linux 2023	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Patches werden automatisch sieben Tage nach der Veröffentlichung genehmigt. Genehmigt außerdem alle Patches mit einer Klassifizierung „Bugfix“ sieben Tage nach der Veröffentlichung.

Name	Unterstütztes Betriebssystem	Details
`AWS-CentOSDefaultPatchBaseline`	CentOS und CentOS Stream	Genehmigt alle Aktualisierungen sieben Tage nach ihrer Verfügbarkeit, einschließlich nicht sicherheitsrelevanter Aktualisierungen.
`AWS-DebianDefaultPatchBaseline`	Debian Server	Genehmigt sofort alle sicherheitsrelevanten Patches für Betriebssysteme mit der Priorität „Required“, „Important“, „Standard“, „Optional“ oder „Extra“. Die Genehmigung erfolgt unverzüglich, weil in den Repositorys keine zuverlässigen Datumsangaben zur Veröffentlichung verfügbar sind.
`AWS-MacOSDefaultPatchBaseline`	macOS	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“. Genehmigt auch alle Pakete mit einem aktuellen Update.

Name	Unterstütztes Betriebssystem	Details
`AWS-OracleLinuxDefaultPatchBaseline`	Oracle Linux	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Important“ oder „Moderate“. Genehmigt außerdem alle als „Bugfix“ eingestuften Patches 7 Tage nach Veröffentlichung. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]
`AWS-DefaultRaspbianPatchBaseline`	Raspberry Pi OS	Genehmigt sofort alle sicherheitsrelevanten Patches für Betriebssysteme mit der Priorität „Required“, „Important“, „Standard“, „Optional“ oder „Extra“. Die Genehmigung erfolgt unverzüglich, weil in den Repositorys keine zuverlässigen Datumsangaben zur Veröffentlichung verfügbar sind.

Name	Unterstütztes Betriebssystem	Details
`AWS-RedHatDefaultPatchBaseline`	Red Hat Enterprise Linux (RHEL)	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Genehmigt außerdem alle Patches mit der Klassifizierung „Bugfix“. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]
`AWS-RockyLinuxDefaultPatchBaseline`	Rocky Linux	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Genehmigt außerdem alle Patches mit der Klassifizierung „Bugfix“. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]
`AWS-SuseDefaultPatchBaseline`	SUSE Linux Enterprise Server (SLES)	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]

Name	Unterstütztes Betriebssystem	Details
`AWS-UbuntuDefaultPatchBaseline`	Ubuntu Server	Genehmigt sofort alle sicherheitsrelevanten Patches für Betriebssysteme mit der Priorität „Required“, „Important“, „Standard“, „Optional“ oder „Extra“. Die Genehmigung erfolgt unverzüglich, weil in den Repositorys keine zuverlässigen Datumsangaben zur Veröffentlichung verfügbar sind.
`AWS-DefaultPatchBaseline`	Windows Server	Genehmigt alle Windows Server Betriebssystem-Patches, die als „CriticalUpdates“ oder „SecurityUpdates“ klassifiziert sind und den MSRC-Schweregrad „Kritisch“ oder „Wichtig“ haben. Patches werden 7 Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[2]
`AWS-WindowsPredefinedPatchBaseline-OS`	Windows Server	Genehmigt alle Windows Server Betriebssystem-Patches, die als „CriticalUpdates“ oder „SecurityUpdates“ klassifiziert sind und den MSRC-Schweregrad „Kritisch“ oder „Wichtig“ haben. Patches werden 7 Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[2]

Name	Unterstütztes Betriebssystem	Details
`AWS-WindowsPredefinedPatchBaseline-OS-Applications`	Windows Server	Genehmigt für das Windows Server Betriebssystem alle Patches, die als „Critical Updates“ oder „Security Updates“ klassifiziert sind und den MSRC-Schweregrad „Kritisch“ oder „Wichtig“ haben. Genehmigt für von Microsoft veröffentlichte Anwendungen alle Patches. Patches für Betriebssysteme und Anwendungen werden 7 Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[2]

1 Für Amazon Linux 1 und Amazon Linux 2 wird die 7-tägige Wartezeit, bis Patches automatisch genehmigt werden`updateinfo.xml`, anhand eines -`Updated Date`Werts in und nicht anhand eines -`Release Date`Werts berechnet. Verschiedene Faktoren können den `Updated Date`-Wert beeinflussen. Andere Betriebssysteme behandeln Veröffentlichungs- und Aktualisierungsdaten unterschiedlich. Informationen dazu, wie Sie unerwartete Ergebnisse durch Verzögerungen bei der automatischen Genehmigung vermeiden können, finden Sie unter So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet.

[2] Für Windows Server enthalten die Standard-Baselines eine Verzögerung von 7 Tagen für die automatische Genehmigung. Um einen Patch innerhalb von 7 Tagen nach der Veröffentlichung zu installieren, müssen Sie eine benutzerdefinierte Baseline erstellen.

Info zu benutzerdefinierten Baselines

Wenn Sie eine eigene Patch-Baseline herstellen, können Sie die Patches wahlweise automatisch genehmigen, indem Sie die folgenden Kategorien verwenden.

- Betriebssystem: Windows Server, Amazon Linux, Ubuntu Server usw.

- Produktname (für Betriebssysteme): Beispielsweise RHEL 6.5, Amazon Linux 2014.09, Windows Server 2012, Windows Server 2012 R2 usw.
- Produktname (Windows Servernur für Anwendungen, die von Microsoft auf veröffentlicht wurden): Zum Beispiel Word 2016, BizTalk Server usw.
- Klassifizierung: Beispielsweise kritische Updates, Sicherheitsupdates usw.
- Schweregrad: Beispielsweise kritisch, wichtig usw.

Für jede von Ihnen erstellte Genehmigungsregel können Sie eine Verzögerung für die automatische Genehmigung oder ein Stichdatum für die Patch-Genehmigung angeben.

> **Note**
>
> Da es nicht möglich ist, die Veröffentlichungsdaten von Updatepaketen für Ubuntu Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

Eine Verzögerung der automatischen Genehmigung ist die Anzahl an Tagen, die gewartet werde soll, nachdem die Patch veröffentlicht oder zuletzt aktualisiert wurde, bevor der Patch automatisch genehmigt wird. Wenn Sie beispielsweise eine Regel mit der `CriticalUpdates`-Klassifizierung erstellen und für sie für eine Verzögerung der automatischen Genehmigung von sieben Tagen konfigurieren, wird ein neuer kritischer Patch, der am 7. Juli veröffentlicht wird, am 14. Juli automatisch genehmigt.

> **Note**
>
> Wenn ein Linux-Repository keine Informationen zum Veröffentlichungsdatum für Pakete bereitstellt, verwendet Systems Manager die Build-Zeit des Pakets als Verzögerung bei der automatischen Genehmigung für Amazon Linux 1, Amazon Linux 2RHEL, und CentOS . Wenn das System nicht in der Lage ist, den Buildzeitpunkt des Pakets zu ermitteln, verwendet Systems Manager für die Festlegung der Verzögerung bis zur automatischen Genehmigung den Wert Null.

Wenn Sie ein Stichdatum für die automatische Genehmigung angeben, wendet Patch Manager automatisch alle Patches an, die an oder vor diesem Datum veröffentlicht oder zuletzt aktualisiert wurden. Wenn Sie beispielsweise den 07. Juli 2023 als Stichtag angeben, werden keine Patches

automatisch installiert, die an oder nach dem 08. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Patch-Baseline erstellen, können Sie für Patches, die von dieser Patch-Baseline genehmigt wurden, einen Schweregrad für die Konformität angeben, beispielsweise `Critical` oder `High`. Wenn der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann ist der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline der von Ihnen angegebene Schweregrad.

Beachten Sie bei der Erstellung einer Patch-Baseline Folgendes:

- Patch Manager stellt eine vordefinierte Patch-Baseline für jedes unterstützte Betriebssystem bereit. Diese vordefinierten Patch-Baselines werden als Standard-Patch-Baselines für alle Betriebssystemtypen verwendet, wenn Sie nicht eigene Patch-Baselines erstellen und diese als Standard für den jeweiligen Betriebssystemtyp festlegen.

  > **Note**
  >
  > Für Windows Server werden drei vordefinierte Patch-Baselines bereitgestellt. Die Patch-Baselines `AWS-DefaultPatchBaseline` und `AWS-WindowsPredefinedPatchBaseline-OS` unterstützen nur Betriebssystemupdates auf dem Windows-Betriebssystem selbst. `AWS-DefaultPatchBaseline` wird als Standard-Patch-Baseline für von Windows Server verwalteten Knoten verwendet, es sei denn, Sie geben eine andere Patch-Baseline an. Die Konfigurationseinstellungen in diesen beiden Patch-Baselines sind identisch. Die neuere der beiden, `AWS-WindowsPredefinedPatchBaseline-OS`, wurde erstellt, um sie von der dritten vordefinierten Patch-Baseline für Windows Server zu unterscheiden. Diese Patch-Baseline, `AWS-WindowsPredefinedPatchBaseline-OS-Applications`, kann verwendet werden, um Patches sowohl auf das Windows Server-Betriebssystem als auch auf unterstützte Anwendungen, die von Microsoft veröffentlicht wurden, anzuwenden.

- Bei On-Premises-Servern und virtuellen Maschinen (VMs) versucht Patch Manager, Ihre benutzerdefinierte Standard-Patch-Baseline zu verwenden. Wenn keine benutzerdefinierte Standard-Patch-Baseline vorhanden ist, verwendet das System die vordefinierte Patch-Baseline für das entsprechende Betriebssystem.

- Wenn ein Patch sowohl als genehmigt als auch als abgelehnt aufgelistet ist, wird der Patch abgelehnt.
- Für einen verwalteten Knoten kann nur eine einzige Patch-Baseline definiert werden.
- Die Formate der Paketnamen, die Sie zu den Listen der genehmigten und abgelehnten Patches für eine Patch-Baseline hinzufügen können, hängen von der Art des Betriebssystems ab, das gepatcht wird.

  Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.
- Wenn Sie eine Patch-Richtlinienkonfiguration in Quick Setup verwenden, werden Aktualisierungen, die Sie an benutzerdefinierten Patch-Baselines vornehmen, einmal pro Stunde mit Quick Setup synchronisiert.

  Wenn eine benutzerdefinierte Patch-Baseline gelöscht wird, auf die in einer Patch-Richtlinie verwiesen wurde, wird auf der Seite mit den Quick Setup-Configuration details (Konfigurationsdetails) ein Banner für Ihre Patch-Richtlinie angezeigt. Das Banner informiert Sie darüber, dass die Patch-Richtlinie auf eine nicht mehr vorhandene Patch-Baseline verweist und nachfolgende Patching-Vorgänge fehlschlagen werden. Kehren Sie in diesem Fall zur Seite Quick Setup-Configurations (Konfigurationen) zurück, wählen Sie die Patch Manager-Konfiguration aus und wählen Sie Actions (Aktionen), Edit configuration (Konfiguration bearbeiten). Der Name der gelöschten Patch-Baseline wird hervorgehoben, und Sie müssen eine neue Patch-Baseline für das betroffene Betriebssystem auswählen.

Informationen zum Erstellen einer Patch-Baseline finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines und Anleitung: Patchen einer Serverumgebung (AWS CLI).

Paketnamen-Formate für Listen genehmigter und abgelehnter Patches

Die Formate der Paketnamen, die Sie zu den Listen der genehmigten und abgelehnten Patches hinzufügen können, hängen von der Art des Betriebssystems ab, das gepatcht wird.

Paketnamen-Formate für Linux-Betriebssysteme

Die Formate, die Sie für genehmigte und abgelehnte Patches in der Patch-Baseline festlegen können, variieren je nach Linux-Typ. Genauer gesagt hängen die unterstützten Formate von dem Paket-Manager ab, der vom Linux-Betriebssystemtyp verwendet wird.

Themen

- Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, Amazon Linux 2023, CentOS , Oracle Linuxund Red Hat Enterprise Linux (RHEL)
- Debian Server, Raspberry Pi OS (früher Raspbian) und Ubuntu Server
- SUSE Linux Enterprise Server (SLES)

Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, Amazon Linux 2023, CentOS , Oracle Linuxund Red Hat Enterprise Linux (RHEL)

Paketmanager: YUM, außer Amazon Linux 2022, Amazon Linux 2023, RHEL 8 und CentOS 8, die DNF als Paketmanager verwenden

Genehmigte Patches: Für genehmigte Patches können Sie Folgendes festlegen:

- Bugzilla-IDs im Format `1234567` (Das System verarbeitet aus Zahlen bestehende Zeichenfolgen als Bugzilla-IDs.)
- CVE-IDs im Format `CVE-2018-1234567`
- Advisory-IDs in Formaten wie `RHSA-2017:0864` und `ALAS-2018-123`
- Vollständige Paketnamen in Formaten wie z. B.:
  - `example-pkg-0.710.10-2.7.abcd.x86_64`
  - `pkg-example-EE-20180914-2.2.amzn1.noarch`
- Paketnamen mit einem einzigen Platzhalter in Formaten wie z. B.:
  - `example-pkg-*.abcd.x86_64`
  - `example-pkg-*-20180914-2.2.amzn1.noarch`
  - `example-pkg-EE-2018*.amzn1.noarch`

Abgelehnte Patches: Für abgelehnte Patches können Sie Folgendes festlegen:

- Vollständige Paketnamen in Formaten wie z. B.:
  - `example-pkg-0.710.10-2.7.abcd.x86_64`
  - `pkg-example-EE-20180914-2.2.amzn1.noarch`
- Paketnamen mit einem einzigen Platzhalter in Formaten wie z. B.:
  - `example-pkg-*.abcd.x86_64`
  - `example-pkg-*-20180914-2.2.amzn1.noarch`
  - `example-pkg-EE-2018*.amzn1.noarch`

Debian Server, Raspberry Pi OS (früher Raspbian) und Ubuntu Server

Paket-Manager: APT

Genehmigte Patches und abgelehnte Patches: Legen Sie für genehmigte sowie abgelehnte Patches Folgendes fest:

- Paketnamen im Format `ExamplePkg33`

> **Note**
>
> Verwenden Sie für Debian Server-Listen, Raspberry Pi OS-Listen und Ubuntu Server-Listen keine Elemente wie Architektur oder Versionen. Beispiel: Sie legen den Paketnamen `ExamplePkg33` fest, um alles Folgende in einer Patch-Liste einzubeziehen:
>
> - `ExamplePkg33.x86.1`
> - `ExamplePkg33.x86.2`
> - `ExamplePkg33.x64.1`
> - `ExamplePkg33.3.2.5-364.noarch`

SUSE Linux Enterprise Server (SLES)

Paket-Manager: Zypper

Genehmigte Patches und abgelehnte Patches: Sie können für genehmigte sowie abgelehnte Patch-Listen Folgendes festlegen:

- Vollständige Paketnamen in Formaten wie z. B.:
  - `SUSE-SLE-Example-Package-12-2018-123`
  - `example-pkg-2018.11.4-46.17.1.x86_64.rpm`
- Paketnamen mit einem einzigen Platzhalter wie z. B.:
  - `SUSE-SLE-Example-Package-12-2018-*`
  - `example-pkg-2018.11.4-46.17.1.*.rpm`

Paketnamen-Formate für macOS

Unterstützte Paketmanager: Softwareupdate, Installationsprogramm, Brew, Brew Cask

Genehmigte Patches und abgelehnte Patches: Geben Sie für genehmigte sowie abgelehnte Patch-Listen vollständige Paketnamen in folgenden formaten an:

- `XProtectPlistConfigData`
- `MRTConfigData`

Platzhalter werden in Listen genehmigter und abgelehnter Patches für macOS nicht unterstützt.

Paketnamen-Formate für Windows-Betriebssysteme

Legen Sie für Windows-Betriebssysteme Patches mit den Microsoft Knowledge Base-IDs und Microsoft Security Bulletins-IDs fest, z. B.:

```
KB2032276,KB2124261,MS10-048
```

Patch-Gruppen

> **Important**
>
> Patch-Gruppen werden nicht in Patch-Vorgängen verwendet, die auf Patch-Richtlinien basieren. Weitere Informationen zur Arbeit mit Patch-Richtlinien finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien.

Sie können eine Patchgruppe verwenden, um verwaltete Knoten einer bestimmten Patch-Baseline zuzuordnenPatch Manager, eine Fähigkeit von AWS Systems Manager. Mit Patch-Gruppen können Sie sicherstellen, dass Sie geeignete Patches basierend auf den zugeordneten Patch-Baseline-Regeln für die richtigen Sätze von Knoten bereitstellen. Patch-Gruppen können außerdem dazu beitragen, die Bereitstellung von Patches zu vermeiden, bevor diese angemessen getestet sind. So können Sie Patch-Gruppen beispielsweise für unterschiedliche Umgebungen (Entwicklung, Test und Produktion) erstellen und jede Patch-Gruppe für eine geeignete Patch-Baseline registrieren.

Wenn Sie `AWS-RunPatchBaseline` ausführen, können Sie verwaltete Knoten über deren ID oder Tags anvisieren. Basierend auf dem Patch-Gruppenwert, den Sie dem verwalteten Knoten hinzugefügt haben, werten dann SSM Agent und Patch Manager aus, welche Patch-Baseline verwendet werden soll.

Sie erstellen eine Patch-Gruppe mit Amazon Elastic Compute Cloud (Amazon EC2)-Tags. Im Gegensatz zu anderen Anwendungsszenarien für Tags in Systems Manager muss eine Patch-

Gruppe mit dem entweder einem Tag-Schlüssel `Patch Group` oder `PatchGroup` definiert werden. Bei dem Schlüssel wird die Groß-/Kleinschreibung berücksichtigt. Sie können einen beliebigen Wert angeben, um die Ressourcen in dieser Gruppe zu identifizieren und darauf auszurichten, z. B. „Webserver“ oder „US-EAST-PROD“, aber der Schlüssel muss `Patch Group` oder `PatchGroup` sein.

Wenn Sie eine Patch-Gruppe erstellt und verwaltete Knoten mit Tags markiert haben, können Sie die Patch-Gruppe für eine Patch-Baseline anmelden. Indem Sie die Patch-Gruppe für eine Patch-Baseline registrieren, stellen Sie sicher, dass die Knoten innerhalb der Patch-Gruppe die in der zugehörigen Patch-Baseline definierten Regeln befolgen.

Weitere Informationen zum Erstellen von Patch-Gruppen und Zuordnen von Patch-Gruppen zu einer Patch-Baseline finden Sie unter Arbeiten mit Patch-Gruppen und Einer Patch-Baseline eine Patch-Gruppe hinzufügen.

Ein Beispiel für das Erstellen einer Patch-Baseline und von Patch-Gruppen über die AWS Command Line Interface (AWS CLI) finden Sie unter Anleitung: Patchen einer Serverumgebung (AWS CLI). Weitere Informationen zu Amazon EC2-Tags finden Sie unter Taggen Ihrer Amazon EC2-Ressourcen im Amazon EC2 EC2-Benutzerhandbuch.

Funktionsweise

Wenn das System eine Patch-Baseline auf einen verwalteten Knoten anwendet, überprüft SSM Agent, ob für den Knoten ein Patch-Gruppenwert definiert wurde. Wenn der Knoten einer Patch-Gruppe zugewiesen wurde, ermittelt Patch Manager anschließend, welche Patch-Baseline für diese Gruppe registriert wurde. Wenn für die Gruppe eine Patch-Baseline gefunden wird, weist Patch Manager SSM Agent an, die zugehörige Patch-Baseline zu verwenden. Wenn ein Knoten nicht für eine Patch-Gruppe konfiguriert wurde, weist Patch Manager SSM Agent automatisch an, die aktuell konfigurierte Standard-Patch-Baseline zu verwenden.

> **Important**
>
> Ein verwalteter Knoten kann sich nur in einer Patch-Gruppe befinden.
> Eine Patch-Gruppe kann nur für eine Patch-Baseline für jeden Betriebssystemtyp registriert werden.
> Sie können das `Patch Group`-Tag (mit einem Leerzeichen) nicht auf eine Amazon-EC2-Instance anwenden, wenn die Option Allow tags in instance metadata (Tags in Instance-Metadaten zulassen) auf der Instance aktiviert ist. Durch das Zulassen von Tags in Instance-Metadaten wird verhindert, dass Tag-Schlüsselnamen Leerzeichen enthalten. Wenn Sie Tags

in EC2-Instance-Metadaten zugelassen haben, müssen Sie den Tag-Schlüssel `PatchGroup` (ohne Leerzeichen) verwenden.

In der folgenden Abbildung sehen Sie ein allgemeines Beispiel der Prozesse, die Systems Manager beim Senden einer Run Command-Aufgabe an Ihre Serverflotte sendet, um mit Patch Manager Patches einzuspielen. Ein ähnlicher Prozess wird verwendet, wenn ein Wartungsfenster konfiguriert wurde, um einen Befehl zum Patchen mithilfe von Patch Manager zu senden.

In diesem Beispiel haben wir drei Gruppen von EC2-Instances für Windows Server mit den folgenden Tags:

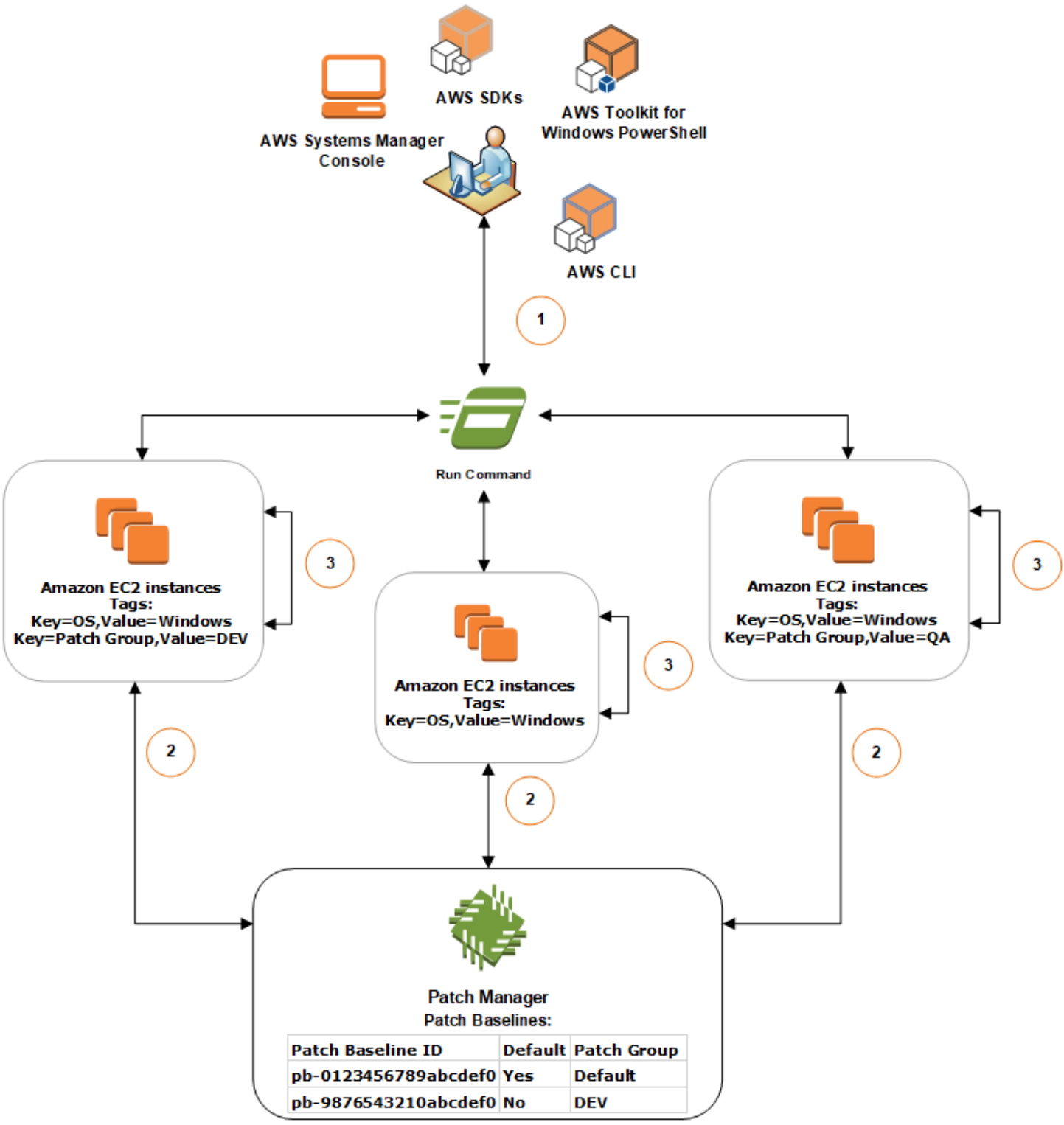
EC2-Instances-Gruppe	Tags
Gruppe 1	`key=OS,value=Windows` `key=PatchGroup,value=DEV`
Gruppe 2	`key=OS,value=Windows`
Gruppe 3	`key=OS,value=Windows` `key=PatchGroup,value=QA`

In diesem Beispiel haben wir außerdem diese beiden Windows Server-Patch-Baselines:

Patch-Baseline-ID	Standard	Zugehörige Patch-Gruppe
`pb-0123456789abcdef0`	Ja	`Default`
`pb-9876543210abcdef0`	Nein	`DEV`

Abbildung 1: Allgemeines Beispiel für den Prozessablauf beim Patch-Vorgang

Das folgende Diagramm zeigt, wie Patch Manager bestimmt, welche Patch-Baselines für Patch-Vorgänge verwendet werden sollen.



Der allgemeine Ablauf zum Scannen bzw. Installieren von Patches mit Run Command, einer Funktion von AWS Systems Manager, und Patch Manager sieht wie folgt aus:

1. Einen Befehl an den Patch senden: Verwenden Sie die Systems Manager Manager-Konsole, SDK, AWS Command Line Interface (AWS CLI), oder AWS Tools for Windows PowerShell um eine Run Command Aufgabe mithilfe des Dokuments zu sendenAWS-RunPatchBaseline. Die Abbildung zeigt eine Run Command-Aufgabe zum Patchen verwalteter Instances mit dem Tag key=OS,value=Windows als Ziel.
2. Patch-Baseline bestimmen: SSM Agent überprüft die auf die EC2-Instance angewendeten Patch-Gruppen-Tags und sendet eine Anfrage an Patch Manager für die entsprechende Patch-Baseline.
   - Passender Patch-Gruppenwert einer Patch-Baseline zugeordnet:
     1. SSM Agent, das auf EC2-Instance in Gruppe 1 installiert ist, empfängt den in Schritt 1 gesendeten Befehl, mit dem Patchvorgang zu beginnen. SSM Agent überprüft, ob die EC2-Instances über den Patch-Gruppen-Tag-Wert DEV verfügen und sendet eine Anfrage an Patch Manager für die zugehörige Patch-Baseline.
     2. Patch Manager überprüft, ob der Patch-Baseline pb-9876543210abcdef0 die Patch-Gruppe DEV zugeordnet ist, und sendet eine Benachrichtigung an SSM Agent.
     3. SSM Agent ruft basierend auf den in Patch Manager konfigurierten Genehmigungsregeln und Ausnahmen einen Snapshot der Patch-Baseline von pb-9876543210abcdef0 ab und fährt mit dem nächsten Schritt fort.
   - Instance verfügt nicht über ein Patch-Gruppen-Tag:
     1. SSM Agent, das auf EC2-Instance in Gruppe 2 installiert ist, empfängt den in Schritt 1 gesendeten Befehl, mit dem Patchvorgang zu beginnen. SSM Agent überprüft, ob die EC2-Instances nicht über das Patch-Gruppen-Tag Patch Group oder PatchGroup verfügen. SSM Agent sendet daraufhin eine Anfrage an Patch Manager für die Standard-Windows-Patch-Baseline.
     2. Patch Manager überprüft, ob die Standard-Patch-Baseline für Windows Server pb-0123456789abcdef0 ist, und benachrichtigt SSM Agent.
     3. SSM Agent ruft basierend auf den in der Standard-Patch-Baseline Patch Manager konfigurierten Genehmigungsregeln und Ausnahmen einen Snapshot der Patch-Baseline von pb-0123456789abcdef0 ab und fährt mit dem nächsten Schritt fort.
   - Es gibt keinen passenden, einer Patch-Baseline zugeordneten Patch-Gruppenwert:
     1. SSM Agent, das auf EC2-Instance in Gruppe 3 installiert ist, empfängt den in Schritt 1 gesendeten Befehl, mit dem Patchvorgang zu beginnen. SSM Agent überprüft, ob die EC2-Instances über den Patch-Gruppen-Tag-Wert QA verfügen und sendet eine Anfrage an Patch Manager für die zugehörige Patch-Baseline.

2. Patch Manager findet keine Patch-Baseline mit der zugeordneten Patch-Gruppe `QA`.
3. Patch Manager benachrichtigt SSM Agent, die Standard-Patch-Baseline für Windows, `pb-0123456789abcdef0`, zu verwenden.
4. SSM Agent ruft basierend auf den in der Standard-Patch-Baseline Patch Manager konfigurierten Genehmigungsregeln und Ausnahmen einen Snapshot der Patch-Baseline von `pb-0123456789abcdef0` ab und fährt mit dem nächsten Schritt fort.

3. Auf Patches scannen oder Patches installieren: Nachdem die anzuwendende Patch-Baseline bestimmt wurde, beginnt SSM Agent basierend auf dem in Schritt 1 festgelegten Vorgangswert entweder damit, nach Patches zu scannen oder diese zu installieren. Nach welchen Patches gescannt bzw. welche Patches installiert werden, wird durch die Genehmigungsregeln und Patch-Ausnahmen bestimmt, die im von Patch Manager bereitgestellten Patch-Baseline-Snapshot definiert sind.

Weitere Informationen

- Grundlegendes zu Patch-Compliance-Statuswerten

Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden

Verwenden Sie die Informationen in diesem Thema, um die Vorbereitung auf Patchanwendungen auf Windows Server mit Patch Manager, einer Funktion von AWS Systems Manager, zu erleichtern.

Patching von Microsoft-Anwendungen

Patching-Support für Anwendungen auf von Windows Server verwalteten Knoten ist auf Anwendungen beschränkt, die von Microsoft veröffentlicht werden.

> **Note**
>
> In einigen Fällen veröffentlicht Microsoft Patches für Anwendungen, die kein aktualisiertes Datum und keine aktualisierte Uhrzeit angeben. In diesen Fällen wird ein aktualisiertes Datum und eine Uhrzeit von `01/01/1970` standardmäßig angegeben.

Patch-Baselines, um von Microsoft veröffentlichte Anwendungen zu patchen

Für Windows Server werden drei vordefinierte Patch-Baselines bereitgestellt. Die Patch-Baselines `AWS-DefaultPatchBaseline` und `AWS-WindowsPredefinedPatchBaseline-OS` unterstützen nur Betriebssystemupdates auf dem Windows-Betriebssystem selbst. `AWS-DefaultPatchBaseline` wird als Standard-Patch-Baseline für von Windows Server verwalteten Knoten verwendet, es sei denn, Sie geben eine andere Patch-Baseline an. Die Konfigurationseinstellungen in diesen beiden Patch-Baselines sind identisch. Die neuere der beiden, `AWS-WindowsPredefinedPatchBaseline-OS`, wurde erstellt, um sie von der dritten vordefinierten Patch-Baseline für Windows Server zu unterscheiden. Diese Patch-Baseline, `AWS-WindowsPredefinedPatchBaseline-OS-Applications`, kann verwendet werden, um Patches sowohl auf das Windows Server-Betriebssystem als auch auf unterstützte Anwendungen, die von Microsoft veröffentlicht wurden, anzuwenden.

Sie können zum Aktualisieren von Anwendungen, die von Microsoft veröffentlicht wurden, auf Windows Server-Computern auch benutzerdefinierte Patch-Baselines erstellen.

Support für Patch-Anwendungen, die von Microsoft auf On-Premises-Servern, Edge-Geräten, VMs und anderen Nicht-EC2-Knoten veröffentlicht wurden

Aktivieren Sie das Advanced-Instances-Kontingent, um Anwendungen, die von Microsoft auf virtuellen Maschinen (VMs) und anderen nicht EC2-verwalteten Knoten veröffentlicht wurden, zu patchen. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Für Patch-Anwendungen, die von Microsoft auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances veröffentlicht wurden, fallen jedoch keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter Konfigurieren von Instance-Kontingenten.

Windows-Update-Option für „andere Microsoft-Produkte“

Damit Patch Manager von Microsoft auf Ihren von Windows Server verwalteten Knoten veröffentlichte Anwendungen patchen kann, muss die Windows-Update-Option Ich möchte Updates für andere Microsoft-Produkte erhalten, wenn ich Windows aktualisiere auf dem verwalteten Knoten aktiviert sein.

Informationen zum Zulassen dieser Option für einen einzelnen verwalteten Knoten finden Sie unter Aktualisieren von Office mit Microsoft Update auf der Microsoft-Support-Website.

Bei einer Flotte von verwalteten Knoten auf Windows Server 2016 und höher können Sie die Einstellung mithilfe eines Group Policy Object (GPO, Gruppenrichtlinienobjekt) aktivieren. Navigieren Sie im Gruppenrichtlinien-Verwaltungseditor zu Computer-Konfiguration, Administrative Vorlagen, Windows-Komponenten, Windows-Updates und wählen Sie Installieren von Updates für andere

Microsoft-Produkte aus. Wir empfehlen außerdem, das GPO mit zusätzlichen Parametern zu konfigurieren, die ungeplante automatische Updates und Neustarts außerhalb von Patch Manager verhindern. Weitere Informationen finden Sie unter Konfigurieren automatischer Updates in einer Umgebung ohne Active Directory auf der Website für technische Dokumentation von Microsoft.

Bei einer Flotte von verwalteten Knoten, die auf Windows Server 2012 oder 2012 R2 ausgeführt werden, können Sie die Option mithilfe eines Skripts aktivieren, wie unter Aktivieren und Deaktivieren von Microsoft Update in Windows 7 über Skript auf der Microsoft-Docs-Blog-Website beschrieben. Sie können z. B. Folgendes tun:

1. Speichern Sie das Skript aus dem Blogbeitrag in einer Datei.
2. Laden Sie die Datei in einen Amazon Simple Storage Service (Amazon S3)-Bucket oder an einem anderen zugänglichen Speicherort hoch.
3. Verwenden Sie Run Command, eine Funktion von AWS Systems Manager, um das Skript auf Ihren verwalteten Knoten mithilfe des Systems-Manager-Dokuments (SSM-Dokument) `AWS-RunPowerShellScript` mit einem dem folgenden ähnlichen Befehl auszuführen.

```
Invoke-WebRequest `
    -Uri "https://s3.aws-api-domain/DOC-EXAMPLE-BUCKET/script.vbs" `
    -Outfile "C:\script.vbs" cscript c:\script.vbs
```

Mindestparameteranforderungen

Um von Microsoft veröffentlichte Anwendungen in Ihre benutzerdefinierte Patch-Baseline aufzunehmen, müssen Sie mindestens das Produkt angeben, das Sie patchen möchten. Der folgende AWS Command Line Interface (AWS CLI)-Befehl zeigt die Mindestanforderungen für das Patchen eines Produkts wie Microsoft 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}]"
```

Wenn Sie die Produktfamilie der Microsoft-Anwendung angeben, müssen alle Produkte der ausgewählten Produktfamilie unterstützt werden. Um beispielsweise das Produkt „Active Directory Rights Management Services Client 2.0“ zu patchen, müssen Sie dessen Produktfamilie als „Active Directory“ und nicht beispielsweise als „Office“ oder „SQL Server“ angeben. Der folgende AWS CLI Befehl zeigt eine Übereinstimmung von Produktfamilie und Produkt.

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**
>
> Wenn Sie eine Fehlermeldung über eine nicht übereinstimmende Produkt- und Familienkopplung erhalten, finden Sie unter Problem: Nicht übereinstimmende Produktfamilien/Produktpaare Tipps zur Lösung des Problems.

Verwenden von Kernel Live Patching auf von Amazon Linux 2 verwalteten Knoten

Kernel Live Patching für Amazon Linux 2 ermöglicht es Ihnen, Patches für Sicherheitsschwachstellen und kritische Fehler auf einen laufenden Linux-Kernel anzuwenden, ohne Neustarts oder Unterbrechungen der laufenden Anwendungen. Sie profitieren damit von einer verbesserten Service- und Anwendungsverfügbarkeit, gleichzeitig bleibt Ihre Infrastruktur sicher und auf dem neuesten Stand. Kernel Live Patching wird auf Amazon-EC2-Instances und AWS IoT Greengrass -Core-Geräten unterstützt sowie auf virtuellen On-Premises-Maschinen, die auf Amazon Linux 2 ausgeführt werden.

Allgemeine Informationen zu Kernel Live Patching finden Sie unter Kernel Live PatchingAmazon Linux 2 im Amazon EC2 EC2-Benutzerhandbuch.

Nachdem Sie einen verwalteten Amazon Linux 2-Knoten Kernel Live Patching eingeschaltet haben, können Sie eine Funktion von verwenden Patch Manager AWS Systems Manager, um Kernel-Live-Patches auf den verwalteten Knoten anzuwenden. Die Verwendung des Patch Manager ist eine Alternative zur Verwendung vorhandener Yum-Workflows auf dem Knoten, um die Updates anzuwenden.

Bevor Sie beginnen

Um mithilfe des Patch Manager Kernel-Live-Patches auf Ihre von Amazon Linux 2 verwalteten Knoten anzuwenden, stellen Sie sicher, dass Ihre Knoten auf der richtigen Architektur und Kernel-Version basieren. Weitere Informationen finden Sie unter Unterstützte Konfigurationen und Voraussetzungen im Amazon EC2 EC2-Benutzerhandbuch.

Themen

- Informationen zu Kernel Live Patching und Patch Manager
- Funktionsweise
- Aktivieren von Kernel Live Patching mit Run Command
- Anwenden von Kernel-Live-Patches unter Verwendung von Run Command
- Deaktivieren von Kernel Live Patching mit Run Command

Informationen zu Kernel Live Patching und Patch Manager

Aktualisieren der Kernel-Version

Sie müssen einen verwalteten Knoten nicht neu starten, nachdem Sie ein Kernel-Live-Patch-Update angewendet haben. AWS Stellt jedoch Kernel-Live-Patches für eine Amazon Linux 2-Kernelversion für bis zu drei Monate nach ihrer Veröffentlichung bereit. Nach Ablauf der dreimonatigen Frist müssen Sie auf eine spätere Kernel-Version aktualisieren, um weiterhin Kernel-Live-Patches zu erhalten. Wir empfehlen Ihnen, mithilfe eines Wartungsfensters mindestens einmal alle drei Monate einen Neustart Ihres Knoten zu planen, um das Update der Kernel-Version zu veranlassen.

Deinstallieren von Kernel-Live-Patches

Kernel-Live-Patches können nicht mit dem Patch Manager deinstalliert werden. Stattdessen können Sie Kernel Live Patching deaktivieren, wodurch die RPM-Pakete für die angewendeten Kernel-Live-Patches entfernt werden. Weitere Informationen finden Sie unter Deaktivieren von Kernel Live Patching mit Run Command.

Kernel-Compliance

In einigen Fällen kann der Kernel durch die Installation aller CVE-Fixes von Live-Patches für die aktuelle Kernel-Version die Compliance-Ebene erreichen, die auch eine neuere Kernel-Version hätte. Wenn dies geschieht, wird die neuere Version als `Installed` und der verwaltete Knoten als `Compliant` gemeldet. Für die neuere Kernel-Version wird jedoch keine Installationszeit gemeldet.

Ein Kernel-Live-Patch, mehrere CVEs

Wenn sich ein Kernel-Live-Patch auf mehrere CVEs bezieht und diese CVEs verschiedene Klassifizierungs- und Schweregradwerte aufweisen, wird für den Patch nur die höchste Klassifizierung und der höchste Schweregrad der CVEs gemeldet.

Im weiteren Teil dieses Abschnitts wird erläutert, wie Patch Manager zum Anwenden von Kernel-Live-Patches auf verwaltete Knoten verwendet wird, die diese Anforderungen erfüllen.

Funktionsweise

AWS veröffentlicht zwei Arten von Kernel-Live-Patches für Amazon Linux 2: Sicherheitsupdates und Bugfixes. Um diese Patch-Typen anzuwenden, verwenden Sie ein Patch-Baseline-Dokument, das

nur auf die in der folgenden Tabelle aufgeführten Klassifizierungen und Schweregrade ausgerichtet ist.

Klassifizierung	Schweregrad
`Security`	`Critical`, `Important`
`Bugfix`	`All`

Sie können eine benutzerdefinierte Patch-Baseline erstellen, die nur auf diese Patches ausgerichtet ist, oder die vordefinierte Patch-Baseline `AWS-AmazonLinux2DefaultPatchBaseline` verwenden. Mit anderen Worten, Sie können `AWS-AmazonLinux2DefaultPatchBaseline` mit von Amazon Linux 2 verwalteten Knoten verwenden, auf denen Kernel Live Patching aktiviert ist, und es werden Kernel-Live-Updates angewendet.

> **Note**
>
> Die `AWS-AmazonLinux2DefaultPatchBaseline`-Konfiguration hat eine Wartezeit von sieben Tagen nach Veröffentlichung oder letzten Aktualisierung eines Patches, bevor er automatisch installiert wird. Wenn Sie nicht sieben Tage warten möchten, bis Kernel-Live-Patches automatisch genehmigt werden, können Sie eine benutzerdefinierte Patch-Baseline erstellen und verwenden. In der Patch-Baseline können Sie keine Wartezeit für automatische Genehmigung oder einen kürzeren oder längeren Zeitraum angeben. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines.

Wir empfehlen die folgende Strategie zum Patchen Ihrer verwalteten Knoten mit Kernel-Live-Updates:

1. Aktivieren Sie Kernel Live Patching auf Ihren von Amazon Linux 2 verwalteten Knoten.
2. Verwenden SieRun Command, eine Funktion von AWS Systems Manager, um einen `Scan` Vorgang auf Ihren verwalteten Knoten unter Verwendung der vordefinierten `AWS-AmazonLinux2DefaultPatchBaseline` oder einer benutzerdefinierten Patch-Baseline auszuführen, die auch nur auf `Security` Updates abzielt, deren Schweregrad als `Critical` und klassifiziert ist`Important`, und dem `Bugfix` Schweregrad von`All`.
3. Verwenden Sie Compliance, eine Funktion von AWS Systems Manager, um zu überprüfen, ob für einen der verwalteten Knoten, die gescannt wurden, Verstöße wegen Patches gemeldet wurden.

Wenn dies der Fall ist, zeigen Sie die Compliance-Details für den Knoten an, um festzustellen, ob Kernel-Live-Patches im verwalteten Knoten fehlen.

4. Um fehlende Kernel-Live-Patches zu installieren, verwenden Sie Run Command mit derselben Patch-Baseline, die Sie zuvor angegeben haben, führen Sie dieses Mal jedoch eine `Install`-Operation anstelle einer `Scan`-Operation aus.

   Da Kernel-Live-Patches installiert werden, ohne dass ein Neustart erforderlich ist, können Sie die Neustartoption `NoReboot` für diese Operation auswählen.

   > **Note**
   >
   > Sie können den verwalteten Knoten dennoch neu starten, wenn dies für andere auf dem Knoten installierte Patch-Typen erforderlich ist oder wenn Sie auf einen neueren Kernel aktualisieren möchten. Wählen Sie in diesen Fällen stattdessen die Neustartoption `RebootIfNeeded` aus.

5. Kehren Sie zu Compliance zurück, um zu überprüfen, ob die Kernel-Live-Patches installiert wurden.

Aktivieren von Kernel Live Patching mit Run Command

Um Kernel Live Patching zu aktivieren, können Sie entweder `yum`-Befehle auf Ihren verwalteten Knoten ausführen oder Run Command und ein benutzerdefiniertes Systems-Manager-Dokument (SSM-Dokument) verwenden, das Sie erstellen.

Informationen zum Einschalten Kernel Live Patching durch direkte Ausführung von `yum` Befehlen auf dem verwalteten Knoten finden Sie unter Aktivieren Kernel Live Patching im Amazon EC2 EC2-Benutzerhandbuch.

> **Note**
>
> Wenn Sie Kernel-Live-Patching aktivieren und der bereits auf dem verwalteten Knoten ausgeführte Kernel eine frühere Version als `kernel-4.14.165-131.185.amzn2.x86_64` (die unterstützte Mindestversion) ist, installiert der Prozess die neueste verfügbare Kernel-Version und startet den verwalteten Knoten neu. Wenn der Knoten bereits `kernel-4.14.165-131.185.amzn2.x86_64` oder höher ausführt, installiert der Prozess keine neuere Version und startet den Knoten nicht neu.

So aktivieren Sie Kernel Live Patching mit Run Command (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) das benutzerdefinierte SSM-Dokument `AWS-ConfigureKernelLivePatching` aus.
5. Geben Sie im Abschnitt Command parameters (Befehlsparameter) an, ob verwaltete Knoten als Teil dieser Operation neu gestartet werden sollen.
6. Weitere Informationen zur Verwendung der übrigen Steuerelemente auf dieser Seite finden Sie unter Ausführen von Befehlen über die Konsole.
7. Wählen Sie Ausführen aus.

Aktivieren von Kernel Live Patching (AWS CLI)

- Führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus.

  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"
  ```

  Ersetzen Sie *instance-id* durch die ID des von Amazon Linux 2 verwalteten Knoten, auf dem Sie das Feature aktivieren möchten, beispielsweise i-02573cafcfEXAMPLE. Um das Feature auf mehreren verwalteten Knoten zu aktivieren, können Sie eines der folgenden Formate verwenden.

- `--targets "Key=instanceids,Values=instance-id1,instance-id2"`
- `--targets "Key=tag:tag-key,Values=tag-value"`

Informationen über andere Optionen, die Sie in dem Befehl verwenden können, finden Sie im Abschnitt send-command in der AWS CLI -Befehlsreferenz.

Anwenden von Kernel-Live-Patches unter Verwendung von Run Command

Um Kernel-Live-Patches anzuwenden, können Sie entweder `yum`-Befehle auf Ihren verwalteten Knoten ausführen oder Run Command und das SSM-Dokument `AWS-RunPatchBaseline` verwenden.

Informationen zum Anwenden von Kernel-Live-Patches durch direkte Ausführung von `yum` Befehlen auf dem verwalteten Knoten finden Sie unter Anwenden von Kernel-Live-Patches im Amazon EC2 EC2-Benutzerhandbuch.

So wenden Sie Kernel-Live-Patches unter Verwendung von Run Command an (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) das SSM-Dokument `AWS-RunPatchBaseline` aus.
5. Führen Sie im Abschnitt Command parameters (Befehlsparameter) einen der folgenden Schritte aus:

   - Wenn Sie prüfen, ob neue Kernel-Live-Patches verfügbar sind, wählen Sie für Operation die Option `Scan` aus. Wenn Ihre verwalteten Knoten nach dieser Operation nicht neu gestartet werden sollen, wählen Sie für Reboot Option (Neustartoption) `NoReboot` aus. Nach Abschluss der Operation können Sie in Compliance prüfen, ob neue Patches vorhanden sind und wie der Compliance-Status lautet.
   - Wenn Sie die Patch-Compliance bereits überprüft haben und bereit sind, verfügbare Kernel-Live-Patches anzuwenden, wählen Sie für Operation die Option `Install` aus. Wenn Ihre verwalteten Knoten nach dieser Operation nicht neu gestartet werden sollen, wählen Sie für Reboot Option (Neustartoption) `NoReboot` aus.

6. Weitere Informationen zur Verwendung der übrigen Steuerelemente auf dieser Seite finden Sie unter Ausführen von Befehlen über die Konsole.
7. Wählen Sie Ausführen aus.

So wenden Sie Kernel-Live-Patches unter Verwendung von Run Command an (AWS CLI)

1. Führen Sie den folgenden Befehl von Ihrem lokalen Computer aus, um eine `Scan`-Operation auszuführen, bevor Sie Ihre Ergebnisse in Compliance überprüfen.

   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
   \"]}
   ```

   Informationen über andere Optionen, die Sie in dem Befehl verwenden können, finden Sie im Abschnitt send-command in der AWS CLI -Befehlsreferenz.
2. Führen Sie den folgenden Befehl von Ihrem lokalen Computer aus, um eine `Install`-Operation auszuführen, nachdem Sie die Ergebnisse in Compliance überprüft haben.

   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\"]}
```

Ersetzen Sie in den beiden vorangegangenen Befehlen *instance-id* durch die ID des von Amazon Linux 2 verwalteten Knoten, auf dem Sie Kernel-Live-Patches anwenden möchten, beispielsweise i-02573cafcfEXAMPLE. Um das Feature auf mehreren verwalteten Knoten zu aktivieren, können Sie eines der folgenden Formate verwenden.

- `--targets "Key=instanceids,Values=instance-id1,instance-id2"`
- `--targets "Key=tag:tag-key,Values=tag-value"`

Informationen über andere Optionen, die Sie in diesen Befehlen verwenden können, finden Sie im Abschnitt send-command in der AWS CLI -Befehlsreferenz.

Deaktivieren von Kernel Live Patching mit Run Command

Um Kernel Live Patching zu deaktivieren, können Sie entweder `yum`-Befehle auf Ihren verwalteten Knoten ausführen oder Run Command und das benutzerdefinierte SSM-Dokument `AWS-ConfigureKernelLivePatching`.

> **Note**
>
> Wenn Sie das Kernel-Live-Patching nicht mehr verwenden möchten, können Sie es jederzeit deaktivieren. In den meisten Fällen ist das Deaktivieren des Features nicht erforderlich.

Informationen zum Ausschalten Kernel Live Patching durch direkte Ausführung von `yum` Befehlen auf dem verwalteten Knoten finden Sie unter Aktivieren Kernel Live Patching im Amazon EC2 EC2-Benutzerhandbuch.

> **Note**
>
> Wenn Sie Kernel Live Patching deaktivieren, deinstalliert der Prozess das Kernel Live Patching-Plug-In und startet dann den verwalteten Knoten neu.

Deaktivieren von Kernel Live Patching mit Run Command (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) das SSM-Dokument `AWS-ConfigureKernelLivePatching` aus.
5. Geben Sie im Abschnitt Command parameters Werte für erforderliche Parameter an.
6. Weitere Informationen zur Verwendung der übrigen Steuerelemente auf dieser Seite finden Sie unter Ausführen von Befehlen über die Konsole.
7. Wählen Sie Ausführen aus.

Deaktivieren von Kernel Live Patching (AWS CLI)

- Verwenden Sie einen Befehl ähnlich dem folgenden:

  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"
  ```

Ersetzen Sie *instance-id* durch die ID des von Amazon Linux 2 verwalteten Knoten, auf dem Sie das Feature deaktivieren möchten, beispielsweise i-02573cafcfEXAMPLE. Um das Feature auf mehreren verwalteten Knoten zu deaktivieren, können Sie eines der folgenden Formate verwenden.

- `--targets "Key=instanceids,Values=instance-id1,instance-id2"`
- `--targets "Key=tag:tag-key,Values=tag-value"`

Informationen über andere Optionen, die Sie in dem Befehl verwenden können, finden Sie im Abschnitt send-command in der AWS CLI -Befehlsreferenz.

Arbeiten mit Patch Manager (Konsole)

Führen Sie Folgendes aus, bevor Sie Patch Manager, eine Funktion von AWS Systems Manager, verwenden. Diese Aufgaben werden später in diesem Abschnitt ausführlich erläutert.

1. Stellen Sie sicher, dass die von AWS vordefinierte Patch-Baseline für die einzelnen verwendeten Betriebssystemtypen Ihre Anforderungen erfüllt. Ist dies nicht der Fall, sollten Sie eine Patch-Baseline erstellen, in der Standard-Patches für diesen Typ von verwalteten Knoten definiert sind, und diese als Standard-Patch-Baseline festlegen.
2. Organisieren Sie verwaltete Knoten mithilfe von Amazon Elastic Compute Cloud (Amazon EC2)-Tags in Patch-Gruppen (optional, aber empfohlen).
3. Führen Sie eine der folgenden Aktionen aus:
   - (Empfohlen) Konfigurieren Sie eine Patch-Richtlinie in Quick Setup, einer Funktion von Systems Manager, mit der Sie fehlende Patches nach einem Zeitplan für eine gesamte Organisation, eine Teilmenge von Organisationseinheiten oder ein einzelnes AWS-Konto installieren können. Weitere Informationen finden Sie unter Patch Manager Patching-Konfiguration der Organisation.
   - Erstellen Sie ein Wartungsfenster, das das Systems Manager-Dokument (SSM-Dokument) `AWS-RunPatchBaseline` in einem Run Command-Aufgabentyp verwendet. Weitere Informationen finden Sie unter Walkthrough: Erstellen eines Wartungsfensters für das Einspielen von Patches (Konsole).
   - Führen Sie `AWS-RunPatchBaseline` manuell in einer Run Command-Operation aus. Weitere Informationen finden Sie unter Ausführen von Befehlen über die Konsole.

- Patchen Sie Knoten bei Bedarf manuell mit der Funktion Patch now (Jetzt patchen). Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten.

4. Überwachen Sie das Patching, um die Compliance zu überprüfen und Fehler zu untersuchen.

Themen

- Erstellen einer Patch-Richtlinie
- Patch-Dashboard-Zusammenfassungen anzeigen
- Arbeiten mit Patch-Compliance-Berichten
- On-Demand-Patchen von verwalteten Knoten
- Arbeiten mit Patch-Baselines
- Anzeigen verfügbarer Patches
- Arbeiten mit Patch-Gruppen
- Arbeiten mit Patch Manager-Einstellungen

Erstellen einer Patch-Richtlinie

Eine Patch-Richtlinie ist eine Konfiguration, die Sie mit Quick Setup, einer Funktion von AWS Systems Manager, einrichten. Patch-Richtlinien bieten eine umfassendere und zentralisiertere Kontrolle über Ihre Patching-Vorgänge, als dies mit anderen Methoden zum Konfigurieren von Patches möglich ist. Eine Patch-Richtlinie definiert den Zeitplan und die Baseline, die beim automatischen Patchen Ihrer Knoten und Anwendungen verwendet werden sollen.

Weitere Informationen finden Sie unter den folgenden Themen:

- Verwenden von Quick Setup-Patch-Richtlinien
- Patch Manager Patching-Konfiguration der Organisation

Patch-Dashboard-Zusammenfassungen anzeigen

Die Registerkarte Dashboard in Patch Manager bietet Ihnen eine Übersichtsansicht in der Konsole, mit der Sie Ihre Patching-Vorgänge in einer konsolidierten Ansicht überwachen können. Patch Managerist eine Fähigkeit von. AWS Systems Manager Auf der Registerkarte Dashboard können Sie folgenden Tabellen anzeigen:

- Ein Snapshot, wie viele verwaltete Knoten mit Patching-Regeln konform bzw. nicht konform sind.

- Ein Snapshot des Alters der Patch-Compliance-Ergebnisse für Ihre verwalteten Knoten.
- Eine verknüpfte Anzahl davon, wie viele nicht konforme verwaltete Knoten für jeden der häufigsten Gründe für Nicht-Compliance vorhanden sind.
- Eine verknüpfte Liste der letzten Patching-Vorgänge.
- Eine verknüpfte Liste der wiederkehrenden Patching-Vorgänge, die eingerichtet wurden.

So zeigen Sie Patch-Dashboard-Übersichten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Dashboard aus.
4. Scrollen Sie zu dem Abschnitt mit zusammenfassenden Daten, den Sie anzeigen möchten:
   - Amazon-EC2-Instance-Management
   - Zusammenfassung der Compliance
   - Anzahl der Nichteinhaltungen der Compliance
   - Compliance-Berichte
   - Nicht auf Patch-Richtlinien basierende Vorgänge
   - Wiederkehrende Aufgaben, die nicht auf Patch-Richtlinien basieren

Arbeiten mit Patch-Compliance-Berichten

Die Informationen in den folgenden Themen helfen Ihnen beim Erstellen von und Arbeiten mit Patch-Compliance-Berichten in Patch Manager, eine Funktion von AWS Systems Manager.

Die Informationen in den folgenden Themen gelten unabhängig davon, welche Methode oder Art der Konfiguration Sie für Ihre Patching-Vorgänge verwenden:

- Eine in Quick Setup konfigurierte Patch-Richtlinie
- Eine in Quick Setup konfigurierte Host-Management-Option
- Ein Wartungsfenster zum Ausführen eines Patch-`Scans` oder einer `Install`-Aufgabe
- Eine On-Demand Patch now-Operation (Jetzt patchen)

> **Important**
>
> Wenn Sie mehrere Arten von Vorgängen einsetzen, um Ihre Instances auf Patch-Compliance zu überprüfen, beachten Sie, dass jeder Scan die Patch-Compliance-Daten der vorherigen Scans überschreibt. Dies kann zu unerwarteten Ergebnissen in Ihren Patch-Compliance-Daten führen. Weitere Informationen finden Sie unter Vermeiden von unbeabsichtigtem Überschreiben von Patch-Compliance-Daten.
> Um zu überprüfen, welche Patch-Baseline verwendet wurde, um die neuesten Compliance-Informationen zu generieren, navigieren Sie zur Registerkarte Compliance-Berichte in Patch Manager, suchen Sie die Zeile für den verwalteten Knoten, zu dem Sie Informationen wünschen, und wählen Sie dann die Baseline-ID in der Spalte Verwendete Baseline-ID aus.

Themen

- Anzeigen der Patch-Compliance-Ergebnisse
- Generieren von Patch-Compliance-Berichten im .csv-Format
- Behebung nicht konformer verwalteter Knoten mit Patch Manager
- Vermeiden von unbeabsichtigtem Überschreiben von Patch-Compliance-Daten

Anzeigen der Patch-Compliance-Ergebnisse

Gehen Sie wie folgt vor, um Patch-Compliance-Informationen zu Ihren verwalteten Knoten anzuzeigen.

Dieses Verfahren gilt für Patchvorgänge, die das `AWS-RunPatchBaseline`-Dokument verwenden. Weitere Informationen zum Anzeigen von Patch-Compliance-Informationen für Patch-Operationen, die das `AWS-RunPatchBaselineAssociation`-Dokument verwenden, finden Sie unter Identifizieren von nicht konformen verwalteten Knoten.

> **Note**
>
> Die Patchscanvorgänge für das Dokument Quick Setup und die Explorer Verwendung des `AWS-RunPatchBaselineAssociation` Dokuments. Quick Setupund Explorer sind beide Funktionen von AWS Systems Manager.

Identifizieren der Patch-Lösung für ein bestimmtes CVE-Problem (Linux)

Für viele Linux-basierte Betriebssysteme geben die Patch-Compliance-Ergebnisse an, welche CVE (Common Vulnerabilities and Exposure) Bulletin-Probleme durch welche Patches behoben werden. Mithilfe dieser Informationen können Sie feststellen, wie dringend Sie einen fehlenden oder fehlgeschlagenen Patch installieren müssen.

CVE-Details sind für unterstützte Versionen der folgenden Betriebssystemtypen enthalten:

- 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**
>
> Standardmäßig stellen CentOS und CentOS Stream keine CVE-Informationen zu Updates bereit. Sie können diese Unterstützung jedoch zulassen, indem Sie Repositorys von Drittanbietern wie das EPEL-Repository (EPEL), das von Fedora veröffentlicht wurde, verwenden. Weitere Informationen finden Sie unter EPEL im Fedora-Wiki.
> Derzeit werden CVE-ID-Werte nur für Patches mit dem Status `Missing` oder `Failed` gemeldet.

Sie können auch CVE-IDs zu Ihren Listen genehmigter oder abgelehnter Patches in Ihren Patch-Baselines hinzufügen, wenn die Situation und Ihre Patching-Ziele dies erfordern.

Weitere Informationen zum Arbeiten mit genehmigten und abgelehnten Patch-Listen finden Sie in den folgenden Themen:

- Arbeiten mit benutzerdefinierten Patch-Baselines
- Paketnamen-Formate für Listen genehmigter und abgelehnter Patches

- Funktionsweise von Patch-Baseline-Regeln auf Linux-basierten Systemen
- Wie Patches installiert werden

> **Note**
>
> In einigen Fällen veröffentlicht Microsoft Patches für Anwendungen, die kein Datum und keine Uhrzeit der Aktualisierung angeben. In diesen Fällen wird ein aktualisiertes Datum und eine Uhrzeit von `01/01/1970` standardmäßig angegeben.

Anzeigen der Patch-Compliance-Ergebnisse

Verwenden Sie die folgenden Verfahren, um Patch-Compliance-Ergebnisse in der AWS Systems Manager -Konsole anzuzeigen.

> **Note**
>
> Weitere Informationen zum Erstellen von Patch-Compliance-Berichten, die in einen Amazon Simple Storage Service (Amazon S3)-Bucket heruntergeladen werden, finden Sie unter Generieren von Patch-Compliance-Berichten im .csv-Format.

Anzeigen der Patch-Compliance-Ergebnisse

1. Führen Sie eine der folgenden Aufgaben aus.

   Option 1 (empfohlen) — Navigieren vonPatch Manager, eine Funktion von AWS Systems Manager:

   - Wählen Sie im Navigationsbereich Patch Manager aus.
   - Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte).
   - Wählen Sie im Bereich Knoten-Patching-Details die Knoten-ID des verwalteten Knotens aus, für den Sie die Ergebnisse der Patch-Konformität überprüfen möchten.
   - Wählen Sie im Bereich Details in der Eigenschaftenliste die Option Patches aus.

   Option 2 — Gehen Sie von Compliance aus, eine der folgenden Funktionen AWS Systems Manager:

- Wählen Sie im linken Navigationsbereich Compliance.
- Für Zusammenfassung von Compliance-Ressourcen wählen Sie in der Spalte für die Typen von Patch-Ressourcen, die Sie überprüfen möchten, z. B. Nicht regelkonforme Ressourcen, eine Zahl aus.
- Wählen Sie unten in der Ressourcenliste die ID des verwalteten Knotens aus, für den Sie die Ergebnisse der Patch-Konformität überprüfen möchten.
- Wählen Sie im Bereich Details in der Eigenschaftenliste die Option Patches aus.

Option 3 — Navigieren vonFleet Manager, eine Funktion von AWS Systems Manager.

- Wählen Sie im Navigationsbereich Fleet Manager aus.
- Wählen Sie im Bereich Verwaltete Instanzen die ID des verwalteten Knotens aus, für den Sie die Ergebnisse der Patch-Konformität überprüfen möchten.
- Wählen Sie im Bereich Details in der Eigenschaftenliste die Option Patches aus.

2. (Optional) Wählen Sie im Suchfeld
()
einen der verfügbaren Filter aus.

   Wählen Sie zum Beispiel für Red Hat Enterprise Linux (RHEL) aus den folgenden Optionen aus:

   - Name
   - Klassifizierung
   - Status
   - Schweregrad

   Wählen Sie für Windows Server eine der folgenden Optionen aus:

   - KB
   - Klassifizierung
   - Status
   - Schweregrad

3. Wählen Sie einen der verfügbaren Werte für den ausgewählten Filtertyp aus. Wenn Sie beispielsweise „Status“ ausgewählt haben, wählen Sie jetzt einen Konformitätsstatus wie InstalledPendingReboot„Fehlgeschlagen“ oder „Fehlend“.

> **Note**
>
> Derzeit werden CVE-ID-Werte nur für Patches mit dem Status `Missing` oder `Failed` gemeldet.

4. Abhängig vom Compliance-Zustand des verwalteten Knoten können Sie auswählen, welche Maßnahmen zum Beheben von nicht konformen Knoten ergriffen werden sollen.

   Sie können beispielsweise wählen, dass Ihre nicht konformen verwalteten Knoten sofort gepatcht werden sollen. Informationen zum On-Demand-Patching Ihrer verwalteten Knoten finden Sie unter On-Demand-Patchen von verwalteten Knoten.

   Weitere Informationen zu Patch-Compliance-Status finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten.

Generieren von Patch-Compliance-Berichten im .csv-Format

Sie können die AWS Systems Manager Konsole verwenden, um Patch-Compliance-Berichte zu erstellen, die als CSV-Datei in einem Amazon Simple Storage Service (Amazon S3) -Bucket Ihrer Wahl gespeichert werden. Sie können einen einzelnen On-Demand-Bericht erstellen oder einen Zeitplan für die automatische Generierung der Berichte angeben.

Berichte können für einen einzelnen verwalteten Knoten oder für alle verwalteten Knoten in Ihrem ausgewählten AWS-Konto und AWS-Region generiert werden. Für einen einzelnen Knoten enthält ein Bericht umfassende Details, einschließlich der IDs von Patches, die sich auf einen nicht konformen Knoten beziehen. Für einen Bericht über alle verwaltete Knoten werden nur zusammenfassende Informationen und die Anzahl der Patches von nicht konformen Knoten bereitgestellt.

Nachdem ein Bericht generiert wurde, können Sie ein Tool wie Amazon verwenden, QuickSight um die Daten zu importieren und zu analysieren. Amazon QuickSight ist ein Business Intelligence (BI) -Service, mit dem Sie Informationen in einer interaktiven visuellen Umgebung untersuchen und interpretieren können. Weitere Informationen finden Sie im QuickSight Amazon-Benutzerhandbuch.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Patch-Baseline erstellen, können Sie für Patches, die von dieser Patch-Baseline genehmigt wurden, einen Schweregrad für die Konformität angeben,

beispielsweise `Critical` oder `High`. Wenn der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann ist der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline der von Ihnen angegebene Schweregrad.

Sie können auch ein Thema zum Amazon Simple Notification Service (Amazon SNS) angeben, um Benachrichtigungen zu senden, wenn ein Bericht erstellt wird.

Servicerollen zum Generieren von Patch-Compliance-Berichten

Wenn Sie zum ersten Mal einen Bericht erstellen, erstellt Systems Manager eine angenommene Automatisierungsrolle mit dem Namen `AWS-SystemsManager-PatchSummaryExportRole`, die für den Exportprozess zu S3 verwendet wird.

Note

Wenn Sie Compliance-Daten in einen verschlüsselten S3-Bucket exportieren, müssen Sie die zugehörige AWS KMS Schlüsselrichtlinie aktualisieren, um die erforderlichen Berechtigungen für bereitzustellen`AWS-SystemsManager-PatchSummaryExportRole`. Fügen Sie beispielsweise der AWS KMS Richtlinie Ihres S3-Buckets eine ähnliche Berechtigung hinzu:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey"
    ],
    "Resource": "role-arn"
}
```

Ersetzen Sie *`role-arn`* durch den Amazon-Ressourcenname (ARN) der in Ihrem Konto erstellten Datei im Format `arn:aws:iam::`*`111222333444`*`:role/service-role/AWS-SystemsManager-PatchSummaryExportRole`.
Weitere Informationen finden Sie unter Schlüsselrichtlinien in AWS KMS im Entwicklerhandbuch für AWS Key Management Service .

Wenn Sie zum ersten Mal einen Bericht nach einem Zeitplan generieren, erstellt Systems Manager eine weitere Servicerolle mit dem Namen `AWS-EventBridge-Start-SSMAutomationRole` zusammen mit der Servicerolle `AWS-SystemsManager-PatchSummaryExportRole` (falls nicht

bereits erstellt), die für den Exportvorgang verwendet werden soll. `AWS-EventBridge-Start-SSMAutomationRole`ermöglicht Amazon EventBridge , eine Automatisierung mit dem Runbook AWS ExportPatchReportToS3 zu starten.

Es wird empfohlen, diese Richtlinien und Rollen zu ändern. Dies kann dazu führen, dass die Erstellung von Patch-Compliance-Berichten fehlschlägt. Weitere Informationen finden Sie unter Problembehandlung bei der Erstellung von Patch-Compliance-Berichten.

Themen

- Was ist in einem generierten Patch-Compliance-Bericht enthalten?
- Generieren von Patch-Compliance-Berichten für einen einzelnen verwalteten Knoten
- Generieren von Patch-Compliance-Berichten für alle verwaltete Knoten
- Berichtsverlauf für Patch-Compliance anzeigen
- Zeitpläne für Patch-Compliance-Berichte anzeigen
- Problembehandlung bei der Erstellung von Patch-Compliance-Berichten

Was ist in einem generierten Patch-Compliance-Bericht enthalten?

Dieses Thema enthält Informationen zu den Inhaltstypen, die in den Patch-Compliance-Berichten enthalten sind, die generiert und in einen angegebenen S3-Bucket heruntergeladen werden.

Berichtsformat für einen einzelnen verwalteten Knoten

Ein für einen einzelnen verwalteten Knoten generierter Bericht liefert sowohl zusammenfassende als auch detaillierte Informationen.

Herunterladen eines Beispielberichts (einzelner Knoten)

Zu den zusammenfassenden Informationen für einen einzelnen verwalteten Knoten gehört Folgendes:

- Index
- Instance-ID
- Instance name
- Instance IP
- Plattformname
- Plattformversion

- SSM Agent-Version
- Patch-Baseline
- Patch-Gruppe
- Compliance status (Compliance-Status)
- Compliance-Schweregrad
- Anzahl nicht konformer Patches mit kritischem Schweregrad
- Anzahl nicht konformer Patches mit hohem Schweregrad
- Anzahl nicht konformer Patches mit der Schweregrad Mittel
- Anzahl nicht konformer Patches mit niedrigem Schweregrad
- Anzahl nicht konformer Patches mit informativen Schweregrad
- Anzahl nicht konformer Patches mit nicht spezifiziertem Schweregrad

Zu den detaillierten Informationen für einen verwalteten einzelnen Knoten gehört Folgendes:

- Index
- Instance-ID
- Instance name
- Patch-Name
- KB-ID/Patch-ID
- Patch-Status
- Zeitpunkt des letzten Berichts
- Compliance-Stufe
- Patch-Schweregrad
- Patch-Klassifizierung
- CVE-ID
- Patch-Baseline
- Logs-URL
- Instance IP
- Plattformname
- Plattformversion

- SSM Agent-Version

> **Note**
>
> Wenn Sie eine benutzerdefinierte Patch-Baseline erstellen, können Sie für Patches, die von dieser Patch-Baseline genehmigt wurden, einen Schweregrad für die Konformität angeben, beispielsweise `Critical` oder `High`. Wenn der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann ist der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline der von Ihnen angegebene Schweregrad.

Berichtsformat für alle verwaltete Knoten

Ein für alle verwaltete Knoten generierter Bericht enthält nur zusammenfassende Informationen.

Herunterladen eines Beispielberichts (alle verwaltete Knoten)

Zu den zusammenfassenden Informationen für alle verwaltete Knoten gehört Folgendes:

- Index
- Instance-ID
- Instance name
- Instance IP
- Plattformname
- Plattformversion
- SSM Agent-Version
- Patch-Baseline
- Patch-Gruppe
- Compliance status (Compliance-Status)
- Compliance-Schweregrad
- Anzahl nicht konformer Patches mit kritischem Schweregrad
- Anzahl nicht konformer Patches mit hohem Schweregrad
- Anzahl nicht konformer Patches mit der Schweregrad Mittel
- Anzahl nicht konformer Patches mit niedrigem Schweregrad
- Anzahl nicht konformer Patches mit informativen Schweregrad

- Anzahl nicht konformer Patches mit nicht spezifiziertem Schweregrad

Generieren von Patch-Compliance-Berichten für einen einzelnen verwalteten Knoten

Gehen Sie wie folgt vor, um einen Patch-Zusammenfassungs-Bericht für einen einzelnen verwalteten Knoten in Ihrem AWS-Konto zu generieren. Der Bericht für einen einzelnen verwalteten Knoten enthält Details zu jedem Patch, der nicht konform ist, einschließlich Patch-Namen und IDs.

So generieren Sie Patch-Compliance-Berichte für einen einzelnen verwalteten Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte) aus.
4. Wählen Sie die Schaltfläche für die Zeile des verwalteten Knoten aus, für den Sie einen Bericht erstellen möchten, und wählen Sie dann View detail (Detail anzeigen) aus.
5. Wählen Sie Abschnitt mit der Patch-Zusammenfassung Exportieren nach S3 aus.
6. Für Berichtname geben Sie einen Namen ein, damit Sie den Bericht später leichter identifizieren können.
7. Für Meldehäufigkeit wählen Sie eine der folgenden Optionen aus:
   - On Demand – Erstellen Sie einen einmaligen Bericht. Fahren Sie mit Schritt 9 fort.
   - Nach einem Plan – Geben Sie einen wiederkehrenden Zeitplan für die automatische Erstellung von Berichten an. Fahren Sie fort mit Schritt 8.
8. Für den Typ „Nach einem Plan“ geben Sie entweder einen Kursausdruck an, z. B. alle 3 Tage, oder geben Sie einen Cron-Ausdruck an, um die Berichtshäufigkeit festzulegen.

   Informationen zu Cron-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.
9. Für Bucket-Name wählen Sie den Namen eines S3-Buckets aus, in dem die CSV-Berichtsdateien gespeichert werden sollen.

   > **Important**
   >
   > Wenn Sie in einem System arbeiten AWS-Region , das nach dem 20. März 2019 gestartet wurde, müssen Sie einen S3-Bucket in derselben Region auswählen. Nach

diesem Datum gestartete Regionen wurden standardmäßig deaktiviert. Weitere Informationen und eine Liste dieser Regionen finden Sie unter Aktivieren einer Region in der Allgemeine Amazon Web Services-Referenz.

10. (Optional) Um Benachrichtigungen zu senden, wenn der Bericht erstellt wird, erweitern Sie den Abschnitt SNS-Thema und wählen Sie dann aus SNS-Thema Amazon-Ressourcenname (ARN) ein vorhandenes Amazon-SNS-Thema aus.
11. Wählen Sie Absenden aus.

Informationen zum Anzeigen eines Verlaufs von generierten Berichten finden Sie unter Berichtsverlauf für Patch-Compliance anzeigen.

Informationen zum Anzeigen von Details zu von Ihnen erstellten Berichtszeitplänen finden Sie unter Zeitpläne für Patch-Compliance-Berichte anzeigen.

Generieren von Patch-Compliance-Berichten für alle verwaltete Knoten

Gehen Sie wie folgt vor, um einen Patch-Zusammenfassungs-Bericht für alle verwaltete Knoten in Ihrem AWS-Konto zu generieren. Der Bericht für alle verwalteten Knoten zeigt an, welche Knoten nicht konform sind und wie viele Patches nicht konform sind. Es gibt keine Namen oder andere Bezeichner der Patches. Für diese zusätzlichen Details können Sie einen Patch-Compliance-Bericht für einen einzelnen verwalteten Knoten erstellen. Informationen finden Sie unter Generieren von Patch-Compliance-Berichten für einen einzelnen verwalteten Knoten weiter vorne in diesem Thema.

Generieren von Patch-Compliance-Berichten für alle verwaltete Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte) aus.
4. Klicken Sie auf Export to S3 (Exportieren nach S3). (Wählen Sie nicht zuerst eine Knoten-ID aus.)
5. Für Berichtname geben Sie einen Namen ein, damit Sie den Bericht später leichter identifizieren können.
6. Für Meldehäufigkeit wählen Sie eine der folgenden Optionen aus:

   - On Demand – Erstellen Sie einen einmaligen Bericht. Fahren Sie mit Schritt 8 fort.

- Nach einem Plan – Geben Sie einen wiederkehrenden Zeitplan für die automatische Erstellung von Berichten an. Fahren Sie fort mit Schritt 7.

7. Für den Typ „Nach einem Plan“ geben Sie entweder einen Kursausdruck an, z. B. alle 3 Tage, oder geben Sie einen Cron-Ausdruck an, um die Berichtshäufigkeit festzulegen.

   Informationen zu Cron-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

8. Für Bucket-Name wählen Sie den Namen eines S3-Buckets aus, in dem die CSV-Berichtsdateien gespeichert werden sollen.

   > **Important**
   >
   > Wenn Sie in einem System arbeiten AWS-Region , das nach dem 20. März 2019 gestartet wurde, müssen Sie einen S3-Bucket in derselben Region auswählen. Nach diesem Datum gestartete Regionen wurden standardmäßig deaktiviert. Weitere Informationen und eine Liste dieser Regionen finden Sie unter Aktivieren einer Region in der Allgemeine Amazon Web Services-Referenz.

9. (Optional) Um Benachrichtigungen zu senden, wenn der Bericht erstellt wird, erweitern Sie den Abschnitt SNS-Thema und wählen Sie dann aus SNS-Thema Amazon-Ressourcenname (ARN) ein vorhandenes Amazon-SNS-Thema aus.
10. Wählen Sie Absenden aus.

Informationen zum Anzeigen eines Verlaufs von generierten Berichten finden Sie unter Berichtsverlauf für Patch-Compliance anzeigen.

Informationen zum Anzeigen von Details zu von Ihnen erstellten Berichtszeitplänen finden Sie unter Zeitpläne für Patch-Compliance-Berichte anzeigen.

Berichtsverlauf für Patch-Compliance anzeigen

Mithilfe der Informationen in diesem Thema können Sie sich Details zu den Patch-Compliance-Berichten anzeigen lassen, die in Ihrem erstellt wurden AWS-Konto.

Anzeigen des Berichtsverlaufs für Patch-Compliance

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte) aus.
4. Klicken Sie auf Alle S3-Exporte anzeigen und danach auf die Registerkarte Exportieren des Verlaufs.

Zeitpläne für Patch-Compliance-Berichte anzeigen

Mithilfe der Informationen in diesem Thema können Sie sich Details zu den in Ihrem erstellten Zeitplan für die Erstellung von Berichten zur Patch-Konformität anzeigen lassen AWS-Konto.

Anzeigen des Berichtsverlaufs für Patch-Compliance

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte) aus.
4. Wählen Sie View all S3 exports (Alle S3-Exporte anzeigen) und danach die Registerkarte Report schedule rules (Regeln für die Berichtsplanung) aus.

Problembehandlung bei der Erstellung von Patch-Compliance-Berichten

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit Patch-Compliance-Berichten in Patch Manager, einer Funktion von AWS Systems Manager.

Themen

- Eine Nachricht meldet, dass die AWS-SystemsManager-PatchManagerExportRolePolicy-Richtlinie beschädigt ist
- Nach dem Löschen von Patch-Compliance-Richtlinien oder -Rollen werden geplante Berichte nicht erfolgreich generiert

Eine Nachricht meldet, dass die **`AWS-SystemsManager-PatchManagerExportRolePolicy`**-Richtlinie beschädigt ist

Problem: Sie erhalten eine Fehlermeldung ähnlich der folgenden, die angibt, dass `AWS-SystemsManager-PatchManagerExportRolePolicy` beschädigt ist:

```
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.
```

- Lösung: Verwenden Sie die Patch Manager Konsole oder löschen AWS CLI Sie die betroffenen Rollen und Richtlinien, bevor Sie einen neuen Patch-Compliance-Bericht erstellen.

  So löschen Sie die beschädigte Richtlinie über die Konsole

  1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
  2. Führen Sie eine der folgenden Aktionen aus:

     On-Demand-Berichte – Wenn das Problem beim Generieren eines einmaligen On-Demand-Berichts aufgetreten ist, wählen Sie in der linken Navigation Richtlinien aus und suchen Sie nach `AWS-SystemsManager-PatchManagerExportRolePolicy`. Löschen Sie dann die Richtlinie. Wählen Sie anschließend Rollen aus, suchen Sie nach `AWS-SystemsManager-PatchSummaryExportRole` und löschen Sie sie.

     Geplante Berichte – Wenn der Fehler während der Erstellung eines geplanten Berichts aufgetreten ist, wählen Sie in der linken Navigation Richtlinien aus, suchen Sie nacheinander nach `AWS-EventBridge-Start-SSMAutomationRolePolicy` und `AWS-SystemsManager-PatchManagerExportRolePolicy` und löschen Sie jede Richtlinie. Wählen Sie anschließend Rollen aus, schen Sie nacheinander nach `AWS-EventBridge-Start-SSMAutomationRole` und `AWS-SystemsManager-PatchSummaryExportRole` und löschen Sie jede Rolle.

  Um die beschädigte Richtlinie mit dem zu löschen AWS CLI

  Ersetzen Sie die *`Platzhalterwerte`* durch Ihre Konto-ID.

  - Wenn das Problem bei der Erstellung eines einmaligen On-Demand-Berichts aufgetreten ist, führen Sie die folgenden Befehle aus:

    ```
    aws iam delete-policy --policy-arn arn:aws:iam::account-id:policy/AWS-
    SystemsManager-PatchManagerExportRolePolicy
    ```

    ```
    aws iam delete-role --role-name AWS-SystemsManager-PatchSummaryExportRole
    ```

Wenn das Problem bei der Erstellung eines Zeitplanberichts auftritt, führen Sie die folgenden Befehle aus:

```
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
```

Nachdem Sie eines der beiden Verfahren abgeschlossen haben, folgen Sie den Schritten, um einen neuen Patch-Compliance-Bericht zu erstellen oder zu planen.

Nach dem Löschen von Patch-Compliance-Richtlinien oder -Rollen werden geplante Berichte nicht erfolgreich generiert

Problem: Wenn Sie zum ersten Mal einen Bericht erstellen, erstellt Systems Manager eine Servicerolle und eine Richtlinie für den Exportprozess (`AWS-SystemsManager-PatchSummaryExportRole` und `AWS-SystemsManager-PatchManagerExportRolePolicy`). Wenn Sie zum ersten Mal einen geplanten Bericht erstellen, erstellt Systems Manager eine weitere Servicerolle und eine Richtlinie (`AWS-EventBridge-Start-SSMAutomationRole` und `AWS-EventBridge-Start-SSMAutomationRolePolicy`). Diese ermöglichen es Amazon, eine Automatisierung mithilfe des Runbooks AWS ExportPatchReportToS3 zu EventBridge starten.

Wenn Sie eine dieser Richtlinien oder Rollen löschen, gehen die Verbindungen zwischen Ihrem Zeitplan und Ihrem angegebenen S3-Bucket und Amazon SNS-Thema möglicherweise verloren.

- Lösung: Um dieses Problem zu umgehen, empfehlen wir, den vorherigen Zeitplan zu löschen und einen neuen Zeitplan zu erstellen, um den zu ersetzen, bei dem Probleme aufgetreten sind.

Behebung nicht konformer verwalteter Knoten mit Patch Manager

Die Themen in diesem Abschnitt enthalten eine Übersicht darüber, wie verwaltete Knoten, die die Patch-Compliance nicht erfüllen, identifiziert werden können und wie sie konform gemacht werden.

Themen

- Identifizieren von nicht konformen verwalteten Knoten
- Grundlegendes zu Patch-Compliance-Statuswerten
- Patchen nicht konformer verwalteter Knoten

Identifizieren von nicht konformen verwalteten Knoten

Out-of-compliance -verwaltete Knoten werden identifiziert, wenn eines von zwei AWS Systems Manager Dokumenten (SSM-Dokumente) ausgeführt wird. Diese SSM-Dokumente verweisen auf die entsprechende Patch-Baseline für jeden verwalteten Knoten in Patch Manager, einer Funktion von AWS Systems Manager. Anschließend werten sie den Patch-Zustand des verwalteten Knoten aus und stellen Ihnen dann Compliance-Ergebnisse zur Verfügung.

Es gibt zwei SSM-Dokumente, die verwendet werden, um nicht konforme verwaltete Knoten zu identifizieren oder zu aktualisieren: `AWS-RunPatchBaseline` und `AWS-RunPatchBaselineAssociation`. Jedes wird von verschiedenen Prozessen verwendet, und ihre Compliance-Ergebnisse sind über verschiedene Kanäle verfügbar. In der folgenden Tabelle werden die Unterschiede zwischen diesen Dokumenten aufgeführt.

> **Note**
>
> Patch-Compliance-Daten von Patch Manager können an AWS Security Hub gesendet werden. Mit dem Security Hub erhalten Sie einen umfassenden Überblick über Ihre Sicherheitswarnungen und den Compliance-Status mit hoher Priorität. Er überwacht auch den Patching-Status Ihrer Flotte. Weitere Informationen finden Sie unter Integrieren Patch Manager mit AWS Security Hub.

	AWS-RunPatchBaseline	**AWS-RunPatchBaselineAssociation**
Prozesse, die das Dokument verwenden	On-Demand-Patchen – Sie können verwaltete Knoten	Quick Setup-Host-Verwaltung für Systems Manager

	AWS-RunPatchBaseline	AWS-RunPatchBaselineAssociation
	bei Bedarf scannen oder patchen, indem Sie die Option Patch now (Jetzt patchen) verwenden. Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten. Quick Setup-Patch-Richtlinien von Systems Manager – Sie können eine Patching-Konfiguration in Quick Setup, einer Funktion von AWS Systems Manager, erstellen, die fehlende Patches in separaten Zeitplänen für eine gesamte Organisation, eine Teilmenge von Organisationseinheiten oder ein einzelnes AWS-Konto scannen oder installieren kann. Weitere Informationen finden Sie unter Patch Manager Patching-Konfiguration der Organisation. Einen Befehl ausführen – Sie können `AWS-RunPatchBaseline` manuell in einer Operation in Run Command, einer Funktion von AWS Systems Manager, ausführen. Weitere Informationen finden Sie unter	– Sie können eine Host-Management-Konfigurationsoption in Quick Setup aktivieren, um Ihre verwalteten Instances täglich auf Patch-Compliance zu scannen. Weitere Informationen finden Sie unter Amazon-EC2-Host-Verwaltung. Systems Manager Explorer – Wenn Sie Explorer, eine Funktion von AWS Systems Manager, aktiviert, überprüft es Ihre verwalteten Instances regelmäßig auf Patch-Compliance und meldet Ergebnisse im Explorer-Dashboard.

	AWS-RunPatchBaseline	AWS-RunPatchBaselineAssociation
	Ausführen von Befehlen über die Konsole. Wartungsfenster – Sie können ein Wartungsfenster erstellen, das das SSM-Dokument `AWS-RunPatchBaseline` in einem Run Command-Aufgabentyp verwendet. Weitere Informationen finden Sie unter Walkthrough: Erstellen eines Wartungsfensters für das Einspielen von Patches (Konsole).	
Format der Patch-Scan-Ergebnisdaten	Nach der Ausführung von `AWS-RunPatchBaseline` sendet Patch Manager ein `AWS:PatchSummary` -Objekt an Inventory, eine Funktion von AWS Systems Manager.	Nach der Ausführung von `AWS-RunPatchBaselineAssociation` sendet Patch Manager ein `AWS:ComplianceItem` -Objekt an Systems Manager Inventory.

	AWS-RunPatchBaseline	AWS-RunPatchBaselineAssociation
So zeigen Sie aktuelle Compliance-Berichte in der Konsole an	Sie können Patch-Compliance-Informationen für Prozesse anzeigen, die `AWS-RunPatchBaseline` in Systems Manager Compliance und Mit verwalteten Knoten arbeiten verwenden. Weitere Informationen finden Sie unter Anzeigen der Patch-Compliance-Ergebnisse.	Wenn Sie Quick Setup verwenden, um Ihre verwalteten Instances auf Patch-Compliance zu überprüfen, finden Sie den Compliance-Bericht unter Systems Manager State Manager, auf das Sie über die Schaltfläche Ergebnisse anzeigen in Quick Setup zugreifen. Wenn Sie Explorer verwenden, um Ihre verwalteten Instances auf Patch-Compliance zu überprüfen, finden Sie den Compliance-Bericht sowohl unter Explorer als auch unter Systems Manager OpsCenter.
AWS CLI-Befehle zum Anzeigen von Patch-Compliance-Ergebnissen	Für Prozesse, die `AWS-RunPatchBaseline` verwenden, können Sie die folgenden AWS CLI-Befehle verwenden, um zusammenfassende Informationen zu Patches auf einem verwalteten Knoten anzuzeigen. • describe-instance-patch-states • describe-instance-patch-states-for-patch-group • describe-patch-group-state	Für Prozesse, die `AWS-RunPatchBaselineAssociation` verwenden, können Sie den folgenden AWS CLI-Befehl verwenden, um zusammenfassende Informationen zu Patches auf einer Instance anzuzeigen. • list-compliance-items

	AWS-RunPatchBaseline	AWS-RunPatchBaselineAssociation
Patch-Operationen	Für Prozesse, die `AWS-RunPatchBaseline` verwenden, geben Sie an, ob der Vorgang nur eine `Scan`-Operation oder eine `Scan and install`-Operation ausführen soll. Wenn Ihr Ziel darin besteht, nicht konforme verwaltete Knoten zu identifizieren und nicht zu beheben, führen Sie nur eine `Scan`-Operation durch.	Quick Setup- und Explorer-Prozesse, die `AWS-RunPatchBaselineAssociation` verwenden, führen nur eine `Scan`-Operation aus.
Weitere Informationen	Informationen über das `AWS-RunPatchBaseline` SSM-Dokument	Informationen über das `AWS-RunPatchBaselineAssociation` SSM-Dokument

Informationen zu den verschiedenen Patch-Compliance-Status, die möglicherweise gemeldet werden, finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten

Informationen zur Behebung verwalteter Knoten, die die Patch-Compliance nicht erfüllen, finden Sie unter Patchen nicht konformer verwalteter Knoten.

Grundlegendes zu Patch-Compliance-Statuswerten

Zu den Informationen über Patches für einen verwalteten Knoten gehört ein Bericht über den Zustand oder den Status jedes einzelnen Patches.

> **Note**
>
> Wenn Sie einem verwalteten Knoten einen bestimmten Patch-Compliance-Status zuweisen möchten, können Sie den Befehl put-compliance-items AWS Command Line Interface

(AWS CLI) oder die PutComplianceItemsAPI-Operation verwenden. Das Zuweisen eines Compliance-Zustands wird in der Konsole nicht unterstützt.

Verwenden Sie die Informationen in den folgenden Tabellen, um zu ermitteln, warum ein verwalteter Knoten möglicherweise nicht die Patch-Compliance erfüllt.

Patch-Compliance-Werte für Debian Server, Raspberry Pi OS und Ubuntu Server

Für Debian Server, Raspberry Pi OS und Ubuntu Server erläutert die folgende Tabelle die Regeln für die Paketklassifizierung in verschiedene Compliance-Zustände.

Note

Beachten Sie bei der Auswertung der Statuswerte Installiert, Installiert Andere und Fehlend Folgendes: Wenn Sie beim Erstellen oder Aktualisieren einer Patch-Baseline nicht die Checkbox Nicht sicherheitsrelevante Aktualisierungen einschließen aktivieren, sind Patch-Kandidaten-Versionen auf Patches beschränkt, die in `trusty-security` (Ubuntu Server 14.04 LTS), `xenial-security` (Ubuntu Server 16.04 LTS), `bionic-security` (Ubuntu Server 18.04 LTS), `focal-security` (Ubuntu Server 20.04 LTS), `groovy-security` (Ubuntu Server 20.10 STR), `jammy-security` (Ubuntu Server 22.04 LTS) oder `debian-security` (Debian Server and Raspberry Pi OS). enthalten sind. Wenn Sie die Option Nicht sicherheitsrelevante Aktualisierungen einschließen auswählen, werden auch Patches aus anderen Repositorys berücksichtigt.

Patch-Status	Beschreibung	Compliance status (Compliance-Status)
`INSTALLED`	Der Patch wird in der Patch-Baseline aufgeführt und ist auf dem verwalteten Knoten installiert. Er könnte entweder manuell von einer Person oder automatisch von Patch Manager installiert worden sein, wenn das `AWS-RunPa`	Konform

Patch-Status	Beschreibung	Compliance status (Compliance-Status)
	`tchBaseline` -Dokument auf dem verwalteten Knoten ausgeführt wurde.	
`INSTALLED_OTHER`	Der Patch ist nicht in der Baseline enthalten oder wird von der Baseline nicht genehmigt, ist aber auf dem verwalteten Knoten installiert. Der Patch wurde möglicherweise manuell installiert, das Paket könnte eine erforderliche Abhängigkeit von einem anderen genehmigten Patch sein, oder der Patch war möglicherweise Teil eines InstallOverrideList Vorgangs. Wenn Sie `Block` nicht als die Zurückgewiesene Patches-Aktion angeben, schließt `Installed_Other` auch installiert, aber abgelehnte Patches ein.	Konform

Patch-Status	Beschreibung	Compliance status (Compliance-Status)
INSTALLED_PENDING_REBOOT	INSTALLED_PENDING_REBOOT kann eines von zwei Dingen bedeuten: • Der Install-Vorgang von Patch Manager hat den Patch zwar auf den verwalteten Knoten angewendet, aber der Knoten wurde seit dem Anwenden des Patches nicht mehr neu gestartet. Dies bedeutet in der Regel, dass für den Parameter RebootOption die Option NoReboot ausgewählt wurde, als das AWS-RunPatchBaseline -Dokument zuletzt auf dem verwalteten Knoten ausgeführt wurde. Weitere Informationen finden Sie unter Parameter name: RebootOption . • Patch ManagerSeit dem letzten Neustart des verwalteten Knotens wurde ein Patch außerhalb von installiert.	Nicht konform

Patch-Status	Beschreibung	Compliance status (Compliance-Status)
INSTALLED_REJECTED	Der Patch wird auf dem verwalteten Knoten installiert, jedoch in einer Liste der Rejected patches (Abgelehnte Patches) angegeben. Dies bedeutet normalerweise, dass der Patch installiert wurde, bevor er einer Liste der abgelehnten Patches hinzugefügt wurde.	Nicht konform
MISSING	Pakete, die über die Baseline gefiltert und noch nicht installiert sind.	Nicht konform
FAILED	Pakete, die während des Patch-Vorgangs nicht installiert werden konnten.	Nicht konform

Patch-Compliance-Werte für andere Betriebssysteme

Für alle Betriebssysteme außer Debian Server, Raspberry Pi OS und Ubuntu Server erläutert die folgende Tabelle die Regeln für die Paketklassifizierung in verschiedene Compliance-Zustände.

Patch-Status	Beschreibung	Compliancewert
INSTALLED	Der Patch wird in der Patch-Baseline aufgeführt und ist auf dem verwalteten Knoten installiert. Er könnte entweder manuell von einer Person oder automatisch von Patch Manager installiert worden sein, als das `AWS-RunPa`	Konform

Patch-Status	Beschreibung	Compliancewert
	`tchBaseline` -Dokument auf dem Knoten ausgeführt wurde.	
`INSTALLED_OTHER` [1]	Der Patch befindet sich nicht an der Baseline, ist aber auf dem verwalteten Knoten installiert. Der Patch wurde möglicherweise manuell installiert, das Paket könnte eine erforderliche Abhängigk eit von einem anderen genehmigten Patch sein. Wenn Sie `Block` nicht als die Zurückgewiesene Patches-Aktion angeben, schließt `Installed_Other` auch installiert, aber abgelehnte Patches ein.	Konform
`INSTALLED_REJECTED`	Der Patch wird auf dem verwalteten Knoten installie rt, jedoch in einer Liste der abgelehnten Patches angegeben. Dies bedeutet normalerweise, dass der Patch installiert wurde, bevor er einer Liste der abgelehnten Patches hinzugefügt wurde.	Nicht konform

Patch-Status	Beschreibung	Compliancewert
INSTALLED_PENDING_REBOOT	INSTALLED_PENDING_REBOOT kann eines von zwei Dingen bedeuten: • Der Install-Vorgang von Patch Manager hat den Patch zwar auf den verwalteten Knoten angewendet, aber der Knoten wurde seit dem Anwenden des Patches nicht mehr neu gestartet. Dies bedeutet in der Regel, dass für den Parameter RebootOption die Option NoReboot ausgewählt wurde, als das AWS-RunPatchBaseline -Dokument zuletzt auf dem verwalteten Knoten ausgeführt wurde. Weitere Informationen finden Sie unter Parameter name: RebootOption . • Patch ManagerSeit dem letzten Neustart des verwalteten Knotens wurde ein Patch außerhalb von installiert.	Nicht konform

Patch-Status	Beschreibung	Compliancewert
`MISSING`	Der Patch wurde in der Baseline genehmigt, aber nicht auf dem verwalteten Knoten installiert. Wenn Sie die `AWS-RunPatchBaseline` -Dokumentaufgabe zum Scannen (nicht Installieren) konfigurieren, meldet das System diesen Status bei Patches, die beim Scan gefunden wurden, aber noch nicht installiert sind.	Nicht konform

Patch-Status	Beschreibung	Compliancewert
`NOT_APPLICABLE` [1]	Der Patch wurde in der Baseline genehmigt, der Service oder dem Feature, die den Patch verwendet, wurde aber auf dem verwalteten Knoten nicht installiert. Beispielsweise würde ein Patch für einen Webserver-Service wie Internet Information Services (IIS) `NOT_APPLICABLE` anzeigen, wenn er in der Baseline genehmigt wurde, der Webservice jedoch nicht auf dem verwalteten Knoten installiert ist. Ein Patch kann auch als `NOT_APPLICABLE` markiert sein, wenn er durch ein nachfolgendes Update ersetzt wurde. Dies bedeutet, dass das spätere Update installiert ist und das `NOT_APPLICABLE` -Update nicht mehr benötigt wird. **Note** Dieser Compliance-Status wird nur auf Windows Server-Betriebssystemen gemeldet.	Nicht zutreffend

Patch-Status	Beschreibung	Compliancewert
FAILED	Der Patch wurde an der Baseline genehmigt, konnte aber nicht installiert werden. Zum Beheben dieses Problems überprüfen Sie die Befehlsausgabe auf Informationen, die Ihnen helfen, das Problem zu verstehen.	Nicht konform

[1] Für Patches mit dem Status INSTALLED_OTHER und NOT_APPLICABLE, lässt Patch Manager einige Daten aus Abfrageergebnissen aus, die auf dem Befehl describe-instance-patches basieren, z. B. die Werte für `Classification` und `Severity`. Dadurch soll verhindert werden, dass das Datenlimit für einzelne Knoten im Inventar überschritten wird, eine Fähigkeit von. AWS Systems Manager Um alle Patch-Details anzuzeigen, können Sie den Befehl describe-available-patches nutzen.

Patchen nicht konformer verwalteter Knoten

Viele der gleichen AWS Systems Manager-Tools und -Prozesse, mit denen Sie verwaltete Knoten auf Patch-Compliance überprüfen können, können auch verwendet werden, damit Knoten die derzeit für sie geltenden Patch-Regeln erfüllen. Damit verwaltete Knoten die Patch-Compliance erfüllen, muss Patch Manager, eine Funktion von AWS Systems Manager, eine `Scan and install`-Operation ausführen. (Wenn es Ihr Ziel ist, nicht konforme verwaltete Knoten nur zu identifizieren und sie nicht zu beheben, führen Sie stattdessen eine `Scan`-Operation aus. Weitere Informationen finden Sie unter Identifizieren von nicht konformen verwalteten Knoten.)

Installieren von Patches mit Systems Manager

Sie können aus mehreren Tools wählen, um eine `Scan and install`-Operation auszuführen:

- (Empfohlen) Konfigurieren Sie eine Patch-Richtlinie in Quick Setup, einer Funktion von Systems Manager, mit der Sie fehlende Patches nach einem Zeitplan für eine gesamte Organisation, eine Teilmenge von Organisationseinheiten oder ein einzelnes AWS-Konto installieren können. Weitere Informationen finden Sie unter Patch Manager Patching-Konfiguration der Organisation.
- Erstellen Sie ein Wartungsfenster, das das Systems Manager-Dokument (SSM-Dokument) `AWS-RunPatchBaseline` in einem Run Command-Aufgabentyp verwendet. Weitere Informationen

finden Sie unter Walkthrough: Erstellen eines Wartungsfensters für das Einspielen von Patches (Konsole).

- Führen Sie `AWS-RunPatchBaseline` manuell in einer Run Command-Operation aus. Weitere Informationen finden Sie unter Ausführen von Befehlen über die Konsole.

- Installieren Sie Patches bei Bedarf mithilfe der Option Patch now (Jetzt patchen). Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten.

Vermeiden von unbeabsichtigtem Überschreiben von Patch-Compliance-Daten

Wenn Sie mehrere Arten von Vorgängen zum Scannen Ihrer Instances auf Patch-Compliance haben, überschreibt jeder Scan die Patch-Compliance-Daten vorheriger Scans. Dies kann zu unerwarteten Ergebnissen in Ihren Patch-Compliance-Daten führen.

Nehmen wir an, Sie erstellen eine Patch-Richtlinie, die jeden Tag um 02:00 Uhr Ortszeit auf Patch-Compliance scannt. Diese Patch-Richtlinie verwendet eine Patch-Baseline, die Patches als Ziel haben, deren Schweregrad mit `Critical`, `Important` und `Moderate` gekennzeichnet ist. Diese Patch-Baseline gibt auch einige speziell abgelehnte Patches an.

Nehmen Sie außerdem an, dass Sie bereits ein Wartungsfenster eingerichtet haben, um jeden Tag um 04:00 Uhr Ortszeit denselben Satz verwalteter Knoten zu scannen, die Sie nicht löschen oder deaktivieren. Die Aufgabe dieses Wartungsfensters verwendet eine andere Patch-Baseline, eine, die nur Patches mit dem Schweregrad `Critical` als Ziel hat und keine bestimmten Patches ausschließt.

Wenn dieser zweite Scan vom Wartungsfenster durchgeführt wird, werden die Patch-Compliance-Daten des ersten Scans gelöscht und durch die Patch-Compliance des zweiten Scans ersetzt.

Daher empfehlen wir dringend, nur eine automatisierte Methode zum Scannen und Installieren in Ihren Patching-Vorgängen zu verwenden. Wenn Sie Patch-Richtlinien einrichten, sollten Sie andere Methoden zum Scannen auf Patch-Compliance löschen oder deaktivieren. Weitere Informationen finden Sie unter den folgenden Themen:

- So entfernen Sie eine Patching-Aufgabe aus einem Wartungsfenster – Aktualisieren oder Abmelden von Wartungsfenster-Aufgaben (Konsole)
- So löschen Sie eine State Manager-Zuordnung – Löschen von Zuordnungen.

Um tägliche Patch-Compliance-Scans in einer Host-Verwaltungskonfiguration zu deaktivieren, gehen Sie in Quick Setup wie folgt vor:

1. Wählen Sie im Navigationsbereich Quick Setup aus.
2. Wählen Sie die zu aktualisierende Host-Management-Konfiguration aus.
3. Wählen Sie Actions, Edit configuration (Aktionen, Konfiguration bearbeiten) aus.
4. Deaktivieren Sie das Kontrollkästchen Scan instances for missing patches daily (Instances täglich auf fehlende Patches scannen).
5. Wählen Sie Aktualisieren.

> **Note**
>
> Die Verwendung von Patch now (Jetzt patchen) zum Überprüfen eines verwalteten Knotens auf Compliance führt auch zu einem Überschreiben von Patch-Compliance-Daten.

On-Demand-Patchen von verwalteten Knoten

Verwenden Sie die Option Patch now (jetzt patchen) in Patch Manager, einer Funktion von AWS Systems Manager, um Patching-Vorgänge auf Abruf über die Systems Manager-Konsole auszuführen. Dies bedeutet, dass Sie keinen Zeitplan erstellen müssen, um den Compliance-Status Ihrer verwalteten Knoten zu aktualisieren oder Patches auf nicht kompatiblen Knoten zu installieren. Die Systems Manager-Konsole muss auch nicht zwischen Patch Manager und Maintenance Windows, einer Funktion von AWS Systems Manager, wechseln, um ein geplantes Patching-Fenster einzurichten oder zu ändern.

Patch now (Jetzt patchen) ist besonders nützlich, wenn Sie so schnell wie möglich Zero-Day-Updates anwenden oder andere kritische Patches auf Ihren verwalteten Knoten installieren müssen.

> **Note**
>
> Patching on Demand wird für jeweils AWS-Konto ein einzelnes AWS-Region Paar unterstützt. Es kann nicht mit Patching-Vorgängen verwendet werden, die auf Patch-Richtlinien basieren. Wir empfehlen die Verwendung von Patch-Richtlinien, um sicherzustellen, dass alle Ihre verwalteten Knoten die Compliance einhalten. Weitere Informationen zur Arbeit mit Patch-Richtlinien finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien.

Themen

- So funktioniert „Patch now“ (Jetzt patchen)
- Ausführen von „Patch now“ (Jetzt patchen)

So funktioniert „Patch now“ (Jetzt patchen)

Um Patch now (Jetzt patchen) auszuführen, müssen Sie nur zwei erforderliche Einstellungen angeben:

- Ob nur nach fehlenden Patches gescannt werden soll oder ob Patches auf Ihren verwalteten Knoten gescannt und installiert werden sollen
- Auf welchen verwalteten Knoten die Operation ausgeführt werden soll

Wenn die Operation Patch now (Jetzt patchen) läuft, bestimmt sie, welche Patch-Baseline auf die gleiche Weise verwendet werden soll, wie eine für andere Patching-Operationen ausgewählt wird. Wenn ein verwalteter Knoten einer Patch-Gruppe zugeordnet ist, wird die für diese Gruppe angegebene Patch-Baseline verwendet. Wenn der verwaltete Knoten nicht einer Patch-Gruppe zugeordnet ist, verwendet die Operation die Patch-Baseline, die derzeit als Standard für den Betriebssystemtyp des verwalteten Knotens festgelegt ist. Dabei kann es sich um eine vordefinierte Baseline oder um die benutzerdefinierte Baseline handeln, die Sie als Standard festgelegt haben. Weitere Informationen zur Patch-Baseline-Auswahl finden Sie unter Patch-Gruppen.

Zu den Optionen, die Sie für Patch now (Jetzt patchen) angeben können, gehört, ob verwaltete Knoten nach dem Patchen neu gestartet werden sollen, indem Sie einen Amazon Simple Storage Service (Amazon S3)-Bucket zum Speichern von Protokolldaten für den Patchvorgang angeben und Systems-Manager-Dokumente (SSM-Dokumente) als Lebenszyklus-Hooks während des Patchings ausführen.

Parallelitäts- und Fehlerschwellenwerte für „Patch now“ (Jetzt patchen)

Für Patch now-Operationen (Jetzt patchen) werden Optionen für Parallelitäts- und Fehlerschwellen von Patch Manager gehandhabt. Sie müssen weder angeben, wie viele verwaltete Knoten gleichzeitig gepatcht werden sollen, noch wie viele Fehler zulässig sind, bevor die Operation fehlschlägt. Patch Manager wendet die Einstellungen für Parallelitäts- und Fehlerschwellenwert an, die in den folgenden Tabellen beschrieben werden, wenn Sie On-Demand-Patches anwenden.

> **Important**
>
> Die folgenden Schwellenwerte gelten für nur für `Scan and install`-Operationen. Für `Scan`-Operationen versucht Patch Manager bis zu 1 000 Knoten gleichzeitig zu scannen und weiter zu scannen, bis bis zu 1 000 Fehler aufgetreten sind.

Gleichzeitigkeit: Installationsvorgänge

Die Gesamtanzahl von verwalteten Knoten in der Operation Patch now (Jetzt patchen)	Anzahl der gleichzeitig gescannten oder gepatchten verwalteten Knoten
Weniger als 25	1
25-100	5 %
101 bis 1.000	8%
Mehr als 1.000	10 %

Fehlerschwellenwert: Installationsvorgänge

Die Gesamtanzahl von verwalteten Knoten in der Operation Patch now (Jetzt patchen)	Anzahl der zulässigen Fehler, bevor der Vorgang fehlschlägt
Weniger als 25	1
25-100	5
101 bis 1.000	10
Mehr als 1.000	10

Verwenden von „Patch now“ (Jetzt patchen)-Lebenszyklus-Hooks

Patch now (Jetzt patchen) bietet Ihnen die Möglichkeit, SSM Command-Dokumente als Lebenszyklus-Hooks während eines `Install`-Patch-Vorgang auszuführen. Sie können diese Hooks für Aufgaben wie das Herunterfahren von Anwendungen vor dem Patchen oder Ausführen von Zustandsprüfungen für Ihre Anwendungen nach dem Patchen oder nach einem Neustart verwenden.

Weitere Informationen über das Verwenden von Lebenszyklus-Hooks finden Sie unter Informationen über das `AWS-RunPatchBaselineWithHooks` SSM-Dokument.

In der folgenden Tabelle werden die Lebenszyklus-Hooks aufgeführt, die für jede der drei Patch now-Neustartoptionen (Jetzt patchen) aufgelistet sowie Beispielanwendungen für jeden Hook.

Lebenszyklus-Hooks und Beispielanwendungen

Neustartoption	Hook: Vor Installation	Hook: Nach Installation	Hook: Beim Verlassen	Hook: Nach geplantem Neustart
Bei Bedarf neu starten	Führen Sie ein SSM-Dokument aus, bevor das Patchen beginnt. Anwendungsbeispiel: Fahren Sie Anwendungen sicher herunter, bevor der Patchvorgang beginnt.	Führen Sie ein SSM-Dokument am Ende der Patching-Operation und vor dem Neustart des verwalteten Knoten aus. Anwendungsbeispiel: Führen Sie Vorgänge wie die Installation von Drittanbieter-Anwendungen vor einem potenziellen Neustart aus.	Führen Sie ein SSM-Dokument aus, nachdem die Patching-Operation abgeschlossen und die Instances neu gestartet wurden. Anwendungsbeispiel: Stellen Sie sicher, dass Anwendungen nach dem Patchen wie erwartet ausgeführt werden.	Nicht verfügbar
Meine Instances nicht neu starten	Wie oben.	Führen Sie ein SSM-Dokument am Ende des Patching-Vorgangs aus.	Nicht verfügbar	Nicht verfügbar

Neustartoption	Hook: Vor Installation	Hook: Nach Installation	Hook: Beim Verlassen	Hook: Nach geplantem Neustart
		Anwendungsbeispiel: Stellen Sie sicher, dass Anwendungen nach dem Patchen wie erwartet ausgeführt werden.		
Neustartzeit planen	Wie oben.	Dasselbe wie für Meine Instances nicht neu starten.	Nicht verfügbar	Führen Sie sofort nach Abschluss eines geplanten Neustarts ein SSM-Dokument aus. Anwendungsbeispiel: Stellen Sie sicher, dass Anwendungen nach dem Neustart wie erwartet ausgeführt werden.

Ausführen von „Patch now“ (Jetzt patchen)

Mit dem folgenden Verfahren können Sie On-Demand-Patches auf Ihre verwalteten Knoten anwenden.

So führen Sie „Patch now“ (Jetzt patchen) aus

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie entweder auf der AWS Systems Manager Patch ManagerSeite oder auf der Seite Patch-Baselines, je nachdem, welche Seite geöffnet wird, die Option Jetzt patchen aus.
4. Für Patch-Operation wählen Sie eine der folgenden Optionen aus:

   - Scan (Scannen): Patch Manager findet die Patches, die in Ihren verwalteten Knoten fehlen, installiert sie aber nicht. Sie können die Ergebnisse im Compliance-Dashboard oder in anderen Tools, die Sie zum Anzeigen der Patch-Compliance verwenden, anzeigen.
   - Scan and install (Scannen und installieren): Patch Manager findet die Patches, die in Ihren verwalteten Knoten fehlen, und installiert sie.
5. Führen Sie diesen Schritt nur aus, wenn Sie im vorherigen Schritt Scan und Installation ausgewählt haben. Wählen Sie bei Neustartoption eine der folgenden Optionen aus:

   - Reboot if needed (Bei Bedarf neu starten): Nach der Installation startet Patch Manager verwaltete Knoten nur dann neu, wenn dies zum Abschluss einer Patch-Installation erforderlich ist.
   - Don‘t reboot my instances (Meine Instances nicht neu starten): Nach der Installation startet Patch Manager verwaltete Knoten nicht neu. Sie können verwaltete Knoten manuell neu starten, wenn Sie Neustarts außerhalb von Patch Manager auswählen oder verwalten.
   - Schedule a reboot time (Neustartzeit planen): Geben Sie das Datum, die Uhrzeit und die UTC-Zeitzone an, wenn Patch Manager Ihre verwalteten Knoten neu starten sollen. Nachdem Sie den Patch now (Jetzt patchen)-Vorgang ausgeführt haben, wird der geplante Neustart als Zuordnung in State Manager mit dem Namen `AWS-PatchRebootAssociation` gelistet.
6. Wählen Sie unter Instances to patch (Zu patchende Instances) eine der folgenden Optionen aus:

   - Alle Instanzen patchen: Patch Manager Führt den angegebenen Vorgang auf allen verwalteten Knoten AWS-Konto in Ihrer aktuellen Version aus. AWS-Region
   - Patch only the target instances I specify (Nur die von mir angegebenen Ziel-Instances patchen): Sie geben im nächsten Schritt an, welche verwalteten Knoten anvisiert werden sollen.
7. Führen Sie diesen Schritt nur aus, wenn Sie im vorherigen Schritt Nur die von mir angegebenen Ziel-Instances patchen ausgewählt haben. Identifizieren Sie für den Abschnitt Target selection

(Zielauswahl) die Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Knoten manuell auswählen oder eine Ressourcengruppe angeben.

> **Note**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.
> Wenn Sie sich für eine Ressourcengruppe entscheiden, beachten Sie, dass Ressourcengruppen, die auf einem AWS CloudFormation Stack basieren, trotzdem mit dem `aws:cloudformation:stack-id` Standard-Tag gekennzeichnet werden müssen. Wenn es entfernt wurde, kann Patch Manager möglicherweise nicht ermitteln, welche verwalteten Knoten zur Ressourcengruppe gehören.

8. (Optional) Für Patch-Protokollspeicher wählen Sie, wenn Sie Protokolle aus diesem Patchvorgang erstellen und speichern möchten, den S3-Bucket zum Speichern der Protokolle aus.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen der für Systems Manager erforderlichen IAM-Servicerolle in Hybrid- und Multicloud-Umgebungen. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

9. (Optional) Wenn Sie SSM-Dokumente als Lebenszyklus-Hooks an bestimmten Punkten des Patching-Vorgangs ausführen möchten, gehen Sie wie folgt vor:

   - Klicken Sie auf Verwenden von Lebenszyklus-Hooks.
   - Wählen Sie für jeden verfügbaren Hook das SSM-Dokument aus, das am angegebenen Punkt des Vorgangs ausgeführt werden soll:

- Vor Installation
- Nach Installation
- Beim Verlassen
- Nach geplantem Neustart

> **Note**
>
> Das Standarddokument, `AWS-Noop`, führt keine Vorgänge aus.

10. Wählen Sie Patch now (Jetzt patchen) aus.

    Die Seite Association execution summary (Zusammenfassung der Zuordnungsausführung) wird geöffnet. (Patch verwendet jetzt Assoziationen inState Manager, eine Fähigkeit von AWS Systems Manager, für seine Operationen.) Im Bereich Operation summary (Operationsübersicht) können Sie den Scan- oder Patch-Status auf den von Ihnen angegebenen verwalteten Knoten überwachen.

Arbeiten mit Patch-Baselines

Eine Patch-Baseline in Patch Manager, einer Funktion von AWS Systems Manager, definiert, welche Patches für die Installation in Ihren verwalteten Knoten genehmigt sind. Sie können für Patches einzeln angeben, ob sie genehmigt oder abgelehnt werden. Sie können auch automatische Genehmigungsregeln erstellen, um festzulegen, dass bestimmte Arten von Updates (z. B. wichtige Updates), automatisch genehmigt werden. Die Liste mit den Ablehnungen setzt sowohl die Regeln als auch die Liste mit Genehmigungen außer Kraft. Wenn Sie ausschließlich eine Liste mit genehmigten Patches verwenden möchten, um spezifische Pakete zu installieren, entfernen Sie erst alle automatischen Genehmigungsregeln. Wenn Sie explizit einen Patch als abgelehnt kennzeichnen, wird er nicht genehmigt oder installiert, selbst wenn er mit allen Kriterien in einer automatischen Genehmigungsregel übereinstimmt. Außerdem wird ein Patch nur dann auf einem verwalteten Knoten installiert, wenn er für die Software auf dem Knoten geeignet ist, auch wenn der Patch anderweitig für den verwalteten Knoten genehmigt wurde.

Themen

- So zeigen Sie von AWS vordefinierte Patch-Baselines an
- Arbeiten mit benutzerdefinierten Patch-Baselines
- Festlegen einer vorhandenen Patch-Baseline als Standard

Weitere Informationen

- Über Patch-Baselines

So zeigen Sie von AWS vordefinierte Patch-Baselines an

Patch Manager, eine Funktion von AWS Systems Manager, beinhaltet eine vordefinierte Patch-Baseline für jedes Betriebssystem, das von unterstützt wird. Patch Manager Sie können diese Patch-Baselines verwenden (Sie können sie jedoch nicht anpassen) oder Sie können eine eigene Patch-Baseline erstellen. Nachfolgend wird beschrieben, wie Sie eine vordefinierte Patch-Baseline anzeigen, um sie auf Ihre Anforderungen hin zu überprüfen. Weitere Informationen zu Patch-Baselines finden Sie unter Info zu vordefinierten und benutzerdefinierten Patch-Baselines.

So zeigen Sie AWS vordefinierte Patch-Baselines an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie in der Liste der Patch-Baselines die Baseline-ID einer der vordefinierten Patch-Baselines aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen, wählen Sie die Registerkarte Patch-Baselines und wählen Sie dann die Baseline-ID einer der vordefinierten Patch-Baselines.

   > **Note**
   >
   > Für Windows Server werden drei vordefinierte Patch-Baselines bereitgestellt. Die Patch-Baselines `AWS-DefaultPatchBaseline` und `AWS-WindowsPredefinedPatchBaseline-OS` unterstützen nur Betriebssystemupdates auf dem Windows-Betriebssystem selbst. `AWS-DefaultPatchBaseline` wird als Standard-Patch-Baseline für von Windows Server verwalteten Knoten verwendet, es sei denn, Sie geben eine andere Patch-Baseline an. Die Konfigurationseinstellungen in diesen beiden Patch-Baselines sind identisch. Die neuere der beiden, `AWS-WindowsPredefinedPatchBaseline-OS`, wurde erstellt, um sie von der dritten vordefinierten Patch-Baseline für Windows Server zu unterscheiden. Diese Patch-Baseline, `AWS-WindowsPredefinedPatchBaseline-OS-Applications`, kann

verwendet werden, um Patches sowohl auf das Windows Server-Betriebssystem als auch auf unterstützte Anwendungen, die von Microsoft veröffentlicht wurden, anzuwenden.
Weitere Informationen finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.

4. Im Abschnitt Genehmigungsregeln überprüfen Sie die Konfiguration der Patch-Baseline-Konfiguration.
5. Wenn die Konfiguration für Ihre verwalteten Knoten geeignet ist, können Sie direkt mit dem Verfahren Arbeiten mit Patch-Gruppen fortfahren.

   –oder–

   Fahren Sie zum Erstellen einer eigenen Standard-Patch-Baseline mit dem Thema Arbeiten mit benutzerdefinierten Patch-Baselines fort.

Arbeiten mit benutzerdefinierten Patch-Baselines

Patch Manager, eine Funktion von AWS Systems Manager, enthält eine vordefinierte Patch-Baseline für jedes der von Patch Manager unterstützten Betriebssysteme. Sie können diese Patch-Baselines verwenden (Sie können sie jedoch nicht anpassen) oder Sie können eine eigene Patch-Baseline erstellen.

In den folgenden Verfahren wird beschrieben, wie Sie eigene benutzerdefinierte Patch-Baselines erstellen, aktualisieren und löschen. Weitere Informationen zu Patch-Baselines finden Sie unter Info zu vordefinierten und benutzerdefinierten Patch-Baselines.

Themen

- So erstellen Sie eine benutzerdefinierte Patch-Baseline (Linux)
- Erstellen einer benutzerdefinierten Patch-Baseline (macOS)
- Erstellen einer benutzerdefinierten Patch-Baseline (Windows)
- Aktualisieren oder Löschen einer benutzerdefinierten Patch-Baseline

So erstellen Sie eine benutzerdefinierte Patch-Baseline (Linux)

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte Patch-Baseline für verwaltete Linux-Knoten in zu erstellenPatch Manager, eine Fähigkeit von AWS Systems Manager.

Informationen zum Erstellen einer Patch-Baseline für macOS-verwaltete Knoten finden Sie unter Erstellen einer benutzerdefinierten Patch-Baseline (macOS). Informationen zum Erstellen einer Patch-Baseline für Windows-verwaltete Knoten finden Sie unter Erstellen einer benutzerdefinierten Patch-Baseline (Windows).

So erstellen Sie eine benutzerdefinierte Patch-Baseline für Linux-verwaltete Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patch-Baselines und dann Patch-Baseline erstellen aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen, wechseln Sie zur Registerkarte Patch-Baselines und wählen Sie dann Patch-Baseline erstellen.
4. Geben Sie im Feld Name (Name) einen Namen für die neue Patch-Baseline ein, z. B. `MyRHELPatchBaseline`.
5. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für diese Patch-Baseline ein.
6. Wählen Sie unter Operating system (Betriebssystem) ein Betriebssystem aus, z. B. `Red Hat Enterprise Linux`.
7. Wenn Sie die Patch-Baseline direkt nach dem Erstellen als Standard für das ausgewählte Betriebssystem verwenden möchten, aktivieren Sie das Kontrollkästchen Set this patch baseline as the default patch baseline for ***`operating system name`*** instances (Diese Patch-Baseline als Standard-Patch-Baseline für Name des Betriebssystems-Instances festlegen).

   > **Note**
   >
   > Diese Option ist nur verfügbar, wenn Sie vor der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben. Weitere Informationen zum Festlegen einer vorhandenen Patch-Baseline als Standard finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.

8. Erstellen Sie im Abschnitt Approval Rules for operating-systems (Genehmigungsregeln für Betriebssysteme) unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

   - Produkte: Die Version der Betriebssysteme, auf die sich die Genehmigungsregel bezieht, z. B. `RedhatEnterpriseLinux7.4`. Die Standardauswahl ist `All`.
   - Classification (Klassifizierung): Der Typ der Patches, auf die sich die Genehmigungsregel bezieht, z. B. `Security` oder `Enhancement`. Die Standardauswahl ist `All`.

   > **Tip**
   >
   > Sie können eine Patch-Baseline konfigurieren, um zu steuern, ob Nebenversions-Upgrades für Linux installiert werden, z. B. RHEL 7.8. Nebenversionsupgrades können vom Patch Manager automatisch installiert werden, wenn das Update im entsprechenden Repository verfügbar ist.
   > Im Fall von Linux-Betriebssystemen werden Nebenversionsupgrades nicht konsistent klassifiziert. Sie können als Fehlerbehebungen oder Sicherheitsupdates klassifiziert (oder nicht klassifiziert) werden, selbst innerhalb derselben Kernel-Version. Im Folgenden werden einige Optionen aufgelistet, mit denen Sie steuern können, ob sie von einer Patch-Baseline installiert werden.
   >
   > - Option 1: Die umfassendste Genehmigungsregel, die sicherzustellt, dass Nebenversionsupgrades installiert werden, wenn verfügbar, besteht in der Angabe von Classification (Klassifizierung) als `All` (*) und der Auswahl der Option Include nonsecurity updates (Auch andere Updates als Sicherheitsupdates einschließen).
   > - Option 2: Um die Installation von Patches für eine Betriebssystemversion sicherzustellen, können Sie ein Platzhalterzeichen (*) verwenden, um das Kernel-Format im Abschnitt Patch exceptions (Patch-Ausnahmen) der Baseline anzugeben. Zum Beispiel ist das Kernel-Format für RHEL 7.* `kernel-3.10.0-*.el7.x86_64`.
   >
   >   Geben Sie `kernel-3.10.0-*.el7.x86_64` in der Liste Approved patches (Genehmigte Patches) in Ihrer Patch-Baseline ein, um die Anwendung aller Patches einschließlich Nebenversionsupgrades auf Ihren von RHEL 7.* verwalteten Knoten sicherzustellen. (Wenn Sie den genauen Paketnamen eines Nebenversionspatches kennen, können Sie diesen stattdessen eingeben.)
   > - Option 3: Mithilfe des InstallOverrideListParameters im `AWS-RunPatchBaseline` Dokument haben Sie die größtmögliche Kontrolle darüber, welche Patches auf

Ihre verwalteten Knoten angewendet werden, einschließlich kleinerer Versions-Upgrades. Weitere Informationen finden Sie unter Informationen über das `AWS-RunPatchBaseline` SSM-Dokument.

- Severity (Schweregrad): Der Schweregradwert von Patches, auf den die Regel anzuwenden ist, z. B. `Critical`. Die Standardauswahl ist `All`.
- Auto-approval (Automatische Genehmigung): Die Methode zum Auswählen von Patches für die automatische Genehmigung.

> **Note**
>
> Da es nicht möglich ist, die Veröffentlichungsdaten von Updatepaketen für Ubuntu Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

  - Approve patches after a specified number of days (Patches nach einer bestimmten Anzahl von Tagen genehmigen): Die Anzahl der Tage, die der Patch Manager warten muss, nachdem ein Patch veröffentlicht oder zuletzt aktualisiert wurde, bevor ein Patch automatisch genehmigt wird. Sie können jede Ganzzahl von Null (0) bis 360 eingeben. Für die meisten Szenarien empfehlen wir, nicht länger als 100 Tage zu warten.
  - Approve patches released up to a specific date (Patches genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden): Das Datum der Patch-Veröffentlichung, an dem der Patch Manager automatisch alle Patches anwendet, die bis zu diesem Datum veröffentlicht oder aktualisiert wurden. Wenn Sie beispielsweise den 07. Juli 2023 angeben, werden Patches, die am oder nach dem 08. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden, nicht automatisch installiert.
- (Optional) Konformitätsbericht : Der Schweregrad, den Sie Patches zuweisen möchten, die von der Baseline genehmigt wurden (z. B. `Critical` oder `High`).

> **Note**
>
> Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann entspricht der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

- Include non-security updates (Nicht sicherheitsrelevante Updates einbeziehen): Aktivieren Sie das Kontrollkästchen zum Installieren von nicht sicherheitsrelevanten Linux-Betriebssystem-Patches, die im Quell-Repository verfügbar gemacht wurden, zusätzlich zu den sicherheitsrelevanten Patches.

> **Note**
>
> Das Kontrollkästchen muss für SUSE Linux Enterprise Server (SLES), nicht aktiviert werden, da sicherheitsrelevante und nicht sicherheitsrelevante Patches standardmäßig auf SLES-verwaltete Knoten installiert werden. Weitere Informationen finden Sie im Content für SLES unter Wie Sicherheitspatches ausgewählt werden.

Weitere Informationen zum Arbeiten mit Genehmigungsregeln in einer benutzerdefinierten Patch-Baseline finden Sie unter Info zu benutzerdefinierten Baselines.

9. Wenn Sie zusätzlich zu den Patches, die Ihren Genehmigungsregeln entsprechen, alle Patches ausdrücklich genehmigen möchten, gehen Sie im Abschnitt Patch exceptions (Patch-Ausnahmen) wie folgt vor:

   - Geben Sie im Feld Approved patches (Genehmigte Patches) eine durch Komma getrennte Liste der Patches ein, die Sie genehmigen möchten.

   > **Note**
   >
   > Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

   - (Optional) Weisen Sie in der Liste Approved patches compliance level (Compliance-Stufe genehmigter Patches) den Patches in der Liste eine Compliance-Stufe zu.
   - Wenn genehmigte Patches, die Sie angeben, nicht sicherheitsbezogen sind, wählen Sie das Kästchen Genehmigte Patches umfassen nicht sicherheitsrelevante Updates aus, damit diese Patches ebenfalls auf Ihrem Linux-Betriebssystem installiert werden.

10. Wenn Sie Patches ablehnen möchten, die ansonsten Ihren Genehmigungsregeln entsprechen, gehen Sie im Abschnitt Patch exceptions (Patch-Ausnahmen) wie folgt vor:

- Geben Sie im Feld Rejected patches (Abgelehnte Patches) eine durch Komma getrennte Liste der Patches ein, die Sie ablehnen möchten.

  > **Note**
  >
  > Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

- Wählen Sie in der Liste Rejected patches action (Aktion für abgelehnte Patches) die Aktion aus, die Patch Manager für Patches in der Liste Rejected patches (Abgelehnte Patches) ausführen soll.
  - Allow as dependency (Als Abhängigkeit zulassen): Ein Paket in der Liste Rejected patches (Abgelehnte Patches) wird nur installiert, wenn es sich um eine Abhängigkeit eines anderen Pakets handelt. Es gilt als konform mit der Patch-Baseline und sein Status wird als gemeldet InstalledOther. Dies ist die Standardaktion, wenn keine Option ausgewählt ist.
  - Blockieren: Pakete in der Liste der abgelehnten Patches sowie Pakete, die diese als Abhängigkeiten enthalten, werden Patch Manager unter keinen Umständen installiert. Wenn ein Paket installiert wurde, bevor es zur Liste der abgelehnten Patches hinzugefügt wurde, oder wenn es außerhalb oder Patch Manager danach installiert wird, gilt es als nicht konform mit der Patch-Baseline und sein Status wird als gemeldet. InstalledRejected

11. (Optional) Wenn Sie alternative Patch-Repositorys für verschiedene Versionen eines Betriebssystems angeben möchten, z. B. AmazonLinux2016.03 und AmazonLinux2017.09, gehen Sie für jedes Produkt im Abschnitt Patchquellen wie folgt vor:

    - Geben Sie in Name (Name) einen Namen ein, um Sie bei der Identifizierung der Quellkonfiguration zu unterstützen.
    - Wählen Sie unter Product (Produkt) die Version der Betriebssysteme aus, für die das Patch-Quell-Repository bestimmt ist, z. B. `RedhatEnterpriseLinux7.4`.
    - Geben Sie unter Configuration den Wert der zu verwendenden Yum-Repository-Konfiguration im folgenden Format ein:

      ```
      [main]
      name=MyCustomRepository
      baseurl=https://my-custom-repository
      enabled=1
      ```

> **Tip**
>
> Informationen zu anderen Optionen für Ihre Yum-Repository-Konfiguration finden Sie unter dnf.conf (5).

Wählen Sie Add another source aus, um ein Quell-Repository für jede zusätzliche Betriebssystemversion anzugeben, bis maximal 20.

Weitere Informationen über alternative Quell-Patch-Repositorys finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux).

12. (Optional) Wählen Sie für Manage tags (Tags verwalten) ein oder mehrere Tag-Schlüsselname/Wertpaare für die Patch-Baseline aus.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise eine Patch-Baseline kennzeichnen, um den Schweregrad der angegebenen Patches, die Betriebssystemfamilie, auf die sie sich bezieht, und den Umgebungstyp zu identifizieren. In diesem Fall können Sie etwa Tags mit den folgenden Schlüsselnamen/Wertpaaren angeben:

    - `Key=PatchSeverity,Value=Critical`
    - `Key=OS,Value=RHEL`
    - `Key=Environment,Value=Production`

13. Wählen Sie die Option Create Patch Baseline.

Erstellen einer benutzerdefinierten Patch-Baseline (macOS)

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte Patch-Baseline für macOS verwaltete Knoten in zu erstellenPatch Manager, eine Fähigkeit von AWS Systems Manager.

Informationen zum Erstellen einer Patch-Baseline für Windows Server-verwaltete Knoten finden Sie unter Erstellen einer benutzerdefinierten Patch-Baseline (Windows). Informationen zum Erstellen einer Patch-Baseline für Linux-verwaltete Knoten finden Sie unter So erstellen Sie eine benutzerdefinierte Patch-Baseline (Linux).

> **Note**
>
> macOSwird nicht in allen unterstützt AWS-Regionen. Weitere Informationen zur Amazon EC2-Unterstützung für macOS finden Sie unter Amazon EC2 Mac-Instances im Amazon EC2 EC2-Benutzerhandbuch.

So erstellen Sie eine benutzerdefinierte Patch-Baseline für macOS-verwaltete Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patch-Baselines und dann Patch-Baseline erstellen aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen, wechseln Sie zur Registerkarte Patch-Baselines und wählen Sie dann Patch-Baseline erstellen.
4. Geben Sie im Feld Name (Name) einen Namen für die neue Patch-Baseline ein, z. B. `MymacOSPatchBaseline`.
5. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für diese Patch-Baseline ein.
6. Wählen Sie unter Operating system (Betriebssystem) die Option macOS aus.
7. Wenn Sie die Patch-Baseline direkt nach dem Erstellen als Standard für macOS verwenden möchten, aktivieren Sie das Kontrollkästchen Set this patch baseline as the default patch baseline for macOS instances (Diese Patch-Baseline als Standard-Patch-Baseline für macOs-Instances festlegen).

   > **Note**
   >
   > Diese Option ist nur verfügbar, wenn Sie vor der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben. Weitere Informationen zum Festlegen einer vorhandenen Patch-Baseline als Standard finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.

8. Erstellen Sie im Abschnitt Approval Rules for operating-systems (Genehmigungsregeln für Betriebssysteme) unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

   - Produkte: Die Version der Betriebssysteme, auf die sich die Genehmigungsregel bezieht, z. B. `Mojave10.14.1` oder `Catalina10.15.1`. Die Standardauswahl ist `All`.

     > **Note**
     >
     > Das Open-Source-Softwarepaketverwaltungssystem Homebrew hat die Unterstützung für macOS 10.14.x (Mojave) und 10.15.x (Catalina) eingestellt. Aus diesem Grund werden Patch-Operationen für diese Versionen derzeit nicht unterstützt.

   - Klassifizierung: Der oder die Paketmanager, auf den/die während des Patchvorgangs Pakete angewendet werden sollen. Sie können aus den folgenden Optionen auswählen:
     - softwareupdate
     - installer
     - brew
     - brew cask

     Die Standardauswahl ist `All`.

   - (Optional) Konformitätsbericht : Der Schweregrad, den Sie Patches zuweisen möchten, die von der Baseline genehmigt wurden (z. B. `Critical` oder `High`).

     > **Note**
     >
     > Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann entspricht der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

   - Include non-security updates (Nicht sicherheitsrelevante Updates einbeziehen): Aktivieren Sie das Kontrollkästchen zum Installieren von nicht sicherheitsrelevanten Betriebssystem-Patches, die im Quell-Repository verfügbar gemacht wurden, zusätzlich zu den sicherheitsrelevanten Patches.

Weitere Informationen zum Arbeiten mit Genehmigungsregeln in einer benutzerdefinierten Patch-Baseline finden Sie unter Info zu benutzerdefinierten Baselines.

9. Wenn Sie zusätzlich zu den Patches, die Ihren Genehmigungsregeln entsprechen, alle Patches ausdrücklich genehmigen möchten, gehen Sie im Abschnitt Patch exceptions (Patch-Ausnahmen) wie folgt vor:

   - Geben Sie im Feld Approved patches (Genehmigte Patches) eine durch Komma getrennte Liste der Patches ein, die Sie genehmigen möchten.

     > **Note**
     >
     > Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

   - (Optional) Weisen Sie in der Liste Approved patches compliance level (Compliance-Stufe genehmigter Patches) den Patches in der Liste eine Compliance-Stufe zu.
   - Wenn genehmigte Patches, die Sie angeben, nicht sicherheitsbezogen sind, wählen Sie das Kästchen Genehmigte Patches umfassen nicht sicherheitsrelevante Updates aus, damit diese Patches ebenfalls auf Ihrem macOS-Betriebssystem installiert werden.

10. Wenn Sie Patches ablehnen möchten, die ansonsten Ihren Genehmigungsregeln entsprechen, gehen Sie im Abschnitt Patch exceptions (Patch-Ausnahmen) wie folgt vor:

    - Geben Sie im Feld Rejected patches (Abgelehnte Patches) eine durch Komma getrennte Liste der Patches ein, die Sie ablehnen möchten.

      > **Note**
      >
      > Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

    - Wählen Sie in der Liste Rejected patches action (Aktion für abgelehnte Patches) die Aktion aus, die Patch Manager für Patches in der Liste Rejected patches (Abgelehnte Patches) ausführen soll.

- Allow as dependency (Als Abhängigkeit zulassen): Ein Paket in der Liste Rejected patches (Abgelehnte Patches) wird nur installiert, wenn es sich um eine Abhängigkeit eines anderen Pakets handelt. Es gilt als konform mit der Patch-Baseline und sein Status wird als gemeldet InstalledOther. Dies ist die Standardaktion, wenn keine Option ausgewählt ist.
- Blockieren: Pakete in der Liste der abgelehnten Patches sowie Pakete, die diese als Abhängigkeiten enthalten, werden Patch Manager unter keinen Umständen installiert. Wenn ein Paket installiert wurde, bevor es zur Liste der abgelehnten Patches hinzugefügt wurde, oder wenn es außerhalb oder Patch Manager danach installiert wird, gilt es als nicht konform mit der Patch-Baseline und sein Status wird als gemeldet. InstalledRejected

11. (Optional) Wählen Sie für Manage tags (Tags verwalten) ein oder mehrere Tag-Schlüsselname/Wertpaare für die Patch-Baseline aus.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise eine Patch-Baseline kennzeichnen, um den Schweregrad der angegebenen Patches, den Paketmanager, auf den sie sich bezieht, und den Umgebungstyp zu identifizieren. In diesem Fall können Sie etwa Tags mit den folgenden Schlüsselnamen/Wertpaaren angeben:

    - `Key=PatchSeverity,Value=Critical`
    - `Key=PackageManager,Value=softwareupdate`
    - `Key=Environment,Value=Production`

12. Wählen Sie die Option Create Patch Baseline.

Erstellen einer benutzerdefinierten Patch-Baseline (Windows)

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte Patch-Baseline für verwaltete Windows-Knoten in zu erstellenPatch Manager, eine Funktion von AWS Systems Manager.

Informationen zum Erstellen einer Patch-Baseline für Linux-verwaltete Knoten finden Sie unter So erstellen Sie eine benutzerdefinierte Patch-Baseline (Linux). Informationen zum Erstellen einer Patch-Baseline für macOS-verwaltete Knoten finden Sie unter Erstellen einer benutzerdefinierten Patch-Baseline (macOS).

Ein Beispiel für das Erstellen einer Patch-Baseline, die auf die Installation von Windows Service Packs eingeschränkt ist, finden Sie unter So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs (Konsole).

So erstellen Sie eine benutzerdefinierte Patch-Baseline (Windows)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patch-Baselines und dann Patch-Baseline erstellen aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen, wechseln Sie zur Registerkarte Patch-Baselines und wählen Sie dann Patch-Baseline erstellen.
4. Geben Sie im Feld Name (Name) einen Namen für die neue Patch-Baseline ein, z. B. `MyWindowsPatchBaseline`.
5. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für diese Patch-Baseline ein.
6. Wählen Sie unter Operating system (Betriebssystem) die Option `Windows` aus.
7. Wenn Sie diese Patch-Baseline direkt nach dem Erstellen als Standard für Windows verwenden möchten, wählen Sie Set this patch baseline as the default patch baseline for Windows Server instances (Diese Patch-Baseline als Standard-Patch-Baseline für Windows Server-Instances festlegen) aus.

   > **Note**
   >
   > Diese Option ist nur verfügbar, wenn Sie vor der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben. Weitere Informationen zum Festlegen einer vorhandenen Patch-Baseline als Standard finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.
8. Erstellen Sie im Abschnitt Approval Rules for operating-systems (Genehmigungsregeln für Betriebssysteme) unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

   - Produkte: Die Version der Betriebssysteme, auf die sich die Genehmigungsregel bezieht, z. B. `WindowsServer2012`. Die Standardauswahl ist `All`.
   - Classification (Klassifizierung): Der Typ der Patches, auf die sich die Genehmigungsregel bezieht, z. B. `CriticalUpdates`, `Drivers` und `Tools`. Die Standardauswahl ist `All`.

> **Tip**
>
> Sie können Windows Service Pack-Installationen in die Genehmigungsregeln einschließen, indem Sie die `ServicePacks` einschließen oder `All` in der Liste Classification (Klassifizierung) auswählen. Ein Beispiel finden Sie unter So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs (Konsole).

- Severity (Schweregrad): Der Schweregradwert von Patches, auf den die Regel anzuwenden ist, z. B. `Critical`. Die Standardauswahl ist `All`.
- Auto-approval (Automatische Genehmigung): Die Methode zum Auswählen von Patches für die automatische Genehmigung.
  - Approve patches after a specified number of days (Patches nach einer bestimmten Anzahl von Tagen genehmigen): Die Anzahl der Tage, die der Patch Manager warten muss, nachdem ein Patch veröffentlicht oder aktualisiert wurde, bevor ein Patch automatisch genehmigt wird. Sie können jede Ganzzahl von Null (0) bis 360 eingeben. Für die meisten Szenarien empfehlen wir, nicht länger als 100 Tage zu warten.
  - Approve patches released up to a specific date (Patches genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden): Das Datum der Patch-Veröffentlichung, an dem der Patch Manager automatisch alle Patches anwendet, die bis zu diesem Datum veröffentlicht oder aktualisiert wurden. Wenn Sie beispielsweise den 07. Juli 2023 angeben, werden Patches, die am oder nach dem 08. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden, nicht automatisch installiert.
- (Optional) Compliance-Bericht : Der Schweregrad, den Sie Patches zuweisen möchten, die von der Baseline genehmigt wurden (z. B. `High`).

> **Note**
>
> Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann entspricht der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

9. (Optional) Erstellen Sie im Abschnitt Approval Rules for applications (Genehmigungsregeln für Anwendungen) unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

> **Note**
>
> Anstatt Genehmigungsregeln anzugeben, können Sie Listen genehmigter und abgelehnter Patches als Patch-Ausnahmen angeben. Siehe Schritte 10 und 11.

- Product family (Produktfamilie): Die allgemeine Microsoft-Produktfamilie, für die Sie eine Regel festlegen möchten, z. B. `Office` oder `Exchange Server`.
- Produkte: Die Version der Anwendung, auf die sich die Genehmigungsregel bezieht, z. B. `Office 2016` oder `Active Directory Rights Management Services Client 2.0 2016`. Die Standardauswahl ist `All`.
- Classification (Klassifikation): Der Typ der Patches, auf die sich die Genehmigungsregel bezieht, z. B. `CriticalUpdates`. Die Standardauswahl ist `All`.
- Severity (Schweregrad): Der Schweregradwert von Patches, auf den die Regel anzuwenden ist, z. B. `Critical`. Die Standardauswahl ist `All`.
- Auto-approval (Automatische Genehmigung): Die Methode zum Auswählen von Patches für die automatische Genehmigung.
  - Approve patches after a specified number of days (Patches nach einer bestimmten Anzahl von Tagen genehmigen): Die Anzahl der Tage, die der Patch Manager warten muss, nachdem ein Patch veröffentlicht oder aktualisiert wurde, bevor ein Patch automatisch genehmigt wird. Sie können jede Ganzzahl von Null (0) bis 360 eingeben. Für die meisten Szenarien empfehlen wir, nicht länger als 100 Tage zu warten.
  - Approve patches released up to a specific date (Patches genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden): Das Datum der Patch-Veröffentlichung, an dem der Patch Manager automatisch alle Patches anwendet, die bis zu diesem Datum veröffentlicht oder aktualisiert wurden. Wenn Sie beispielsweise den 07. Juli 2023 angeben, werden Patches, die am oder nach dem 08. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden, nicht automatisch installiert.
- (Optional) Konformitätsbericht : Der Schweregrad, den Sie Patches zuweisen möchten, die von der Baseline genehmigt wurden (z. B. `Critical` oder `High`).

> **Note**
>
> Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann entspricht der insgesamt

gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

10. (Optional) Wenn Sie Patches explizit genehmigen möchten, anstatt Patches gemäß Genehmigungsregeln auszuwählen, gehen Sie im Abschnitt Patch-Ausnahmen folgendermaßen vor:

    - Geben Sie im Feld Approved patches (Genehmigte Patches) eine durch Komma getrennte Liste der Patches ein, die Sie genehmigen möchten.

      > **Note**
      >
      > Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

    - (Optional) Weisen Sie in der Liste Approved patches compliance level (Compliance-Stufe genehmigter Patches) den Patches in der Liste eine Compliance-Stufe zu.

11. Wenn Sie Patches ablehnen möchten, die ansonsten Ihren Genehmigungsregeln entsprechen, gehen Sie im Abschnitt Patch exceptions (Patch-Ausnahmen) wie folgt vor:

    - Geben Sie im Feld Rejected patches (Abgelehnte Patches) eine durch Komma getrennte Liste der Patches ein, die Sie ablehnen möchten.

      > **Note**
      >
      > Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

    - Wählen Sie in der Liste Rejected patches action (Aktion für abgelehnte Patches) die Aktion aus, die Patch Manager für Patches in der Liste Rejected patches (Abgelehnte Patches) ausführen soll.
      - Allow as dependency (Als Abhängigkeit zulassen): Ein Paket in der Liste Rejected patches (Abgelehnte Patches) wird nur installiert, wenn es sich um eine Abhängigkeit eines anderen Pakets handelt. Es gilt als konform mit der Patch-Baseline und sein Status wird als gemeldet InstalledOther. Dies ist die Standardaktion, wenn keine Option ausgewählt ist.

- Blockieren: Pakete in der Liste der abgelehnten Patches sowie Pakete, die diese als Abhängigkeiten enthalten, werden Patch Manager unter keinen Umständen installiert. Wenn ein Paket installiert wurde, bevor es zur Liste der abgelehnten Patches hinzugefügt wurde, oder wenn es außerhalb oder Patch Manager danach installiert wird, gilt es als nicht konform mit der Patch-Baseline und sein Status wird als gemeldet. InstalledRejected

12. (Optional) Wählen Sie für Manage tags (Tags verwalten) ein oder mehrere Tag-Schlüsselname/Wertpaare für die Patch-Baseline aus.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise eine Patch-Baseline kennzeichnen, um den Schweregrad der angegebenen Patches, die Betriebssystemfamilie, auf die sie sich bezieht, und den Umgebungstyp zu identifizieren. In diesem Fall können Sie etwa Tags mit den folgenden Schlüsselnamen/Wertpaaren angeben:

    - `Key=PatchSeverity,Value=Critical`
    - `Key=OS,Value=RHEL`
    - `Key=Environment,Value=Production`

13. Wählen Sie die Option Create Patch Baseline.

Aktualisieren oder Löschen einer benutzerdefinierten Patch-Baseline

Sie können eine benutzerdefinierte Patch-Baseline, die Sie in erstellt haben, aktualisieren oder löschenPatch Manager, eine Funktion von AWS Systems Manager. Wenn Sie eine Patch-Baseline aktualisieren, können Sie deren Namen oder Beschreibung, die Genehmigungsregeln sowie die Ausnahmen für genehmigte und abgelehnte Patches ändern. Sie können auch die Tags aktualisieren, die auf die Patch-Baseline angewendet werden. Sie können den Betriebssystemtyp, für den eine Patch-Baseline erstellt wurde, nicht ändern, und Sie können keine Änderungen an einer vordefinierten Patch-Baseline vornehmen, die von bereitgestellt wird AWS.

Aktualisieren oder Löschen einer Patch-Baseline

Gehen Sie wie folgt vor, um eine Patch-Baseline zu aktualisieren oder zu löschen.

> **Important**
>
> Gehen Sie vorsichtig vor, wenn Sie eine benutzerdefinierte Patch-Baseline löschen, die möglicherweise von einer Patch-Richtlinienkonfiguration in Quick Setup verwendet wird.

Wenn Sie eine Patch-Richtlinienkonfiguration in Quick Setup verwenden, werden Aktualisierungen, die Sie an benutzerdefinierten Patch-Baselines vornehmen, einmal pro Stunde mit Quick Setup synchronisiert.
Wenn eine benutzerdefinierte Patch-Baseline gelöscht wird, auf die in einer Patch-Richtlinie verwiesen wurde, wird auf der Seite mit den Quick Setup-Configuration details (Konfigurationsdetails) ein Banner für Ihre Patch-Richtlinie angezeigt. Das Banner informiert Sie darüber, dass die Patch-Richtlinie auf eine nicht mehr vorhandene Patch-Baseline verweist und nachfolgende Patching-Vorgänge fehlschlagen werden. Kehren Sie in diesem Fall zur Seite Quick Setup-Configurations (Konfigurationen) zurück, wählen Sie die Patch Manager-Konfiguration aus und wählen Sie Actions (Aktionen), Edit configuration (Konfiguration bearbeiten). Der Name der gelöschten Patch-Baseline wird hervorgehoben, und Sie müssen eine neue Patch-Baseline für das betroffene Betriebssystem auswählen.

So aktualisieren oder löschen Sie eine Patch-Baseline

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Patch-Baseline aus, die Sie aktualisieren oder löschen möchten, und führen Sie dann einen der folgenden Schritte aus:

   - Um die Patch-Baseline aus Ihrem zu entfernen AWS-Konto, wählen Sie Löschen. Sie werden aufgefordert, Ihre Aktionen zu bestätigen.
   - Wenn Sie den Namen oder die Beschreibung, die Genehmigungsregeln oder Patch-Ausnahmen der Patch-Baseline ändern möchten, wählen Sie Edit (Bearbeiten) aus. Nehmen Sie auf der Seite Edit patch baseline (Patch-Baseline bearbeiten) die gewünschten Änderungen vor und klicken Sie dann auf Save changes (Änderungen speichern).
   - Wenn Sie auf die Patch-Baseline angewendete Tags hinzufügen, ändern oder löschen möchten, klicken Sie auf die Registerkarte Tags (Tags) und dann auf Edit tags (Tags bearbeiten). Nehmen Sie auf der Seite Edit patch baseline tags (Patch-Baseline-Tags bearbeiten) die gewünschten Änderungen vor und klicken Sie dann auf Save changes (Änderungen speichern).

   Weitere Informationen zu den Konfigurationsoptionen, die Sie ausführen können, finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines.

Festlegen einer vorhandenen Patch-Baseline als Standard

> **Important**
>
> Alle hier getroffenen Standardauswahlen für die Patch-Baseline gelten nicht für Patching-Vorgänge, die auf einer Patch-Richtlinie basieren. Patch-Richtlinien verwenden ihre eigenen Patch-Baseline-Spezifikationen. Weitere Informationen zu Patch-Richtlinien finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien.

Bereits beim Erstellen einer benutzerdefinierten Patch-Baseline in Patch Manager, einer Funktion von AWS Systems Manager, können Sie die Baseline als Standard für den zugehörigen Betriebssystemtyp festlegen. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines.

Sie können auch eine vorhandene Patch-Baseline als Standard für einen Betriebssystemtyp festlegen.

> **Note**
>
> Welche Schritte Sie ausführen, hängt davon ab, ob Sie vor oder nach der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 auf Patch Manager zugegriffen haben. Wenn Sie Patch Manager vor diesem Datum verwendet haben, können Sie das Konsolenverfahren verwenden. Verwenden Sie andernfalls das AWS CLI Verfahren. Das Menü Aktionen, auf das im Konsolenverfahren verwiesen wird, wird in Regionen, in denen Patch Manager vor der Veröffentlichung der Patch-Richtlinien nicht verwendet wurde, nicht angezeigt.

So legen Sie eine Patch-Baseline als Standard fest

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patch-Baselines aus.
4. Wählen Sie in der Liste der Patch-Baselines die Schaltfläche einer Patch-Baseline aus, die derzeit nicht als Standard für ein Betriebssystem festgelegt ist.

Die Spalte Default baseline (Standard-Baseline) gibt an, welche Baselines derzeit als Standardwerte festgelegt sind.

5. Wählen Sie im Menü Actions (Aktionen) die Option Set default patch baseline (Standard-Patch-Baseline festlegen) aus.

> **Important**
>
> Das Aktionsmenü ist nicht verfügbar, wenn Sie nicht vor dem 22. Dezember 2022 mit Patch Manager in der aktuellen Version AWS-Konto und in der Region gearbeitet haben. Weitere Informationen finden Sie in der Anmerkung weiter oben in diesem Thema.

6. Wählen Sie im Bestätigungsdialogfeld Set default (Als Standard festlegen) aus.

So legen Sie eine Patch-Baseline als Standard fest (AWS CLI)

1. Führen Sie den Befehl describe-patch-baselines aus, um eine Liste der verfügbaren Patch-Baselines und ihrer IDs und Amazon-Ressourcennamen (ARNs) anzuzeigen.

```
aws ssm describe-patch-baselines
```

2. Führen Sie den Befehl register-default-patch-baseline aus, um eine Baseline als Standard für das Betriebssystem festzulegen, mit dem sie verknüpft ist. Ersetzen Sie *baseline-id-or-ARN* durch die ID der zu verwendenden benutzerdefinierten Patch-Baseline oder vordefinierten Baseline.

   Linux & macOS

```
aws ssm register-default-patch-baseline \
    --baseline-id baseline-id-or-ARN
```

   Im Folgenden finden Sie ein Beispiel für die Festlegung einer benutzerdefinierten Baseline als Standard.

```
aws ssm register-default-patch-baseline \
    --baseline-id pb-abc123cf9bEXAMPLE
```

Im Folgenden finden Sie ein Beispiel für die Einstellung einer vordefinierten Baseline, die AWS standardmäßig verwaltet wird.

```
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
```

Im Folgenden finden Sie ein Beispiel für die Festlegung einer benutzerdefinierten Baseline als Standard.

```
aws ssm register-default-patch-baseline ^
    --baseline-id pb-abc123cf9bEXAMPLE
```

Im Folgenden finden Sie ein Beispiel für die Einstellung einer vordefinierten Baseline, die AWS standardmäßig verwaltet wird.

```
aws ssm register-default-patch-baseline ^
    --baseline-id arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-071da192df1226b63
```

Anzeigen verfügbarer Patches

Mit Patch Manager einer Funktion von AWS Systems Manager können Sie alle verfügbaren Patches für ein bestimmtes Betriebssystem und optional für eine bestimmte Betriebssystemversion anzeigen.

> **Tip**
>
> Um eine Liste verfügbarer Patches zu generieren und diese in einer Datei zu speichern, können Sie den Befehl describe-available-patches verwenden und Ihre bevorzugte Ausgabe angeben.

Anzeigen verfügbarer Patches

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patches aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen und dann die Registerkarte Patches aus.

   > **Note**
   >
   > Für Windows Server zeigt die Registerkarte Patches Updates an, die vom Windows Server Update Service (WSUS) verfügbar sind.

4. Für Betriebssystem wählen Sie das Betriebssystem aus, für das Sie verfügbare Patches anzeigen möchten, z. B. `Windows` oder `Amazon Linux`.
5. (Optional) Für Product (Produkt) wählen Sie eine Betriebssystemversion aus, z. B. `WindowsServer2019` oder `AmazonLinux2018.03`.
6. (Optional) Um Informationsspalten für Ihre Ergebnisse hinzuzufügen oder zu entfernen, wählen Sie die Konfigurationsschaltfläche () oben rechts in der Liste Patches aus. (Standardmäßig zeigt die Registerkarte Patches nur Spalten für einige der verfügbaren Patch-Metadaten an.)

   Informationen zu den Arten von Metadaten, die Sie Ihrer Ansicht hinzufügen können, finden Sie unter Patch in der AWS Systems Manager -API-Referenz.

Arbeiten mit Patch-Gruppen

Wenn Sie in Ihrem Betrieb keine Patching-Richtlinien verwenden, können Sie Ihre Patching-Aufgaben organisieren, indem Sie verwaltete Knoten mithilfe von Tags zu Patch-Gruppen hinzufügen.

> **Important**
>
> Patch-Gruppen werden nicht in Patch-Vorgängen verwendet, die auf Patch-Richtlinien basieren. Weitere Informationen zur Arbeit mit Patch-Richtlinien finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien.

Um Tags bei Patching-Operationen zu verwenden, müssen Sie den Tag-Schlüssel `Patch Group` oder `PatchGroup` auf Ihre verwalteten Knoten anwenden. Sie müssen auch den Namen, den Sie der Patch-Gruppe geben möchten, als Wert des Tags angeben. Sie können einen beliebigen Tag-Wert angeben, aber der Tag-Schlüssel muss `Patch Group` oder `PatchGroup` lauten.

`PatchGroup` (ohne Leerzeichen) ist erforderlich, wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben.

Nachdem Sie Ihre verwalteten Knoten mithilfe von Tags gruppiert haben, fügen Sie den Patch-Gruppenwert einer Patch-Baseline hinzu. Mit der Registrierung der Patch-Gruppe für eine Patch-Baseline können Sie sicherstellen, dass beim Einspielen von Patches die richtigen Patches installiert werden. Weitere Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

Führen Sie die Aufgaben in diesem Thema aus, um Ihre verwalteten Knoten für das Patching vorzubereiten, indem Sie Tags mit Ihren Knoten und der Patch-Baseline verwenden. Aufgabe 1 ist nur erforderlich, wenn Sie Amazon-EC2-Instances patchen. Aufgabe 2 ist nur erforderlich, wenn Sie Nicht-EC2-Instances in einer Hybrid- und Multi-Cloud-Umgebung patchen. Aufgabe 3 ist für alle verwalteten Knoten erforderlich.

> **Tip**
>
> Sie können verwalteten Knoten auch mithilfe des AWS CLI Befehls `add-tags-to-resource` oder der Systems Manager Manager-API-Operation Tags hinzufügen`AddTagsToResource`.

Aufgaben

- Aufgabe 1: Hinzufügen von EC2-Instances zu einer Patch-Gruppe mithilfe von Tags
- Aufgabe 2: Hinzufügen von verwalteten Knoten zu einer Patch-Gruppe mithilfe von Tags
- Aufgabe 3: Hinzufügen einer Patch-Gruppe zu einer Patch-Baseline

Aufgabe 1: Hinzufügen von EC2-Instances zu einer Patch-Gruppe mithilfe von Tags

Sie können EC-Instances über die Systems-Manager-Konsole oder die Amazon-EC2-Konsole Tags hinzufügen. Diese Aufgabe ist nur erforderlich, wenn Sie Amazon-EC2-Instances patchen.

> **Important**
>
> Sie können das `Patch Group`-Tag (mit einem Leerzeichen) nicht auf eine Amazon-EC2-Instance anwenden, wenn die Option Allow tags in instance metadata (Tags in Instance-Metadaten zulassen) auf der Instance aktiviert ist. Durch das Zulassen von Tags in Instance-Metadaten wird verhindert, dass Tag-Schlüsselnamen Leerzeichen enthalten. Wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben, müssen Sie den Tag-Schlüssel `PatchGroup` (ohne Leerzeichen) verwenden.

Option 1: So fügen Sie EC2-Instances zu einer Patch-Gruppe hinzu (Systems-Manager-Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie in der Liste Verwaltete Knoten die ID einer verwalteten EC2-Instance, die Sie für das Patching konfigurieren möchten. Knoten-IDs für EC2-Instances beginnen mit `i-`.

   > **Note**
   >
   > Wenn Sie die Amazon EC2 EC2-Konsole und verwenden AWS CLI, ist es möglich, `Key = PatchGroup` Or-Tags auf Instances anzuwendenKey `= Patch Group`, die noch nicht für die Verwendung mit Systems Manager konfiguriert sind.
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

4. Wählen Sie die Registerkarte Tags und dann Bearbeiten aus.
5. Geben Sie in der linken Spalte **`Patch Group`** oder **`PatchGroup`** ein. Wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben, müssen Sie `PatchGroup` (ohne Leerzeichen) verwenden.

6. Geben Sie in der rechten Spalte einen Tag-Wert ein, der als Name für die Patch-Gruppe dienen soll.
7. Wählen Sie Speichern.
8. Wiederholen Sie dieses Verfahren, um andere EC2-Instances zur selben Patch-Gruppe hinzuzufügen.

Option 2: So fügen Sie EC2-Instances einer Patch-Gruppe hinzu (Amazon EC2-Konsole)

1. Öffnen Sie im Navigationsbereich die Amazon EC2-Konsole und wählen Sie die Option Instances aus.
2. Wählen Sie in der Liste der Instances eine Instance aus, die Sie für das Einspielen von Patches konfigurieren möchten.
3. Wählen Sie im Menü Aktionen die Option Instance-Einstellungen, Tags verwalten aus.
4. Wählen Sie Neues Tag hinzufügen aus.
5. Geben Sie für Key (Schlüssel) **`Patch Group`** oder **`PatchGroup`** ein. Wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben, müssen Sie `PatchGroup` (ohne Leerzeichen) verwenden.
6. Geben Sie für Wert einen Wert ein, der als Name für die Patch-Gruppe dienen soll.
7. Wählen Sie Speichern.
8. Wiederholen Sie dieses Verfahren, um andere Instances zur selben Patch-Gruppe hinzuzufügen.

Aufgabe 2: Hinzufügen von verwalteten Knoten zu einer Patch-Gruppe mithilfe von Tags

Folgen Sie den Schritten in diesem Thema, um Tags zu AWS IoT Greengrass Kerngeräten und verwalteten Knoten (mi-*) ohne EC2-Hybrid-Aktivierung hinzuzufügen. Diese Aufgabe ist nur erforderlich, wenn Sie Nicht-EC2 Instances in einer Hybrid- und Multi-Cloud-Umgebung patchen.

> **Note**
>
> Sie können über die Amazon-EC2-Konsole keine Tags für Nicht-EC2-verwaltete Knoten hinzufügen.

So fügen Sie Nicht-EC2-verwaltete Knoten einer Patch-Gruppe hinzu (Systems-Manager-Konsole)

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/systems-manager/. AWS Systems Manager
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie in der Liste der verwalteten Knoten einen verwalteten Knoten, für den Sie das Patching konfigurieren möchten.

   > **Note**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

4. Wählen Sie die Registerkarte Tags und dann Bearbeiten aus.
5. Geben Sie in der linken Spalte **`Patch Group`** oder **`PatchGroup`** ein. Wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben, müssen Sie `PatchGroup` (ohne Leerzeichen) verwenden.
6. Geben Sie in der rechten Spalte einen Tag-Wert ein, der als Name für die Patch-Gruppe dienen soll.
7. Wählen Sie Speichern.
8. Wiederholen Sie dieses Verfahren, um andere verwaltete Knoten zur selben Patch-Gruppe hinzuzufügen.

Aufgabe 3: Hinzufügen einer Patch-Gruppe zu einer Patch-Baseline

Um Ihren verwalteten Knoten eine bestimmte Patch-Baseline zuzuordnen, müssen Sie den Patch-Gruppenwert der Patch-Baseline hinzufügen. Mit der Registrierung der Patch-Gruppe für eine Patch-Baseline können Sie sicherstellen, dass beim Einspielen von Patches die richtigen Patches installiert werden. Diese Aufgabe ist unabhängig davon erforderlich, ob Sie EC2-Instances, verwaltete Nicht-EC2-Knoten oder beides patchen.

Weitere Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

> **Note**
>
> Welche Schritte Sie ausführen, hängt davon ab, ob Sie vor oder nach der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben.

So fügen Sie eine Patch-Gruppe einer Patch-Baseline hinzu (Systems-Manager-Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wenn Sie auf Patch Manager zum ersten Mal in der aktuellen AWS-Region zugreifen und sich die Patch Manager-Startseite öffnet, wählen Sie Mit einer Übersicht beginnen.
4. Wählen Sie die Registerkarte Patch-Baselines und wählen Sie dann in der Liste Patch-Baselines den Namen der Patch-Baseline, die Sie für Ihre Patch-Gruppe konfigurieren möchten.

   Wenn Sie erst nach der Veröffentlichung der Patch-Richtlinien auf Patch Manager zugegriffen haben, müssen Sie eine von Ihnen erstellte benutzerdefinierte Baseline wählen.
5. Wenn die Detailseite der Baseline-ID ein Menü Aktionen enthält, gehen Sie wie folgt vor:

   - Wählen Sie Actions (Aktionen) und dann Modify patch groups (Patch-Gruppen modifizieren) aus.
   - Geben Sie den Tag-Wert, den Sie Ihren verwalteten Knoten in Aufgabe 2: Hinzufügen von verwalteten Knoten zu einer Patch-Gruppe mithilfe von Tags hinzugefügt haben, und wählen Sie dann Hinzufügen.

   Wenn die Detailseite der Baseline-ID kein Menü Aktionen enthält, können Patch-Gruppen in der Konsole nicht konfiguriert werden. Sie können stattdessen eine der folgenden Aktionen ausführen:

   - (Empfohlen) Richten Sie eine Patch-Richtlinie mit der Fähigkeit von ein Quick Setup AWS Systems Manager, um eine Patch-Baseline einer oder mehreren EC2-Instances zuzuordnen.

     Weitere Informationen finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien und Automatisieren des unternehmensweiten Patchen mithilfe einer Quick Setup-Patch-Richtlinie.

- Verwenden Sie den register-patch-baseline-for-patch-groupBefehl in AWS Command Line Interface (AWS CLI), um eine Patchgruppe zu konfigurieren.

Arbeiten mit Patch Manager-Einstellungen

Themen

- Integrieren Patch Manager mit AWS Security Hub

Integrieren Patch Manager mit AWS Security Hub

AWS Security Hubbietet Ihnen einen umfassenden Überblick über Ihren Sicherheitsstatus in AWS. Security Hub sammelt Sicherheitsdaten von verschiedenen AWS-Konten und unterstützten Partnerprodukten von Drittanbietern. AWS-Services Mit Security Hub können Sie sich Ihren Sicherheitsstatus ansehen und Ihre Umgebung anhand der Standards und bewährten Methoden der Sicherheitsbranche überprüfen. Security Hub hilft Ihnen dabei, Ihre Sicherheitstrends zu analysieren und Sicherheitsprobleme mit höchster Priorität zu identifizieren.

Mithilfe der Integration zwischenPatch Manager, einer Funktion von AWS Systems Manager und Security Hub können Sie Erkenntnisse über nicht konforme Knoten von Patch Manager an Security Hub senden. Ein Ergebnis ist der beobachtbare Datensatz einer Sicherheitsprüfung oder sicherheitsrelevanten Erkennung. Security Hub kann diese Patch-bezogenen Ergebnisse dann in die Analyse Ihres Sicherheitsstatus einbeziehen.

Die Informationen in den folgenden Themen gelten unabhängig davon, welche Methode oder Art der Konfiguration Sie für Ihre Patching-Vorgänge verwenden:

- Eine in Quick Setup konfigurierte Patch-Richtlinie
- Eine in Quick Setup konfigurierte Host-Management-Option
- Ein Wartungsfenster zum Ausführen eines Patch-`Scans` oder einer `Install`-Aufgabe
- Eine On-Demand Patch now-Operation (Jetzt patchen)

Inhalt

- So sendet Patch Manager Erkenntnisse an Security Hub
  - Arten von Erkenntnissen, die Patch Manager sendet
  - Latenz für das Senden von Erkenntnissen
  - Wiederholen, wenn Security Hub nicht verfügbar ist

  - Ergebnisse im Security Hub anzeigen
- Typische Erkenntnis von Patch Manager
- Aktivieren und Konfigurieren der Integration
- So beenden Sie das Senden von Ergebnissen

So sendet Patch Manager Erkenntnisse an Security Hub

Im Security Hub werden Sicherheitsprobleme als Erkenntnisse verfolgt. Einige Ergebnisse stammen aus Problemen, die von anderen AWS-Services oder von Drittanbietern entdeckt wurden. Security Hub verwendet ebenfalls verschiedene Regeln, um Sicherheitsprobleme zu erkennen und Ergebnisse zu generieren.

Patch Manager ist eine der Systems Manager-Funktionen, die Ergebnisse an den Security Hub sendet. Nachdem Sie einen Patchvorgang durchgeführt haben, indem Sie ein SSM-Dokument (`AWS-RunPatchBaseline`,`AWS-RunPatchBaselineAssociation`, oder`AWS-RunPatchBaselineWithHooks`) ausführen, werden die Patchinformationen an Inventar oder Compliance, Funktionen von AWS Systems Manager oder an beide gesendet. Nachdem Inventory, Compliance oder beide die Daten erhalten haben, erhält Patch Manager eine Benachrichtigung. Dann wertet Patch Manager die Daten auf Genauigkeit, Formatierung und Compliance aus. Wenn alle Bedingungen erfüllt sind, leitet Patch Manager die Daten an den Security Hub weiter.

Security Hub bietet Tools zur Verwaltung von Erkenntnissen aus all diesen Quellen. Sie können Listen mit Erkenntnissen anzeigen und filtern und Details zu einer Erkenntnis anzeigen. Weitere Informationen finden Sie unter Anzeigen der Erkenntnisse im AWS Security Hub -Benutzerhandbuch. Sie können auch den Status einer Untersuchung zu einer Erkenntnis nachverfolgen. Weitere Informationen finden Sie unter Ergreifen von Maßnahmen zu Erkenntnissen im AWS Security Hub -Benutzerhandbuch.

Alle Ergebnisse in Security Hub verwenden ein standardmäßiges JSON-Format, das AWS Security Finding Format (ASFF). Das ASFF enthält Details über die Ursache des Problems, die betroffenen Ressourcen und den aktuellen Status der Erkenntnis. Weitere Informationen finden Sie unter AWS - Security Finding-Format (ASFF) im AWS Security Hub -Benutzerhandbuch.

Arten von Erkenntnissen, die Patch Manager sendet

Patch Manager sendet die Ergebnisse unter Verwendung des AWS Security Finding Format (ASFF) an Security Hub. In ASFF gibt das `Types`-Feld die Art der Erkenntnis an. Die Ergebnisse von Patch Manager können den folgenden Wert für `Types` haben:

- Software- und Konfigurationsprüfungen/Patchverwaltung

Patch Manager sendet ein Ergebnis pro nicht konformen verwalteten Knoten. Das Ergebnis wird mit dem Ressourcentyp `AwsEc2Instance` gemeldet, damit die Ergebnisse mit anderen Security-Hub-Integrationen korreliert werden können, die `AwsEc2Instance`-Ressourcentypen melden. Patch Manager leitet ein Ergebnis nur dann an den Security Hub weiter, wenn die Operation festgestellt hat, dass der verwaltete Knoten nicht konform ist. Das Ergebnis enthält die Ergebnisse der Patch-Zusammenfassung.

> **Note**
>
> Nach dem Melden eines nicht konformen Knotens an Security Hub. Patch Managersendet kein Update an Security Hub, nachdem der Knoten konform gemacht wurde. Sie können die Ergebnisse in Security Hub manuell beheben, nachdem die erforderlichen Patches auf den verwalteten Knoten angewendet wurden.

Weitere Informationen zu Compliance-Definitionen finden Sie unter Grundlegendes zu Patch-Compliance-Statuswerten. Weitere Informationen zu `PatchSummary` finden Sie PatchSummaryin der AWS Security Hub API-Referenz.

Latenz für das Senden von Erkenntnissen

Wenn Patch Manager ein neues Ergebnis erstellt, wird es normalerweise innerhalb von wenigen Sekunden bis 2 Stunden an den Security Hub gesendet. Die Geschwindigkeit hängt vom Verkehr zu diesem Zeitpunkt in der AWS-Region verarbeiteten Verkehr ab.

Wiederholen, wenn Security Hub nicht verfügbar ist

Bei einem Dienstausfall wird eine AWS Lambda Funktion ausgeführt, um die Nachrichten wieder in die Hauptwarteschlange zu verschieben, nachdem der Dienst wieder ausgeführt wird. Nachdem sich die Nachrichten in der Hauptwarteschlange befinden, erfolgt die Wiederholung automatisch.

Wenn der Security Hub nicht verfügbar ist, versucht Patch Manager so lange erneut, die Ergebnisse zu senden, bis sie empfangen wurden.

Ergebnisse im Security Hub anzeigen

In diesem Verfahren wird beschrieben, wie Sie in Security Hub Erkenntnisse über verwaltete Knoten in Ihrer Flotte anzeigen können, bei denen die Patch-Konformität nicht gegeben ist.

Um die Ergebnisse von Security Hub auf Patch-Konformität zu überprüfen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Security Hub Konsole unter https://console.aws.amazon.com/securityhub/.
2. Wählen Sie im Navigationsbereich Findings aus.
3. Wählen Sie das Feld Filter hinzufügen ().
4. Wählen Sie im Menü unter Filter die Option Produktname aus.
5. Wählen Sie in dem sich öffnenden Dialogfeld im ersten Feld die Option ist und geben Sie dann **`Systems Manager Patch Manager`** im zweiten Feld ein.
6. Wählen Sie Apply (Anwenden) aus.
7. Fügen Sie weitere Filter hinzu, um Ihre Ergebnisse einzugrenzen.
8. Wählen Sie in der Ergebnisliste den Titel eines Erkenntnisses aus, zu dem Sie weitere Informationen wünschen.

   Auf der rechten Seite des Bildschirms wird ein Bereich mit weiteren Informationen zur Ressource, dem erkannten Problem und einer empfohlenen Lösung geöffnet.

   > **Important**
   >
   > Derzeit meldet Security Hub den Ressourcentyp aller verwalteten Knoten als `EC2 Instance`. Dazu gehören On-Premises-Server und virtuelle Maschinen (VMs), die Sie für die Verwendung mit Systems Manager registriert haben.

Schweregradklassifizierungen

Die Liste der Erkenntnisse für **`Systems Manager Patch Manager`** enthält einen Bericht über den Schweregrad des Befundes. Zu den Schweregraden gehören die folgenden, vom niedrigsten zum höchsten:

- INFORMATIV – Es wurde kein Problem gefunden.
- NIEDRIG — Das Problem muss nicht behoben werden.
- MITTEL – Das Problem muss angegangen werden, aber ist nicht dringend.
- HOCH – Das Problem muss vorrangig behandelt werden.
- KRITISCH – Das Problem muss sofort behoben werden, um eine Eskalation zu vermeiden.

Der Schweregrad wird durch das schwerwiegendste nicht konforme Paket auf einer Instance bestimmt. Da Sie mehrere Patch-Baselines mit verschiedenen Schweregraden haben können, wird der höchste Schweregrad von allen nicht konformen Paketen gemeldet. Nehmen wir zum Beispiel an, Sie haben zwei nicht konforme Pakete, wobei der Schweregrad von Paket A „Kritisch“ und der von Paket B „Gering“ ist. „Kritisch“ wird als Schweregrad angegeben werden.

Beachten Sie, dass das Schweregradfeld direkt mit dem Feld Patch Manager `Compliance` korreliert. Dies ist ein Feld, das Sie einzelnen Patches zuweisen, die der Regel entsprechen. Da dieses `Compliance`-Feld einzelnen Patches zugewiesen ist, wird es nicht auf der Ebene der Patch-Zusammenfassung wiedergegeben.

Verwandter Inhalt

- Erkenntnisse im AWS Security Hub -Benutzerhandbuch
- Multi-Account Patch-Compliance mit Patch Manager und Security Hub im AWS -Management & Governance Blog

Typische Erkenntnis von Patch Manager

Patch Manager sendet Ergebnisse unter Verwendung des AWS Security Finding Format (ASFF) an Security Hub.

Hier ist ein Beispiel für ein typisches Ergebnis von 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/managed-
instances/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"
  }
}
```

Aktivieren und Konfigurieren der Integration

Um die Patch Manager Integration mit Security Hub verwenden zu können, müssen Sie den Security Hub aktivieren. Informationen zur Aktivierung von Security Hub finden Sie unter Einrichten von Security Hub im AWS Security Hub -Benutzerhandbuch.

Im folgenden Verfahren wird beschrieben, wie Sie Patch Manager und Security Hub integrieren, wenn Security Hub bereits aktiv ist, die Patch Manager-Integration aber deaktiviert ist. Sie müssen diesen Vorgang nur abschließen, wenn die Integration manuell deaktiviert wurde.

So fügen Sie Patch Manager der Security Hub-Integration hinzu

1. Wählen Sie im Navigationsbereich Patch Manager aus.
2. Wählen Sie die Registerkarte Settings.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen und dann die Registerkarte Einstellungen aus.
3. Wählen Sie im Abschnitt Exportieren in Security Hub rechts neben Patch-Compliance-Ergebnisse werden nicht in den Security Hub exportiert Aktivieren aus.

So beenden Sie das Senden von Ergebnissen

Um keine Ergebnisse mehr an Security Hub zu senden, können Sie entweder die Security Hub-Konsole oder die API verwenden.

Weitere Informationen finden Sie in folgenden Themen im AWS Security Hub -Benutzerhandbuch:

- Deaktivieren und Aktivieren des Flows von Ergebnissen aus einer Integration (Konsole)
- Den Fluss von Erkenntnissen aus einer Integration deaktivieren (Security Hub API, AWS CLI)

Arbeiten mit Patch Manager (AWS CLI)

Der Abschnitt enthält Beispiele für AWS Command Line Interface (AWS CLI)-Befehle, mit denen Sie Konfigurationsaufgaben für Patch Manager, eine Funktion von AWS Systems Manager, ausführen können.

Ein Beispiel für die Verwendung der AWS CLI zum Patchen einer Serverumgebung mittels einer benutzerdefinierten Patch-Baseline finden Sie unter Anleitung: Patchen einer Serverumgebung (AWS CLI).

Weitere Informationen zur Verwendung der AWS CLI für AWS Systems Manager-Aufgaben finden Sie im AWS Systems Manager-Abschnitt der AWS CLI-Befehlsreferenz.

Themen

- AWS CLI-Befehle für Patch-Baselines
- AWS CLI-Befehle für Patch-Gruppen
- AWS CLI-Befehle zum Anzeigen von Patch-Zusammenfassungen und -details
- AWS CLI-Befehle zum Scannen und Patchen von verwalteten Knoten

AWS CLI-Befehle für Patch-Baselines

Beispielbefehle für Patch-Baselines

- Erstellen einer Patch-Baseline
- Erstellen einer Patch-Baseline mit benutzerdefinierten Repositorys für verschiedene Betriebssystemversionen
- Aktualisieren einer Patch-Baseline
- Umbenennen einer Patch-Baseline
- Löschen einer Patch-Baseline
- Auflisten aller Patch-Baselines
- Auflisten aller von AWS bereitgestellten Patch-Baselines
- Auflisten der eigenen Patch-Baselines
- Anzeigen einer Patch-Baseline
- Abrufen einer Standard-Patch-Baseline
- Eine benutzerdefinierte Patch-Baseline als Standard festlegen

- Eine AWS-Patch-Baseline als Standard zurücksetzen
- Markieren einer Patch-Baseline
- Auflisten aller Tags für eine Patch-Baseline
- Entfernen eines Tags aus einer Patch-Baseline

Erstellen einer Patch-Baseline

Der folgende Befehl erstellt eine Patch-Baseline, die alle kritischen und wichtigen Sicherheitsaktualisierungen für Windows Server 2012 R2 fünf Tage nach ihrer Veröffentlichung genehmigt. Patches wurden auch für die Listen „Genehmigt“ und „Zurückgewiesen“ angegeben. Darüber hinaus wurde die Patch-Baseline mit Tags versehen, um sie für die Produktionsumgebung freizugeben.

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}]"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Erstellen einer Patch-Baseline mit benutzerdefinierten Repositorys für verschiedene Betriebssystemversionen

Gilt nur für Linux-verwaltete Knoten. Der folgende Befehl zeigt, wie das Patch-Repository für eine bestimmte Version des Amazon Linux-Betriebssystems anzugeben ist. Dieses Beispiel verwendet ein standardmäßig aktiviertes Quell-Repository auf Amazon Linux 2017.09, könnte aber auf ein anderes Quell-Repository angepasst werden, das Sie für einen verwalteten Knoten konfiguriert haben.

> **Note**
>
> Um diesen komplexeren Befehl besser zu erklären, wird die Option `--cli-input-json` mit zusätzlichen, in einer externen JSON-Datei gespeicherten Optionen verwendet.

1. Erstellen Sie eine JSON-Datei mit einem Namen wie `my-patch-repository.json` und fügen Sie den folgenden Inhalt hinzu.

```
{
    "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. Führen Sie im Verzeichnis, in dem Sie die Datei gespeichert haben, den folgenden Befehl aus.

```
aws ssm create-patch-baseline --cli-input-json file://my-patch-repository.json
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "BaselineId": "pb-0c10e65780EXAMPLE"
}
```

Aktualisieren einer Patch-Baseline

Mit dem folgenden Befehl werden zwei Patches abgelehnt und ein weiterer Patch für eine vorhandenen Patch-Baseline genehmigt.

> **Note**
>
> Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für Listen genehmigter und abgelehnter Patches.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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"
}
```

Umbenennen einer Patch-Baseline

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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"
}
```

Löschen einer Patch-Baseline

```
aws ssm delete-patch-baseline --baseline-id "pb-0c10e65780EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Auflisten aller Patch-Baselines

```
aws ssm describe-patch-baselines
```

Das System gibt unter anderem folgende Informationen zurück

```
{
```

```
   "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"
      }
   ]
}
```

Nachstehend finden Sie einen weiteren Befehl zur Auflistung aller Patch-Baselines in einer AWS-Region.

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]"
```

Das System gibt unter anderem folgende Informationen zurück

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

Auflisten aller von AWS bereitgestellten Patch-Baselines

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]"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"AWS-DefaultPatchBaseline",
         "DefaultBaseline":true,
         "BaselineDescription":"Default Patch Baseline Provided by AWS.",
         "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
      }
   ]
}
```

Auflisten der eigenen Patch-Baselines

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]"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"Windows-Server-2012R2",
         "DefaultBaseline":false,
         "BaselineDescription":"Windows Server 2012 R2, Important and Critical security
 updates",
         "BaselineId":"pb-0c10e65780EXAMPLE"
      }
   ]
}
```

Anzeigen einer Patch-Baseline

```
aws ssm get-patch-baseline --baseline-id pb-0c10e65780EXAMPLE
```

> **Note**
>
> Bei benutzerdefinierten Patch-Baselines können Sie entweder die Patch-Baseline-ID oder den vollständigen Amazon-Ressourcennamen (ARN) angeben. Für von AWS bereitgestellte Patch-Baselines müssen Sie den vollständigen ARN angeben. Zum Beispiel `arn:aws:ssm:us-east-2:075727635805:patchbaseline/pb-0c10e65780EXAMPLE`.

Das System gibt unter anderem folgende Informationen zurück


```
{
   "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"
}
```

Abrufen einer Standard-Patch-Baseline

```
aws ssm get-default-patch-baseline --region us-east-2
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/pb-0c10e65780EXAMPLE"
}
```

Eine benutzerdefinierte Patch-Baseline als Standard festlegen

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Eine AWS-Patch-Baseline als Standard zurücksetzen

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Markieren einer Patch-Baseline

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"
```

Auflisten aller Tags für eine Patch-Baseline

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"
```

Entfernen eines Tags aus einer Patch-Baseline

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"
```

AWS CLI-Befehle für Patch-Gruppen

Beispielbefehle für Patch-Gruppen

- Erstellen einer Patch-Gruppe
- Registrieren einer Patch-Gruppe „Webserver“ für eine Patch-Baseline
- Registrieren einer Patch-Gruppe „Backend“ für die von AWS bereitgestellte Patch-Baseline
- Anzeigen der Registrierungen für Patch-Gruppen
- Aufheben der Registrierung einer Patch-Gruppe für eine Patch-Baseline

Erstellen einer Patch-Gruppe

Um das Organisieren Ihrer Patching-Aufgaben zu erleichtern, empfehlen wir, dass Sie verwaltete Knoten mithilfe von Tags zu Patch-Gruppen hinzufügen. Patch-Gruppen erfordern die Nutzung des Tag-Schlüssels `Patch Group` oder `PatchGroup`. Wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben, müssen Sie `PatchGroup` (ohne Leerzeichen) verwenden. Sie können einen beliebigen Tag-Wert angeben, aber der Tag-Schlüssel muss `Patch Group` oder `PatchGroup` lauten. Weitere Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

Nachdem Sie Ihre verwalteten Knoten mithilfe von Tags gruppiert haben, fügen Sie den Patch-Gruppenwert einer Patch-Baseline hinzu. Mit der Registrierung der Patch-Gruppe für eine Patch-Baseline können Sie sicherstellen, dass beim Einspielen von Patches die richtigen Patches installiert werden.

Aufgabe 1: Hinzufügen von EC2-Instances zu einer Patch-Gruppe mithilfe von Tags

> **Note**
>
> Bei Verwendung der Amazon Elastic Compute Cloud (Amazon EC2)-Konsole und AWS CLI ist es möglich, `Key = Patch Group`-oder `Key = PatchGroup`-Tags auf Instances anzuwenden, die noch nicht für die Verwendung mit Systems Manager konfiguriert sind. Wenn eine EC2-Instance, die Sie in Patch Manager erwarten, nach dem Anwenden des `Patch Group`-oder `Key = PatchGroup`-Tag nicht aufgeführt ist, finden Sie Tipps zur Fehlerbehebung unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten.

Führen Sie den folgenden Befehl aus, um das Tag `PatchGroup` einer EC2-Instance hinzuzufügen.

```
aws ec2 create-tags --resources "i-1234567890abcdef0" --tags
 "Key=PatchGroup,Value=GroupValue"
```

Aufgabe 2: Hinzufügen von verwalteten Knoten zu einer Patch-Gruppe mithilfe von Tags

Führen Sie den folgenden Befehl aus, um das Tag `PatchGroup` einem verwalteten Knoten hinzuzufügen.

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"
```

Aufgabe 3: Hinzufügen einer Patch-Gruppe zu einer Patch-Baseline

Führen Sie den folgenden Befehl aus, um der angegebenen Patch-Baseline einen `PatchGroup`-Tag-Wert zuzuordnen.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
  "PatchGroup": "Development",
  "BaselineId": "pb-0c10e65780EXAMPLE"
}
```

Registrieren einer Patch-Gruppe „Webserver“ für eine Patch-Baseline

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "PatchGroup":"Web Servers",
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Registrieren einer Patch-Gruppe „Backend“ für die von AWS bereitgestellte Patch-Baseline

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "PatchGroup":"Backend",
   "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/pb-0c10e65780EXAMPLE"
```

```
}
```

Anzeigen der Registrierungen für Patch-Gruppen

```
aws ssm describe-patch-groups --region us-east-2
```

Das System gibt unter anderem folgende Informationen zurück

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

Aufheben der Registrierung einer Patch-Gruppe für eine Patch-Baseline

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "PatchGroup":"Production",
   "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/pb-0c10e65780EXAMPLE"
}
```

AWS CLI-Befehle zum Anzeigen von Patch-Zusammenfassungen und -details

Beispielbefehle zum Anzeigen von Patch-Zusammenfassungen und -details

- Abrufen aller Patches, die in einer bestimmten Patch-Baseline definiert sind
- Alle Patches für AmazonLinux2018.03 mit der Klassifizierung SECURITY und einem Schweregrad von Critical erhalten
- Abrufen aller Patches für Windows Server 2012 mit einem MSRC-Schweregrad von Critical
- Abrufen aller verfügbaren Patches
- Abrufen der zusammengefassten Patch-Zustände pro verwalteten Knoten
- Abrufen der Patch-Compliance-Details für einen verwalteten Knoten
- Anzeigen der Patch-Compliance-Ergebnisse (AWS CLI)

Abrufen aller Patches, die in einer bestimmten Patch-Baseline definiert sind

> **Note**
>
> Dieser Befehl wird nur für Windows Server-Patch-Baselines unterstützt.

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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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---
```

Alle Patches für AmazonLinux2018.03 mit der Klassifizierung **`SECURITY`** und einem Schweregrad von **`Critical`** erhalten

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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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---
```

Abrufen aller Patches für Windows Server 2012 mit einem MSRC-Schweregrad von **`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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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---
```

Abrufen aller verfügbaren Patches

```
aws ssm describe-available-patches --region us-east-2
```

Das System gibt unter anderem folgende Informationen zurück


```
{
   "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---
```

Abrufen der zusammengefassten Patch-Zustände pro verwalteten Knoten

Diese Zusammenfassung pro verwalteten Knoten zeigt Ihnen die Anzahl der Patches mit den folgenden Zuständen pro Knoten an: „NotApplicable“, „Missing“, „Failed“, „InstalledOther“ und „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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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---
```

Abrufen der Patch-Compliance-Details für einen verwalteten Knoten

```
aws ssm describe-instance-patches --instance-id i-08ee91c0b17045407
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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---
```

Anzeigen der Patch-Compliance-Ergebnisse (AWS CLI)

Anzeigen von Patch-Compliance-Ergebnissen für einen einzelnen verwalteten Knoten

Führen Sie den folgenden Befehl in der AWS Command Line Interface (AWS CLI) aus, um die Patch-Compliance-Ergebnisse für einen einzelnen verwalteten Knoten anzuzeigen.

```
aws ssm describe-instance-patch-states --instance-id instance-id
```

Ersetzen Sie *instance-id* mit der ID des verwalteten Knoten, für den Sie Ergebnisse anzeigen möchten, im Format `i-02573cafcfEXAMPLE` oder `mi-0282f7c436EXAMPLE`.

Das System gibt unter anderem folgende Informationen zurück.

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

So zeigen Sie eine Patch-Anzahl-Zusammenfassung für alle EC2-Instances in einer Region an

Der `describe-instance-patch-states` unterstützt das Abrufen von Ergebnissen für jeweils eine verwaltete Instance. Wenn Sie jedoch ein benutzerdefiniertes Skript mit dem `describe-instance-patch-states`-Befehl verwenden, können Sie einen detaillierteren Bericht erstellen.

Wenn zum Beispiel das jq filter tool auf Ihrem lokalen Computer installiert ist, können Sie den folgenden Befehl ausführen, um zu ermitteln, welche Ihrer EC2-Instances in einer bestimmten AWS-Region einen Status von `InstalledPendingReboot` haben.

```
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}'
```

`region` repräsentiert die Kennung für eine von AWS-Region unterstützte AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`Region`*-Werte finden Sie in der Spalte Region unter Systems-Manager-Service-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

Beispiele:

```
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}'
```

Das System gibt unter anderem folgende Informationen zurück

```
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
```

Zusätzlich zu `InstalledPendingRebootCount` können Sie nach den folgenden Anzahltypen suchen:

- `CriticalNonCompliantCount`
- `SecurityNonCompliantCount`
- `OtherNonCompliantCount`
- `UnreportedNotApplicableCount`
- `InstalledPendingRebootCount`
- `FailedCount`
- `NotApplicableCount`
- `InstalledRejectedCount`
- `InstalledOtherCount`
- `MissingCount`
- `InstalledCount`

AWS CLI-Befehle zum Scannen und Patchen von verwalteten Knoten

Nachdem Sie die folgenden Befehle ausgeführt haben, um nach Patch-Compliance zu scannen oder Patches zu installieren, können Sie mit Befehlen im AWS CLI-Befehle zum Anzeigen von Patch-Zusammenfassungen und -details-Abschnitt Informationen zu Patch-Status und -Compliance anzeigen.

Beispielbefehle

- Verwaltete Knoten auf Patch-Compliance scannen (AWS CLI)

- Installieren von Patches auf verwalteten Knoten (AWS CLI)

Verwaltete Knoten auf Patch-Compliance scannen (AWS CLI)

So scannen Sie spezifische verwaltete Knoten auf Patch-Compliance

Führen Sie den folgenden Befehl aus.

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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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---

    }
}
```

So scannen Sie verwaltete Knoten nach Patch-Gruppentag auf Patch-Compliance

Führen Sie den folgenden Befehl aus.

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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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---

    }
}
```

Installieren von Patches auf verwalteten Knoten (AWS CLI)

So installieren Sie Patches auf spezifischen verwalteten Knoten

Führen Sie den folgenden Befehl aus.

> **Note**
>
> Die anvisierten verwalteten Knoten werden nach Bedarf neu gestartet, um die Patch-Installation abzuschließen. Weitere Informationen finden Sie unter Informationen über das `AWS-RunPatchBaseline` SSM-Dokument.

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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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---

    }
}
```

So installieren Sie Patches auf verwalteten Knoten in einer spezifischen Patch-Gruppe

Führen Sie den folgenden Befehl aus.

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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "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---

    }
}
```

AWS Systems ManagerPatch ManagerTutorials

Die Anleitungen in diesem Abschnitt zeigen die Verwendung von Patch Manager, einer Funktion von AWS Systems Manager, für verschiedene Patch-Szenarien.

Themen

- So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs (Konsole)
- Tutorial: Aktualisieren von Anwendungsabhängigkeiten, Patchen eines verwalteten Knotens und Durchführen einer anwendungsspezifischen Zustandsprüfung
- Anleitung: Patchen einer Serverumgebung (AWS CLI)

So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs (Konsole)

Wenn Sie eine benutzerdefinierte Patch-Baseline erstellen, können Sie angeben, ob alle, einige oder nur ein einziger unterstützter Patch-Typ installiert wird.

In den Patch-Baselines für Windows können Sie `ServicePacks` als einzige Klassifizierungsoption auswählen, um Patching-Updates auf Service Packs einzuschränken. Service Packs können mit einer Funktion von Patch Manager automatisch installiert werden AWS Systems Manager, sofern das Update in Windows Update oder Windows Server Update Services (WSUS) verfügbar ist.

Sie können eine Patch-Baseline konfigurieren, um festzulegen, ob Service Packs für alle Windows-Versionen oder nur für bestimmte Windows-Versionen installiert werden, z. B. Windows 7 oder Windows Server 2016.

Gehen Sie wie folgt vor, um eine benutzerdefinierte Patch-Baseline zu erstellen, die ausschließlich für die Installation aller Service Packs auf Ihren Windows-verwalteten Knoten verwendet wird.

So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patch-Baselines und dann Patch-Baseline erstellen aus.
4. Geben Sie im Feld Name (Name) einen Namen für die neue Patch-Baseline ein, z. B. `MyWindowsServicePackPatchBaseline`.
5. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für diese Patch-Baseline ein.
6. Wählen Sie unter Operating system (Betriebssystem) die Option `Windows` aus.
7. Wenn Sie diese Patch-Baseline direkt nach dem Erstellen als Standard für Windows verwenden möchten, wählen Sie Set this patch baseline as the default patch baseline for Windows Server instances (Diese Patch-Baseline als Standard-Patch-Baseline für Windows Server-Instances festlegen) aus.

   > **Note**
   >
   > Diese Option ist nur verfügbar, wenn Sie vor der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben. Weitere Informationen zum Festlegen einer vorhandenen Patch-Baseline als Standard finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.

8. Erstellen Sie im Abschnitt Approval Rules for operating-systems (Genehmigungsregeln für Betriebssysteme) unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

   - Produkte: Die Betriebssystemversionen, auf die sich die Genehmigungsregel bezieht, z. B. `WindowsServer2012`. Sie können eine, mehr als eine oder alle unterstützten Windows-Versionen auswählen. Die Standardauswahl ist `All`.

- Classification (Klassifizierung): Wählen Sie `ServicePacks` aus.
- Severity (Schweregrad): Der Schweregradwert der Patches, auf die die Regel angewendet werden soll. Um sicherzustellen, dass alle Service Packs von der Regel eingeschlossen werden, wählen Sie `All` aus.
- Auto-approval (Automatische Genehmigung): Die Methode zum Auswählen von Patches für die automatische Genehmigung.
  - Approve patches after a specified number of days (Patches nach einer bestimmten Anzahl von Tagen genehmigen): Die Anzahl der Tage, die der Patch Manager warten muss, nachdem ein Patch veröffentlicht oder aktualisiert wurde, bevor ein Patch automatisch genehmigt wird. Sie können jede Ganzzahl von Null (0) bis 360 eingeben. Für die meisten Szenarien empfehlen wir, nicht länger als 100 Tage zu warten.
  - Approve patches released up to a specific date (Patches genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden): Das Datum der Patch-Veröffentlichung, an dem der Patch Manager automatisch alle Patches anwendet, die bis zu diesem Datum veröffentlicht oder aktualisiert wurden. Wenn Sie beispielsweise den 07. Juli 2023 angeben, werden Patches, die am oder nach dem 08. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden, nicht automatisch installiert.
- (Optional) Compliance reporting (Compliance-Berichte): Der Schweregrad, den Sie Service Packs zuweisen möchten, die von der Baseline genehmigt wurden, z. B. `High`.

> **Note**
>
> Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Service-Packs als `Missing` gemeldet wird, dann entspricht der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

9. (Optional) Wählen Sie für Manage tags (Tags verwalten) ein oder mehrere Tag-Schlüsselname/Wertpaare für die Patch-Baseline aus.

   Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Für diese Patch-Baseline, die der Aktualisierung von Service Packs gewidmet ist, könnten Sie Schlüssel-Wert-Paare angeben, z. B.:

   - `Key=OS,Value=Windows`

- `Key=Classification,Value=ServicePacks`

10. Wählen Sie die Option Create Patch Baseline.

Tutorial: Aktualisieren von Anwendungsabhängigkeiten, Patchen eines verwalteten Knotens und Durchführen einer anwendungsspezifischen Zustandsprüfung

In vielen Fällen muss ein verwalteter Knoten neu gestartet werden, nachdem er mit dem neuesten Softwareupdate gepatcht wurde. Ein Neustart eines verwalteten Knotens in der Produktion ohne vorhandene Sicherheitsvorkehrungen kann jedoch mehrere Probleme verursachen, z. B. das Aufrufen von Alarmen, das Aufzeichnen falscher Metrikdaten und das Unterbrechen von Datensynchronisationen.

Diese Anleitung zeigt, wie Sie Probleme wie diese vermeiden können, indem Sie das AWS Systems Manager -Dokument (SSM-Dokument) `AWS-RunPatchBaselineWithHooks` verwenden, um einen komplexen, mehrstufigen Patchvorgang zu erreichen, der Folgendes ausführt:

1. Verhindern neuer Verbindungen mit der Anwendung
2. Installieren von Betriebssystem-Updates
3. Aktualisieren der Paketabhängigkeiten der Anwendung
4. Neustart des Systems
5. Durchführen einer anwendungsspezifischen Zustandsprüfung

Für dieses Beispiel haben wir unsere Infrastruktur auf diese Weise eingerichtet:

- Die anvisierten virtuellen Maschinen werden als verwaltete Knoten mit Systems Manager registriert.
- `Iptables` wird als lokale Firewall verwendet.
- Die auf den verwalteten Knoten gehostete Anwendung wird auf Port 443 ausgeführt.
- Die Anwendung, die auf den verwalteten Knoten gehostet wird, ist eine `nodeJS`-Anwendung.
- Die auf den verwalteten Knoten gehostete Anwendung wird vom pm2-Prozessmanager verwaltet.
- Die Anwendung verfügt bereits über einen angegebenen Zustandsprüfungs-Endpunkt.
- Der Endpunkt der Zustandsprüfung der Anwendung erfordert keine Endbenutzerauthentifizierung. Der Endpunkt ermöglicht eine Zustandsprüfung, die die Anforderungen der Organisation beim Festlegen der Verfügbarkeit erfüllt. (In Ihrer Umgebung reicht es möglicherweise aus,

sicherzustellen, dass die nodeJS-Anwendung ausgeführt wird und in der Lage ist, auf Anfragen zu warten. In anderen Fällen möchten Sie möglicherweise überprüfen, ob bereits eine Verbindung zur Caching-Ebene oder zur Datenbankebene hergestellt wurde).

Die Beispiele in dieser Anleitung dienen nur zu Demonstrationszwecken und sind nicht dafür gedacht, in Produktionsumgebungen implementiert zu werden. Beachten Sie auch, dass das Lebenszyklus-Hook-Feature von Patch Manager, einer Funktion von Systems Manager, mit dem `AWS-RunPatchBaselineWithHooks`-Dokument zahlreiche andere Szenarien unterstützen kann. Im Folgenden finden Sie einige Beispiele.

- Stoppen Sie einen Metriken meldenden Agenten, bevor Sie ihn patchen und neu starten, nachdem der verwaltete Knoten neu gestartet wurde.
- Trennen Sie den verwalteten Knoten vor dem Patchen von einem CRM- oder PCS-Cluster und fügen Sie sie nach dem Neustart des Knoten erneut an.
- Aktualisieren Sie Software von Drittanbietern (z. B. Java, Tomcat, Adobe-Anwendungen usw.) auf Windows Server-Maschinen nach dem Anwenden von Betriebssystem-Updates, jedoch vor dem Neustart des verwalteten Knoten.

So aktualisieren Sie Anwendungsabhängigkeite, patchen einen verwalteten Knoten und führen eine anwendungsspezifische Zustandsprüfung durch

1. Erstellen Sie ein SSM-Dokument für Ihr Vorinstallations-Skript mit dem folgenden Inhalt und geben Sie ihm den Namen `NodeJSAppPrePatch`. Ersetzen Sie *`your_application`* mit dem Namen Ihrer Anwendung.

   Dieses Skript blockiert sofort neue eingehende Anforderungen und lässt fünf Sekunden, damit bereits aktive Anforderungen abgeschlossen werden können, bevor der Patchvorgang gestartet wird. Für die `sleep`-Option geben Sie einen Wert in Sekunden an, der größer ist als die Dauer, bis eingehende Anforderungen normalerweise abgeschlossen werden.

   ```
   # 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
```

Informationen zum Erstellen von SSM-Dokumenten finden Sie unter Erstellen von SSM-Dokumentinhalten.

2. Erstellen Sie ein weiteres SSM-Dokument mit folgendem Inhalt für Ihr Postinstall-Skript, um Ihre Anwendungsabhängigkeiten zu aktualisieren, und nennen Sie es `NodeJSAppPostPatch`. Ersetzen Sie */your/application/path* mit dem Pfad zu Ihrer Anwendung.

```
cd /your/application/path
npm update
# you can use npm-check-updates if you want to upgrade major versions
```

3. Erstellen Sie ein weiteres SSM-Dokument mit folgendem Inhalt für Ihr `onExit`-Skript, um Ihre Anwendung zu sichern und eine Zustandsprüfung durchzuführen. Nennen Sie dieses SSM-Dokument `NodeJSAppOnExitPatch`. Ersetzen Sie *your_application* mit dem Namen Ihrer Anwendung.

```
# 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. Erstellen Sie eine Zuordnung inState Manager, eine Fähigkeit von AWS Systems Manager, um den Vorgang auszuführen, indem Sie die folgenden Schritte ausführen:

   1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
   2. Wählen Sie im Navigationsbereich State Manager und anschließend Create association (Zuordnung erstellen) aus.
   3. Für Name geben Sie einen Namen ein, um den Zweck der Zuordnung zu identifizieren.
   4. Wählen Sie in der Liste Dokument die Option `AWS-RunPatchBaselineWithHooks` aus.

5. Wählen Sie für Operation die Option Install (Installieren) aus.
6. (Optional) Für Snapshot-ID, stellen Sie eine GUID bereit, die Sie generieren, um den Vorgang zu beschleunigen und Konsistenz zu gewährleisten. Der GUID-Wert kann so einfach sein wie `00000000-0000-0000-0000-111122223333`.
7. Für Pre Install Hook Doc Name geben Sie `NodeJSAppPrePatch` ein.
8. Für Post Install Hook Doc Name geben Sie `NodeJSAppPostPatch` ein.
9. Geben Sie für On ExitHook Doc-Name den Wert ein`NodeJSApp0nExitPatch`.

5. Für Targets (Ziele), identifizieren Sie Ihre verwalteten Knoten, indem Sie Tags angeben, Knoten manuell auswählen, eine Ressourcengruppe auswählen oder alle verwaltete Knoten auswählen.
6. Für Specify schedule (Zeitplan angeben) geben Sie an, wie oft die Zuordnung ausgeführt werden soll. Für einen verwalteten Knoten ist das Patchen einmal pro Woche beispielsweise eine übliche Kadenz.
7. Wählen Sie im Abschnitt Rate control (Ratensteuerung) Optionen für die Ausführung der Zuordnung auf mehreren verwalteten Knoten aus. Stellen Sie sicher, dass nur ein Teil der verwalteten Knoten gleichzeitig aktualisiert wird. Andernfalls könnte die gesamte oder die meisten Ihrer Flotte gleichzeitig offline geschaltet werden. Weitere Informationen zu Ratensteuerungen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.
8. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Dienstrolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

9. Wählen Sie Create Association.

Anleitung: Patchen einer Serverumgebung (AWS CLI)

In der folgenden Prozedur wird beschrieben, wie Sie eine Serverumgebung mithilfe einer angepassten Patch-Baseline, Patch-Gruppen und einem Wartungsfenster patchen.

Bevor Sie beginnen

- Installieren oder Aktualisieren des SSM Agent auf Ihren verwalteten Knoten. Um von Linux verwaltete Knoten zu patchen, müssen Ihre Knoten SSM Agent der Version 2.0.834.0 oder höher ausführen. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.
- Konfigurieren Sie Rollen und Berechtigungen für Maintenance Windows, eine Funktion von AWS Systems Manager. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.
- Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), wenn noch nicht erfolgt.

  Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

So konfigurieren Sie Patch Manager und spielen Patches für verwaltete Knoten ein (Befehlszeile)

1. Führen Sie den folgenden Befehl aus, um eine Patch-Baseline für Windows mit dem Namen `Production-Baseline` zu erstellen. Diese Patch-Baseline genehmigt Patches für eine Produktionsumgebung sieben Tage nach ihrer Veröffentlichung oder letzten Aktualisierung. Darüber hinaus wurde die Patch-Baseline markiert, um anzuzeigen, dass sie für eine Produktionsumgebung bestimmt ist.

   > **Note**
   >
   > Der `OperatingSystem`-Parameter und `PatchFilters` variieren je nach Betriebssystem der anvisierten verwalteten Knoten, für die die Patch-Baseline gilt. Weitere Informationen finden Sie unter OperatingSystem und 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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

2. Führen Sie die folgenden Befehle aus, um die Patch-Baseline „Production-Baseline“ für zwei Patchgruppen zu registrieren. Die Gruppen heißen „Datenbankserver“ und „Front-End-Server“.

   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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "PatchGroup":"Front-End Servers",
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

3. Führen Sie die folgenden Befehle aus, um zwei Wartungsfenster für die Produktionsserver zu erstellen. Das erste Zeitfenster beginnt jeden Dienstag um 20:00 Uhr. Das zweite Zeitfenster beginnt jeden Samstag um 22:00 Uhr. Darüber hinaus wird das Wartungsfenster mit Tags versehen, um anzugeben, das es für eine Produktionsumgebung vorgesehen ist.

   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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowId":"mw-9a8b7c6d5eEXAMPLE"
}
```

4. Führen Sie die folgenden Befehle aus, um die Server-Patch-Gruppen `Database` und `Front-End` mit ihren jeweiligen Wartungsfenstern zu registrieren.

   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"
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
      "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"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowTargetId":"faa01c41-1d57-496c-ba77-ff9caEXAMPLE"
}
```

5. Führen Sie die folgenden Befehle aus, um eine Patch-Aufgabe zu registrieren, die während der entsprechenden Wartungsfenster fehlende Updates auf den Servern `Database` und `Front-End` installiert.

   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}}"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "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}}"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowTaskId":"8a5c4629-31b0-4edd-8aea-33698EXAMPLE"
}
```

6. Führen Sie den folgenden Befehl aus, um für eine Patch-Gruppe eine allgemeine Zusammenfassung zur Patch-Compliance abzurufen. Die allgemeine Zusammenfassung der Patch-Compliance enthält die Anzahl der verwalteten Knoten mit Patches in den jeweiligen Patch-Zuständen.

> **Note**
>
> Es werden Nullen für die Anzahl der verwalteten Knoten in der Zusammenfassung erwartet, bis die Patch-Aufgabe während des ersten Wartungsfensters ausgeführt wird.

Linux & macOS

```
aws ssm describe-patch-group-state \
    --patch-group "Database Servers"
```

Windows Server

```
aws ssm describe-patch-group-state ^
    --patch-group "Database Servers"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "Instances": number,
   "InstancesWithFailedPatches": number,
   "InstancesWithInstalledOtherPatches": number,
   "InstancesWithInstalledPatches": number,
   "InstancesWithInstalledPendingRebootPatches": number,
   "InstancesWithInstalledRejectedPatches": number,
   "InstancesWithMissingPatches": number,
   "InstancesWithNotApplicablePatches": number,
```

```
   "InstancesWithUnreportedNotApplicablePatches": number
}
```

7. Führen Sie den folgenden Befehl aus, um für eine Patch-Gruppe eine Übersicht über den Patch-Zustand auf der Ebene einzelner verwalteter Knoten abzurufen. Die Zusammenfassung pro verwalteter Knoten enthält eine Anzahl von Patches in den jeweiligen Patch-Zuständen pro verwalteten Knoten für eine Patch-Gruppe.

   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"
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
      "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
        }
    ]
}
```

Weitere Beispiele anderer AWS CLI-Befehle, die Sie für Patch Manager-Konfigurationsaufgaben verwenden, können, finden Sie unter Arbeiten mit Patch Manager (AWS CLI).

Fehlerbehebung für Patch Manager

Verwenden Sie die folgenden Informationen, um Probleme mit Patch Manager einer Funktion von zu beheben AWS Systems Manager.

Themen

- Problem: Fehler „Invoke-PatchBaselineOperation : Zugriff verweigert“ oder Fehler „Datei kann nicht von S3 heruntergeladen werden“ für baseline_overrides.json
- Problem: Das Patchen schlägt fehl, ohne dass eine offensichtliche Ursache oder Fehlermeldung vorliegt
- Problem: Unerwartete Patch-Compliance-Ergebnisse
- Fehler beim Ausführen von AWS-RunPatchBaseline unter Linux
- Fehler beim Ausführen von AWS-RunPatchBaseline unter Windows Server
- Kontaktaufnahme mit AWS Support

Problem: Fehler „Invoke-PatchBaselineOperation : Zugriff verweigert“ oder Fehler „Datei kann nicht von S3 heruntergeladen werden“ für **`baseline_overrides.json`**

Problem: Wenn die von Ihrer Patch-Richtlinie festgelegten Patching-Vorgänge ausgeführt werden, erhalten Sie eine Fehlermeldung ähnlich dem folgenden Beispiel.

Example error on Windows Server

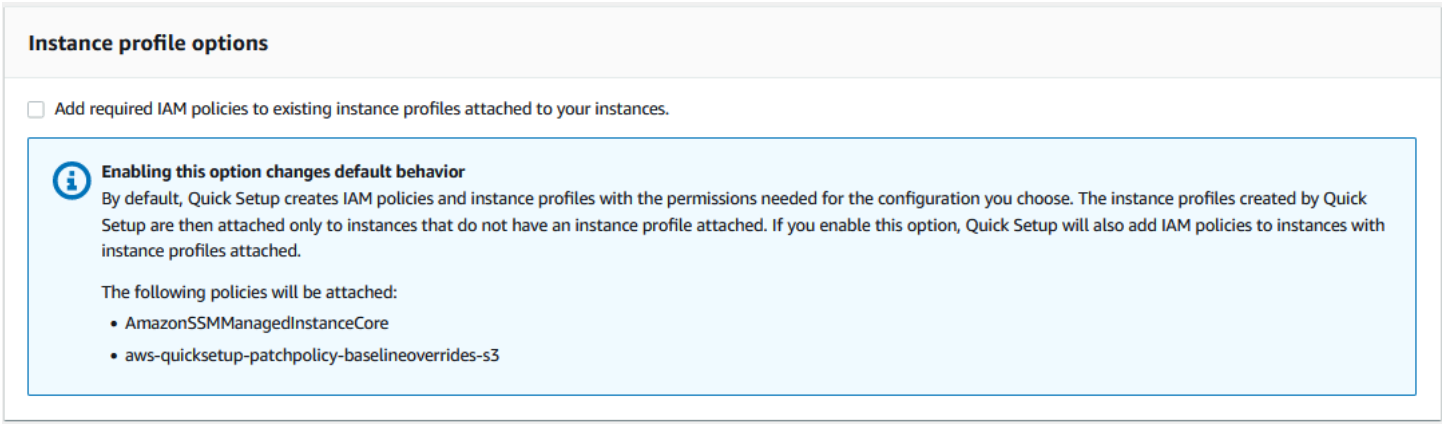
```
----------ERROR-------
Invoke-PatchBaselineOperation : Access Denied
At C:\ProgramData\Amazon\SSM\InstanceData\i-02573cafcfEXAMPLE\document\orchestr
ation\792dd5bd-2ad3-4f1e-931d-abEXAMPLE\PatchWindows\_script.ps1:219 char:13
+ $response = Invoke-PatchBaselineOperation -Operation Install -Snapsho ...
```

```
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
+ 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.
```

Ursache: Sie haben eine Patch-Richtlinie in Quick Setup erstellt, und einige Ihrer verwalteten Knoten waren bereits mit einem Instance-Profil (für EC2-Instances) oder einer Servicerolle (für Nicht-EC2-Maschinen) versehen. Sie haben jedoch das Kontrollkästchen Erforderliche IAM-Richtlinien zu vorhandenen Instance-Profilen hinzufügen, die an Ihre Instances angehängt sind, nicht aktiviert, wie in der folgenden Abbildung dargestellt.



Wenn Sie eine Patch-Richtlinie erstellen, wird auch ein Amazon-S3-Bucket erstellt, in dem die `baseline_overrides.json` Konfigurationsdatei der Richtlinie gespeichert wird. Wenn Sie bei der Erstellung der Richtlinie das Kontrollkästchen Erforderliche IAM-Richtlinien zu bestehenden Instance-Profilen hinzufügen, die mit Ihren Instances verbunden sind, nicht aktivieren, werden die IAM-Richtlinien und Ressourcen-Tags, die für den Zugriff auf `baseline_overrides.json` im S3-Bucket erforderlich sind, nicht automatisch zu Ihren bestehenden IAM-Instance-Profilen und Servicerollen hinzugefügt.

Lösung 1: Löschen Sie die bestehende Patch-Richtlinienkonfiguration und erstellen Sie dann eine neue. Aktivieren Sie dabei das Kontrollkästchen Erforderliche IAM-Richtlinien zu bestehenden Instance-Profilen hinzufügen, die mit Ihren Instances verbunden sind. Diese Auswahl wendet die mit dieser Quick Setup-Konfiguration erstellten IAM-Richtlinien auf Knoten an, denen bereits ein Instance-Profil oder eine Servicerolle zugewiesen ist. (Quick Setup fügt standardmäßig die erforderlichen Richtlinien zu Instances und Knoten hinzu, die noch nicht über Instance-Profile oder Servicerollen verfügen.) Weitere Informationen finden Sie unter Automatisieren von unternehmensweitem Patching mithilfe einer Quick Setup-Patch-Richtlinie.

Lösung 2: Fügen Sie die erforderlichen Berechtigungen und Tags manuell zu jedem IAM-Instance-Profil und jeder IAM-Servicerolle hinzu, die Sie mit Quick Setup verwenden. Anweisungen finden Sie unter Berechtigungen für den S3-Bucket mit der Patch-Richtlinie.

Problem: Das Patchen schlägt fehl, ohne dass eine offensichtliche Ursache oder Fehlermeldung vorliegt

Problem: Ein Patch-Vorgang schlägt fehl, ohne dass eine Fehlermeldung zurückgegeben wird.

Mögliche Ursache: Wenn mehr als ein Aufruf von `AWS-RunPatchBaseline` gleichzeitig erfolgt, können sie miteinander in Konflikt geraten, sodass Patch-Aufgaben fehlschlagen. Dies wird möglicherweise nicht in den Patchprotokollen angegeben.

Um zu überprüfen, ob sich gleichzeitige Patch-Vorgänge möglicherweise gegenseitig unterbrochen haben, überprüfen Sie den Befehlsverlauf inRun Command, eine Funktion von. AWS Systems Manager Prüfen Sie bei einem verwalteten Knoten mit einem Patching-Fehler, ob mehrere Vorgänge innerhalb von 2 Minuten nacheinander versucht haben, die Maschine zu patchen. Dieses Szenario kann manchmal zu einem Fehler führen.

Sie können die AWS Command Line Interface (AWS CLI) auch verwenden, um mithilfe des folgenden Befehls nach gleichzeitigen Patchversuchen zu suchen. Ersetzen Sie den Wert für *`node-id`* durch die ID für Ihren verwalteten Knoten.

```
aws ssm list-commands \
    --filter "key=DocumentName,value=AWS-RunPatchBaseline" \
    --query 'Commands[*].
{CommandId:CommandId,RequestedDateTime:RequestedDateTime,Status:Status}' \
    --instance-id node-id \
    --output table
```

Lösung: Wenn Sie feststellen, dass das Patching aufgrund konkurrierender Patching-Operationen auf demselben verwalteten Knoten fehlgeschlagen ist, passen Sie Ihre Patching-Konfigurationen an, damit dies nicht noch einmal geschieht. Wenn zum Beispiel zwei Wartungsfenster sich überschneidende Patching-Zeiten angeben, entfernen oder ändern Sie eines davon. Wenn in einem Wartungsfenster eine Patching-Operation angegeben ist, in einer Patch-Richtlinie jedoch eine andere für dieselbe Zeit angegeben ist, sollten Sie die Aufgabe aus dem Wartungsfenster entfernen.

Wenn Sie feststellen, dass widersprüchliche Patching-Operationen in diesem Szenario nicht die Ursache für den Ausfall waren, empfehlen wir Ihnen, sich an AWS Support zu wenden.

Problem: Unerwartete Patch-Compliance-Ergebnisse

Problem: Bei der Überprüfung der nach einem `Scan`-Vorgang generierten Details zur Patching-Compliance enthalten die Ergebnisse Informationen, die nicht die in Ihrer Patch-Baseline festgelegten Regeln widerspiegeln. Beispielsweise wird eine Ausnahme, die Sie der Liste Rejected patches (Abgelehnte Patches) in einer Patch-Baseline hinzugefügt haben, als `Missing` aufgeführt. Oder als `Important` klassifizierte Patches werden als fehlend aufgeführt, obwohl Ihre Patch-Baseline nur `Critical`-Patches angibt.

Ursache: Patch Manager unterstützt derzeit mehrere Methoden zum Ausführen von `Scan`-Operationen:

- Eine in Quick Setup konfigurierte Patch-Richtlinie
- Eine in Quick Setup konfigurierte Host-Management-Option
- Ein Wartungsfenster zum Ausführen eines Patch-`Scan`s oder einer `Install`-Aufgabe
- Eine On-Demand Patch now-Operation (Jetzt patchen)

Wenn eine `Scan`-Operation ausgeführt wird, überschreibt dies die Compliance-Details aus dem letzten Scan. Wenn Sie mehr als eine Methode zum Ausführen einer `Scan`-Operation eingerichtet haben und diese unterschiedliche Patch-Baselines mit unterschiedlichen Regeln verwenden, führt dies zu unterschiedlichen Patch-Compliance-Ergebnissen.

Lösung: Um unerwartete Patch-Compliance-Ergebnisse zu vermeiden, empfehlen wir, jeweils nur eine Methode zum Ausführen der Patch Manager `Scan`-Operation zu verwenden. Weitere Informationen finden Sie unter Vermeiden von unbeabsichtigtem Überschreiben von Patch-Compliance-Daten.

Fehler beim Ausführen von **`AWS-RunPatchBaseline`** unter Linux

Themen

- Problem: Fehler 'No such file or directory'
- Problem: Fehler 'another process has acquired yum lock'
- Problem: Fehler 'Permission denied / failed to run commands'
- Problem: Fehler 'Unable to download payload'
- Problem: Fehler 'unsupported package manager and python version combination'
- Problem: Patch Manager wendet keine Regeln an, die zum Ausschließen bestimmter Pakete angegeben sind
- Problem: Patching schlägt fehl und Patch Manager meldet, dass die Erweiterung „Servername Indication“ für TLS nicht verfügbar ist
- Problem: Patch Manager meldet 'No more mirrors to try'
- Problem: Patching schlägt fehl mit 'Error code returned from curl is 23'
- Problem: Patching schlägt mit der Meldung 'Error unpacking rpm package...' fehl
- Problem: Das Patchen schlägt fehl und die Meldung „Beim Herunterladen von Paketen sind Fehler aufgetreten“ wird angezeigt
- Problem: Patching schlägt fehl mit der Meldung 'Die folgenden Signaturen konnten nicht verifiziert werden, da der öffentliche Schlüssel nicht verfügbar ist'
- Problem: Das Patchen schlägt fehl und es wird eine Meldung 'NoMoreMirrorsRepoError' angezeigt
- Problem: Das Patchen schlägt mit der Meldung „Payload kann nicht heruntergeladen werden“ fehl
- Problem: Das Patchen schlägt fehl und es wird die Meldung „Installationsfehler: dpkg: Fehler:dpkg-Frontend ist durch einen anderen Prozess gesperrt“ angezeigt
- Problem: Das Patchen auf Ubuntu Server schlägt fehl und es wird der Fehler „dpkg wurde unterbrochen“ angezeigt
- Problem: Das Paketmanager-Dienstprogramm kann eine Paketabhängigkeit nicht auflösen

Problem: Fehler 'No such file or directory'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit einem der folgenden Fehler fehl.

```
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
```

Ursache 1: Zwei Befehle zum Ausführen von `AWS-RunPatchBaseline` wurden gleichzeitig auf demselben verwalteten Knoten ausgeführt. Dies erzeugt eine Race-Bedingung, die in der temporären `file patch-baseline-operations*` nicht richtig erstellt oder auf die nicht richtig zugegriffen wird.

Ursache 2: Unzureichender Speicherplatz verbleibt im `/var`-Verzeichnis.

Lösung 1: Stellen Sie sicher, dass kein Wartungsfenster zwei oder mehr Run Command-Aufgaben hat, die `AWS-RunPatchBaseline` mit der gleichen Prioritätsstufe und auf denselben Ziel-IDs ausführen. Wenn dies der Fall ist, ordnen Sie die Priorität neu an. Run Command ist eine Funktion von AWS Systems Manager.

Lösung 2: Stellen Sie sicher, dass jeweils nur ein Wartungsfenster Run Command-Aufgaben ausführt, die `AWS-RunPatchBaseline` auf denselben Zielen und nach demselben Zeitplan verwenden. Ändern Sie in diesem Fall den Zeitplan.

Lösung 3: Stellen Sie sicher, dass nur eine State Manager-Zuordnung `AWS-RunPatchBaseline` nach demselben Zeitplan ausführt und die gleichen verwalteten Knoten anvisiert. State Manager ist eine Funktion von AWS Systems Manager.

Lösung 4: Machen Sie genügend Speicherplatz im `/var`-Verzeichnis für die Update-Pakete. frei

Problem: Fehler 'another process has acquired yum lock'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit dem folgenden Fehler fehl.

```
12/20/2019 21:41:48 root [INFO]: another process has acquired yum lock, waiting 2 s and
 retry.
```

Ursache: Das `AWS-RunPatchBaseline`-Dokument wurde auf einem verwalteten Knoten ausgeführt, in dem es bereits in einer anderen Operation ausgeführt wird und den `yum`-Paketmanager-Prozess erhalten hat.

Lösung: Stellen Sie sicher, dass keine State Manager-Zuordnung, Aufgaben im Wartungsfenster oder andere Konfigurationen, die `AWS-RunPatchBaseline`nach einem Zeitplan ausführen, ungefähr zur gleichen Zeit denselben verwalteten Knoten als Ziel haben.

Problem: Fehler 'Permission denied / failed to run commands'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit dem folgenden Fehler fehl.

```
sh:
/var/lib/amazon/ssm/instanceid/document/orchestration/commandid/PatchLinux/_script.sh:
 Permission denied
failed to run commands: exit status 126
```

Ursache: `/var/lib/amazon/` könnte mit `noexec`-Berechtigungen gemountet sein. Dies ist ein Problem, weil SSM Agent Payload-Skripte auf `/var/lib/amazon/ssm` herunterlädt und sie von diesem Speicherort ausführt.

Lösung: Stellen Sie sicher, dass Sie exklusive Partitionen für `/var/log/amazon` und `/var/lib/amazon` konfiguriert haben und sind mit `exec`-Berechtigungen gemountet sind.

Problem: Fehler 'Unable to download payload'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit dem folgenden Fehler fehl.

```
Unable to download payload: https://s3.DOC-EXAMPLE-BUCKET.region.amazonaws.com/
aws-ssm-region/patchbaselineoperations/linux/payloads/patch-baseline-operations-
X.XX.tar.gz.failed to run commands: exit status 156
```

Ursache: Der verwaltete Knoten verfügt nicht über die erforderlichen Berechtigungen für den Zugriff auf den angegebenen Amazon Simple Storage Service (Amazon S3)-Bucket.

Lösung: Aktualisieren Sie Ihre Netzwerkkonfiguration so, dass S3-Endpunkte erreichbar sind. Weitere Informationen finden Sie unter den Informationen zum erforderlichen Zugriff auf S3-Buckets für Patch Manager in SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets.

Problem: Fehler 'unsupported package manager and python version combination'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit dem folgenden Fehler fehl.

```
An unsupported package manager and python version combination was found. Apt requires
 Python3 to be installed.
failed to run commands: exit status 1
```

Ursache: Eine unterstützte Version von Python3 ist nicht auf der Instance Debian Server, Raspberry Pi OS oder Ubuntu Server installiert.

Lösung: Installieren Sie eine unterstützte Version von python3 (3.0–3.10) auf dem Server, die für verwaltete Debian Server-, Raspberry Pi OS- und Ubuntu Server-Knoten erforderlich ist.

Problem: Patch Manager wendet keine Regeln an, die zum Ausschließen bestimmter Pakete angegeben sind

Problem: Sie haben versucht, bestimmte Pakete auszuschließen, indem Sie sie in der `/etc/yum.conf`-Datei im Format `exclude=`*`package-name`* angeben, aber sie werden nicht während der Patch Manager-Operation `Install` ausgeschlossen.

Ursache: Patch Manager enthält keine Ausschlüsse, die in der `/etc/yum.conf`-Datei angegeben sind.

Lösung: Um bestimmte Pakete auszuschließen, erstellen Sie eine benutzerdefinierte Patch-Baseline und eine Regel, um die Pakete auszuschließen, die nicht installiert werden sollen.

Problem: Patching schlägt fehl und Patch Manager meldet, dass die Erweiterung „Servername Indication“ für TLS nicht verfügbar ist

Problem: Der Patchvorgang gibt die folgende Meldung aus.

```
/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
```

Ursache: Diese Meldung zeigt keinen Fehler an. Stattdessen ist dies eine Warnung, dass die ältere Version von Python, die mit dem Betriebssystem vertrieben wird, TLS Server Name Indication nicht

unterstützt. Das Systems Manager Manager-Patch-Payload-Skript gibt diese Warnung aus, wenn eine Verbindung zu AWS APIs hergestellt wird, die SNI unterstützen.

Lösung: Um Patching-Fehler zu beheben, wenn diese Meldung gemeldet wird, überprüfen Sie den Inhalt der `stdout`- und `stderr`-Dateien. Wenn Sie die Patch-Baseline nicht so konfiguriert haben, dass diese Dateien in einem S3-Bucket oder in Amazon CloudWatch Logs gespeichert werden, können Sie die Dateien am folgenden Speicherort auf Ihrem verwalteten Linux-Node finden.

`/var/lib/amazon/ssm/`*`instance-id`*`/document/orchestration/`*`Run-Command-execution-id`*`/awsrunShellScript/PatchLinux`

Problem: Patch Manager meldet 'No more mirrors to try'

Problem: Der Patchvorgang gibt die folgende Meldung aus.

```
[Errno 256] No more mirrors to try.
```

Ursache: Die auf dem verwalteten Knoten konfigurierten Repositorys funktionieren nicht richtig. Mögliche Gründe hierfür sind:

- Das yum-Cache ist beschädigt.
- Eine Repository-URL kann aufgrund von Netzwerkproblemen nicht erreicht werden.

Lösung: Patch Manager verwendet den Standard-Paketmanager des verwalteten Knoten, um die Patching-Operation durchzuführen. Überprüfen Sie, ob Repositorys richtig konfiguriert sind und funktionieren.

Problem: Patching schlägt fehl mit 'Error code returned from curl is 23'

Problem: Eine Patching-Operation, die `AWS-RunPatchBaseline` verwendet, schlägt mit einer Fehlermeldung ähnlich der folgenden fehl:

```
05/01/2023 17:04:30 root [ERROR]: Error code returned from curl is 23
```

Ursache: Das auf Ihren Systemen verwendete Curl-Tool verfügt nicht über die erforderlichen Rechte, um in das Dateisystem zu schreiben. Dies kann vorkommen, wenn das Standard-Curl-Tool des Paketmanagers durch eine andere Version ersetzt wurde, beispielsweise durch eine, die mit snap installiert wurde.

Lösung: Wenn die vom Paketmanager bereitgestellte curl-Version deinstalliert wurde, während eine andere Version installiert wurde, installieren Sie sie erneut.

Wenn Sie mehrere curl-Versionen installiert halten müssen, stellen Sie sicher, dass sich die mit dem Paketmanager verknüpfte Version im ersten in der PATH-Variable aufgeführten Verzeichnis befindet. Sie können dies überprüfen, indem Sie den Befehl `echo $PATH` ausführen, um die aktuelle Reihenfolge der Verzeichnisse zu sehen, die auf Ihrem System auf ausführbare Dateien überprüft werden.

Problem: Patching schlägt mit der Meldung 'Error unpacking rpm package...' fehl

Problem: Ein Patch-Vorgang schlägt mit einem Fehler ähnlich dem folgenden fehl:

```
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
```

Ursache 1: Wenn ein bestimmtes Paket in mehreren Paket-Installationsprogrammen vorhanden ist, z. B. sowohl in `pip` als auch in `yum` oder `dnf`, kann es bei der Verwendung des Standard-Paketmanagers zu Konflikten kommen.

Ein häufiges Beispiel ist das `urllib3`-Paket, das sich in `pip`, `yum` und `dnf` befindet.

Ursache 2: Das `python-urllib3`-Paket ist beschädigt. Dies kann passieren, wenn die Paketdateien von `pip` installiert oder aktualisiert wurden, nachdem das `rpm`-Paket zuvor von `yum` oder `dnf` installiert wurde.

Lösung: Entfernen Sie das `python-urllib3`-Paket aus Pip, indem Sie den Befehl `sudo pip uninstall urllib3` ausführen, und behalten Sie das Paket nur im Standard-Paketmanager (`yum` oder `dnf`) bei.

Problem: Das Patchen schlägt fehl und die Meldung „Beim Herunterladen von Paketen sind Fehler aufgetreten“ wird angezeigt

Problem: Beim Patchen erhalten Sie eine Fehlermeldung, die der folgenden ähnelt:

```
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',
```

Ursache: Dieser Fehler kann auftreten, wenn auf einem verwalteten Knoten nicht genügend Speicher verfügbar ist.

Lösung: Konfigurieren Sie den Swap-Speicher oder aktualisieren Sie die Instance auf einen anderen Typ, um die Speicherunterstützung zu erhöhen. Starten Sie dann einen neuen Patch-Vorgang.

Problem: Patching schlägt fehl mit der Meldung 'Die folgenden Signaturen konnten nicht verifiziert werden, da der öffentliche Schlüssel nicht verfügbar ist'

Problem: Das Patchen schlägt bei Ubuntu Server mit einer Fehlermeldung ähnlich der folgenden fehl:

```
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
```

Ursache: Der Schlüssel für GNU Privacy Guard (GPG) ist abgelaufen oder fehlt.

Lösung: Aktualisieren Sie den GPG-Schlüssel, oder fügen Sie den Schlüssel erneut hinzu.

Anhand des zuvor gezeigten Fehlers sehen wir zum Beispiel, dass der `467B942D3A79BD29`-Schlüssel fehlt und hinzugefügt werden muss. Führen Sie dazu einen der folgenden Befehle aus:

```
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
```

Oder, um alle Schlüssel zu aktualisieren, führen Sie den folgenden Befehl aus:

```
sudo apt-key adv --keyserver hkps://keyserver.ubuntu.com --refresh-keys
```

Wenn der Fehler danach weiterhin auftritt, empfehlen wir, das Problem an die Organisation zu melden, die das Repository verwaltet. Bis ein Fix verfügbar ist, können Sie die `/etc/apt/sources.list`-Datei so bearbeiten, dass das Repository während des Patchvorgangs ausgelassen wird.

Öffnen Sie dazu die `sources.list`-Datei zur Bearbeitung, suchen Sie die Zeile für das Repository und fügen Sie am Anfang der Zeile ein #-Zeichen ein, um sie auszukommentieren. Speichern und schließen Sie dann die Datei.

Problem: Das Patchen schlägt fehl und es wird eine Meldung 'NoMoreMirrorsRepoError' angezeigt

Problem: Sie erhalten eine Fehlermeldung ähnlich der folgenden:

```
NoMoreMirrorsRepoError: failure: repodata/repomd.xml from pgdg94: [Errno 256] No more
 mirrors to try.
```

Ursache: Im Quell-Repository ist ein Fehler aufgetreten.

Lösung: Wir empfehlen, das Problem der Organisation zu melden, die das Repository verwaltet. Bis der Fehler behoben ist, können Sie das Repository auf Betriebssystemebene deaktivieren. Führen Sie dazu den folgenden Befehl aus und ersetzen Sie den Wert für *repo-name* durch Ihren Repository-Namen:

```
yum-config-manager --disable repo-name
```

Im Folgenden sehen Sie ein Beispiel.

```
yum-config-manager --disable pgdg94
```

Nachdem Sie diesen Befehl ausgeführt haben, führen Sie einen weiteren Patch-Vorgang aus.

Problem: Das Patchen schlägt mit der Meldung „Payload kann nicht heruntergeladen werden“ fehl

Problem: Sie erhalten eine Fehlermeldung ähnlich der folgenden:

```
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
```

Ursache: Die Konfiguration des verwalteten Knotens ist fehlerhaft oder unvollständig.

Lösung: Versichern Sie sich, dass der verwaltete Knoten wie folgt konfiguriert ist:

- Ausgehende TCP-443-Regel in der Sicherheitsgruppe.
- Ausgehende TCP-443-Regel in NACL.
- TCP-Regel 1024-65535 für eingehenden Datenverkehr in NACL.
- NAT/IGW in der Routing-Tabelle, um Konnektivität zu einem S3-Endpunkt bereitzustellen. Wenn die Instance keinen Internetzugang hat, stellen Sie ihr Konnektivität mit dem S3-Endpunkt zur

Verfügung. Fügen Sie dazu einen S3-Gateway-Endpunkt in der VPC hinzu und integrieren Sie ihn in die Routing-Tabelle des verwalteten Knotens.

Problem: Das Patchen schlägt fehl und es wird die Meldung „Installationsfehler: dpkg: Fehler:dpkg-Frontend ist durch einen anderen Prozess gesperrt“ angezeigt

Problem: Das Patchen schlägt mit einem Fehler ähnlich dem folgenden fehl:

```
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
```

Ursache: Der Paketmanager führt bereits einen anderen Prozess auf einem verwalteten Knoten auf Betriebssystemebene aus. Wenn der Abschluss dieses anderen Prozesses viel Zeit in Anspruch nimmt, kann es bei der Patch-Operation von Patch Manager zu einem Timeout kommen und ein Fehler auftreten.

Lösung: Führen Sie nach Abschluss des anderen Prozesses, der den Paketmanager verwendet, einen neuen Patchvorgang aus.

Problem: Das Patchen auf Ubuntu Server schlägt fehl und es wird der Fehler „dpkg wurde unterbrochen“ angezeigt

Problem: Auf Ubuntu Server schlägt das Patchen mit einem Fehler ähnlich dem folgenden fehl:

```
E: dpkg was interrupted, you must manually run
'dpkg --configure -a' to correct the problem.
```

Ursache: Ein oder mehrere Pakete sind falsch konfiguriert.

Lösung: Führen Sie die folgenden Schritte aus:

1. Prüfen Sie, welche Pakete betroffen sind und welche Probleme bei den einzelnen Paketen bestehen, indem Sie nacheinander die folgenden Befehle ausführen:

   ```
   sudo apt-get check
   ```

   ```
   sudo dpkg -C
   ```

```
dpkg-query -W -f='${db:Status-Abbrev} ${binary:Package}\n' | grep -E ^.[^nci]
```

2. Korrigieren Sie die fehlerhaften Pakete, indem Sie den folgenden Befehl ausführen:

```
sudo dpkg --configure -a
```

3. Wenn der vorherige Befehl das Problem nicht vollständig behoben hat, führen Sie den folgenden Befehl aus:

```
sudo apt --fix-broken install
```

Problem: Das Paketmanager-Dienstprogramm kann eine Paketabhängigkeit nicht auflösen

Problem: Der native Paketmanager auf dem verwalteten Knoten kann eine Paketabhängigkeit nicht auflösen und das Patchen schlägt fehl. Das folgende Beispiel für eine Fehlermeldung weist auf diese Art von Fehler auf einem Betriebssystem hin, das `yum` als Paketmanager verwendet.

```
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']
```

Ursache: Patch Manager verwendet auf Linux-Betriebssystemen den systemeigenen Paketmanager auf der Maschine, um Patch-Operationen wie `yum`, `dnf`, `apt` und `zypper` auszuführen. Die Anwendungen erkennen, installieren, aktualisieren oder entfernen abhängige Pakete bei Bedarf automatisch. Einige Bedingungen können jedoch dazu führen, dass der Paketmanager einen Abhängigkeitsvorgang nicht abschließen kann, wie zum Beispiel:

- Auf dem Betriebssystem sind mehrere widersprüchliche Repositorys konfiguriert.
- Auf eine Remote-Repository-URL kann aufgrund von Netzwerkproblemen nicht zugegriffen werden.
- Im Repository wurde ein Paket für die falsche Architektur gefunden.

Lösung: Das Patchen kann aufgrund eines Abhängigkeitsproblems aus einer Vielzahl von Gründen fehlschlagen. Wir empfehlen Ihnen daher, sich an uns AWS Support zu wenden, um Hilfe bei der Fehlerbehebung zu erhalten.

Fehler beim Ausführen von **`AWS-RunPatchBaseline`** unter Windows Server

Themen

- Problem: Nicht übereinstimmende Produktfamilien/Produktpaare
- Problem: AWS-RunPatchBaseline-Ausgabe gibt einen HRESULT (Windows Server) zurück
- Problem: Der verwaltete Knoten hat keinen Zugriff auf Windows Update Catalog oder WSUS
- Problem: Das PatchBaselineOperations PowerShell Modul kann nicht heruntergeladen werden
- Problem: fehlende Patches

Problem: Nicht übereinstimmende Produktfamilien/Produktpaare

Problem: Wenn Sie eine Patch-Baseline in der Systems Manager-Konsole erstellen, geben Sie eine Produktfamilie und ein Produkt an. Beispiel:

- Product Family (Produktfamilie): `Office`

  Produkt: `Office 2016`

Ursache: Wenn Sie versuchen, eine Patch-Baseline mit nicht übereinstimmender Produktfamilie/Produkt zu erstellen, wird eine Fehlermeldung angezeigt. Dies kann folgende Ursachen haben:

- Sie haben eine gültige Kombination aus Produktfamilie und Produktpaar ausgewählt, dann jedoch die Auswahl der Produktfamilie entfernt.
- Sie haben ein Produkt aus der Unterliste Obsolete or mismatched options (Veraltete oder nicht übereinstimmende Optionen) statt aus der Unterliste Available and matching options (Verfügbare und übereinstimmende Optionen) ausgewählt.

  Artikel in der Produktunterliste Veraltete oder nicht übereinstimmende Optionen wurden möglicherweise fälschlicherweise über ein SDK oder den Befehl AWS Command Line Interface ()AWS CLI eingegeben. `create-patch-baseline` Dadurch kann es zu einem Schreibfehler oder einer falschen Zuordnung eines Produkts zu einer Produktfamilie kommen. Ein Produkt kann auch in der Unterliste Obsolete or mismatched options (Veraltete oder nicht übereinstimmende Optionen) enthalten sein, wenn es für eine vorherige Patch-Baseline angegeben wurde, aber keine Patches für das Produkt von Microsoft verfügbar sind.

Lösung: Um dieses Problem in der Konsole zu vermeiden, wählen Sie immer Optionen aus den Unterlisten Currently available options (Derzeit verfügbare Optionen) aus.

Um diejenigen Produkte anzuzeigen, für die Patches verfügbar sind, können Sie auch den Befehl `describe-patch-properties` in der AWS CLI oder den API-Befehl `DescribePatchProperties` verwenden.

Problem: **`AWS-RunPatchBaseline`**-Ausgabe gibt einen **`HRESULT`** (Windows Server) zurück

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
----------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
```

Ursache: Diese Ausgabe zeigt an, dass die nativen Windows Update-APIs die Patching-Vorgänge nicht ausführen konnten.

Lösung: Überprüfen Sie den `HResult`-Code in den folgenden Themen auf microsoft.com, um Schritte zur Fehlerbehebung zum Beheben des Fehlers zu ermitteln:

- Windows-Update-Fehlercodes nach Komponenten
- Häufige Fehler und Abhilfemaßnahmen für Windows Update

Problem: Der verwaltete Knoten hat keinen Zugriff auf Windows Update Catalog oder WSUS

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
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----
```

Ursache: Dieser Fehler kann mit den Windows Update-Komponenten oder einer fehlenden Konnektivität zum Windows Update Catalog oder Windows Server Update Services (WSUS) zusammenhängen.

Lösung: Bestätigen Sie, dass der verwaltete Knoten über ein Internet-Gateway, ein NAT-Gateway oder eine NAT-Instance eine Verbindung zum Microsoft Update Catalog hergestellt hat. Wenn Sie WSUS verwenden, bestätigen Sie, dass der verwaltete Knoten eine Verbindung zum WSUS-Server in Ihrer Umgebung hat. Wenn Konnektivität für das beabsichtigte Ziel verfügbar ist, überprüfen Sie die Microsoft-Dokumentation auf andere mögliche Ursachen für `HResult 0x80072EE2`. Dies kann auf ein Problem auf Betriebssystemebene hinweisen.

Problem: Das PatchBaselineOperations PowerShell Modul kann nicht heruntergeladen werden

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
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
```

Lösung: Überprüfen Sie die Konnektivität und Berechtigungen für Amazon Simple Storage Service (Amazon S3) des verwalteten Knoten. Für die Rolle des verwalteten Knotens AWS Identity and Access Management (IAM) müssen die unter angegebenen Mindestberechtigungen verwendet werden. SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets Der Knoten muss über den Amazon-S3-Gateway-Endpunkt, das NAT-Gateway oder das Internet-Gateway mit dem Amazon-

S3-Endpunkt kommunizieren. Weitere Informationen zu den VPC-Endpunktanforderungen für AWS Systems Manager SSM Agent (SSM Agent) finden Sie unterVerbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

Problem: fehlende Patches

Problem: `AWS-RunPatchbaseline` wurde erfolgreich abgeschlossen, aber es fehlen einige Patches.

Nachfolgend finden Sie einige häufige Auslöser und deren Lösungen.

Ursache 1: Die Baseline ist nicht effektiv.

Lösung 1: Führen Sie die folgenden Schritte aus, um zu überprüfen, ob dies die Ursache ist.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie die Registerkarte Befehlsverlauf und dann den Befehl aus, dessen Baseline Sie überprüfen möchten.
4. Wählen Sie den verwalteten Knoten aus, dem Patches fehlen.
5. Wählen Sie Schritt 1 – Ausgabe aus und finden Sie den `BaselineId`-Wert.
6. Aktivieren Sie die zugewiesene Patch-Baseline-Konfiguration, d. h. Betriebssystem, Produktname, Klassifizierung und Schweregrad für die Patch-Baseline.
7. Rufen Sie den Microsoft Update Catalog auf.
8. Durchsuchen Sie die Artikel-IDs der Microsoft Knowledge Base (KB) (beispielsweise KB3216916).
9. Stellen Sie sicher, dass der Wert unter Product (Produkt) dem Ihres verwalteten Knotens entspricht, und wählen Sie den entsprechenden Title (Titel) aus. Ein neues Fenser Details aktualisieren wird geöffnet.
10. In der Registerkarte Übersicht müssen Klassifizierung und Schweregrad des MSRC der Patch-Baseline-Konfiguration entsprechen, die Sie zuvor gefunden haben.

Ursache 2: Das Patch wurde ersetzt.

Lösung 2: Führen Sie die folgenden Schritte aus, um zu überprüfen, ob dies der Fall ist.

1. Rufen Sie den Microsoft Update Catalog auf.
2. Durchsuchen Sie die Artikel-IDs der Microsoft Knowledge Base (KB) (beispielsweise KB3216916).
3. Stellen Sie sicher, dass der Wert unter Product (Produkt) dem Ihres verwalteten Knotens entspricht, und wählen Sie den entsprechenden Title (Titel) aus. Ein neues Fenser Details aktualisieren wird geöffnet.
4. Gehen Sie zur Registerkarte Paketdetails. Suchen Sie nach einem Eintrag unter dem Header Dieses Update wurde durch die folgenden Updates ersetzt:.

Ursache 3: Dasselbe Patch hat möglicherweise unterschiedliche KB-Nummern, da die WSUS- und Windows-Online-Updates von Microsoft als unabhängige Versionskanäle behandelt werden.

Lösung 3: Überprüfen Sie die Berechtigung des Patches. Wenn das Paket unter WSUS nicht verfügbar ist, installieren Sie OS Build 14393.3115. Wenn das Paket für alle Betriebssystem-Builds verfügbar ist, installieren Sie OS-Builds 18362.1256 und 18363.1256.

Kontaktaufnahme mit AWS Support

Wenn Sie Problembehandlungs-Lösungen in diesem Abschnitt oder im bschnitt zu Systems-Manager-Problemen in AWS re:Post nicht finden können und einen Developer-, Business- oder Enterprise- AWS Support -Plan haben, können Sie unter AWS Support einen technischen Supportfall erstellen.

Sammeln Sie die folgenden Gegenstände AWS Support, bevor Sie Kontakt aufnehmen:

- SSM-Agent-Protokolle
- Run Command-Befehls-ID, Wartungsfenster-ID oder Automatisierungsausführungs-ID
- Sammeln Sie für von Windows Server verwaltete Knoten auch Folgendes:
  - `%PROGRAMDATA%\Amazon\PatchBaselineOperations\Logs`, wie auf der Windows-Registerkarte von Wie Patches installiert werden beschrieben
  - Windows Update-Protokolle: Für Windows Server 2012 R2 und älter verwenden Sie `%windir%/WindowsUpdate.log`. Führen Sie für Windows Server 2016 und neuere Versionen zuerst den PowerShell Befehl aus, `Get-WindowsUpdateLog`bevor Sie `%windir%/WindowsUpdate.log`
- Sammeln Sie für Linux-verwaltete Knoten auch Folgendes:
  - Der Inhalt des Verzeichnisses `/var/lib/amazon/ssm/`*`instance-id`*`/document/orchestration/`*`Run-Command-execution-id`*`/awsrunShellScript/PatchLinux`

AWS Systems Manager Distributor

Distributor, eine Funktion von AWS Systems Manager, unterstützt Sie beim Verpacken und Veröffentlichen von Software auf AWS Systems Manager verwalteten Knoten. Sie können Ihre eigene Software verpacken und veröffentlichen oder verwenden, Distributor um AWSvon bereitgestellte Agentensoftwarepakete wie AmazonCloudWatchAgentoder Pakete von Drittanbietern wie Trend Micro zu finden und zu veröffentlichen. Durch die Veröffentlichung eines Pakets werden bestimmte Versionen des Paketdokuments für verwaltete Knoten angekündigt, die Sie mithilfe der Knoten-IDs , AWS-Konto IDs , Tags oder eines identifizieren AWS-Region. Um mit Distributor zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Distributor aus.

Nachdem Sie in Distributor ein Paket erstellt haben, können Sie das Paket auf eine der folgenden Weisen installieren:

- Einmalig mithilfe von AWS Systems Manager Run Command
- Anhand eines Zeitplans mithilfe von AWS Systems Manager State Manager

> **Important**
>
> Pakete, die von Drittanbietern vertrieben werden, werden nicht von verwaltet AWS und vom Anbieter des Pakets veröffentlicht. Wir empfehlen Ihnen, zusätzliche gebührenpflichtige -Prüfungen durchzuführen, um die Einhaltung Ihrer internen Sicherheitskontrollen sicherzustellen. Sicherheit ist eine geteilte Verantwortung zwischen AWS und Ihnen. Dies wird als Modell der geteilten Verantwortung beschrieben. Weitere Informationen hierzu finden Sie in Modell der geteilten Verantwortung.

Welche Vorteile bietet Distributor meiner Organisation?

Distributor bietet die folgenden Vorteile:

- Ein Paket, viele Plattformen

  Wenn Sie ein Paket in Distributor erstellen, erstellt das System ein AWS Systems Manager -Dokument (SSM-Dokument). Sie können ZIP-Dateien an dieses Dokument anfügen. Wenn Sie Distributor ausführen, verarbeitet das System die Anweisungen im SSM-Dokument und installiert das Softwarepaket in der ZIP-Datei auf den angegebenen Zielen. Distributor unterstützt mehrere Betriebssysteme, darunter Windows, Ubuntu Server, Debian Server und Red Hat Enterprise Linux.

Weitere Informationen zu unterstützten Plattformen finden Sie unter Unterstützte Paketplattformen und -architekturen.

- Kontrolle über den Paketzugriff über mehrere Gruppen verwalteter Instances hinweg

  Sie können Run Command oder State Manager verwenden, um zu steuern, welche Ihrer verwalteten Knoten ein Paket erhalten und welche Version dieses Paket haben soll. Run Command und State Manager sind Funktionen von AWS Systems Manager. Verwaltete Knoten können nach Instance- oder Geräte-IDs , AWS-Konto Nummern, Tags oder gruppiert werden AWS-Regionen. Mithilfe von State Manager-Zuordnungen können Sie verschiedene Versionen eines Pakets für verschiedene Gruppen von Instances bereitstellen.

- Viele AWS Agentenpakete enthalten und einsatzbereit

  Distributor enthält viele AWS Agentenpakete, die Sie auf verwalteten Knoten bereitstellen können. Suchen Sie auf der Distributor `Packages`-Listenseite nach Paketen, die von `Amazon` veröffentlicht werden. Beispiele hierfür sind `AmazonCloudWatchAgent` und `AWSPVDriver`.

- Automatische Bereitstellung

  Um Ihre Umgebung auf dem aktuellen Stand zu halten, verwenden Sie State Manager, um Pakete für die automatische Bereitstellung auf ausgesuchten verwalteten Knoten zu planen, wenn diese Maschinen zum ersten Mal gestartet werden.

An wen richtet sich Distributor?

- Jeder AWS Kunde, der neue Softwarepakete erstellen oder vorhandene bereitstellen möchte, einschließlich AWS veröffentlichter Pakete, auf mehreren von Systems Manager verwalteten Knoten gleichzeitig.
- Softwareentwickler, die Softwarepakete erstellen.
- Administratoren, die dafür verantwortlich sind, von Systems Manager verwaltete Knoten mit den meisten up-to-date Softwarepaketen auf dem neuesten Stand zu halten.

Über welche Features verfügt Distributor?

- Bereitstellung von Paketen auf Windows- ebenso wie auf Linux-Instances

  Mit können DistributorSie Softwarepakete auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances und - AWS IoT Greengrass Core-Geräten für Linux und bereitstellenWindows Server.

Eine Liste der unter den jeweiligen Betriebssystemen unterstützten Instance-Typen finden Sie unter the section called “Unterstützte Paketplattformen und -architekturen”.

> **Note**
>
> Distributor wird auf dem macOS-Betriebssystem unterstützt.

- Einmalige Bereitstellung von Paketen oder nach automatisiertem Zeitplan

  Sie können wählen, ob die Pakete einmalig, nach einem regelmäßigen Zeitplan oder immer dann, wenn die Standardpaketversion auf eine andere umgestellt wird, aktualisiert werden sollen.

- Vollständige Neuinstallation von Paketen oder Durchführen von direkten Aktualisierungen

  Um eine neue Paketversion zu installieren, können Sie die aktuelle Version vollständig deinstallieren und stattdessen eine neue Version installieren oder die aktuelle Version nur entsprechend einem von Ihnen bereitgestellten Aktualisierungsskript mit neuen und aktualisierten Komponenten aktualisieren. Ihre Paketanwendung ist während einer Neuinstallation nicht verfügbar, kann aber während einer direkten Aktualisierung weiterhin verfügbar bleiben. Direkte Aktualisierungen sind besonders nützlich für Anwendungen zur Sicherheitsüberwachung oder andere Szenarien, in denen Sie Anwendungsausfälle vermeiden müssen.

- PowerShellKonsolen-, CLI- und SDK-Zugriff auf -DistributorFunktionen

  Sie können mit arbeiten, Distributor indem Sie die Systems Manager-Konsole AWS Tools for PowerShell, AWS Command Line Interface (AWS CLI), oder das AWS SDK Ihrer Wahl verwenden.

- IAM-Zugriffskontrolle

  Mit AWS Identity and Access Management (IAM)-Richtlinien können Sie steuern, welche Mitglieder Ihrer Organisation Pakete oder Paketversionen erstellen, aktualisieren, bereitstellen oder löschen können. Beispiel: Sie möchten einem Administrator Berechtigungen zum Bereitstellen von Paketen gewähren, nicht jedoch zum Ändern von Paketen oder zum Erstellen neuer Paketversionen.

- Support für Protokollierungs- und Prüfungsfunktionen

  Sie können Distributor Benutzeraktionen in Ihrem AWS-Konto durch Integration mit anderen prüfen und protokollieren AWS-Services. Weitere Informationen finden Sie unter Prüfen und Protokollieren von Distributor-Aktivitäten.

Was ist ein Paket?

Ein Paket ist eine Sammlung installierbarer Software oder Komponenten. Beispiele hierfür sind:

- Eine ZIP-Datei mit Software pro Ziel-Betriebssystemplattform. Jede ZIP-Datei muss Folgendes enthalten:
  - Ein install und ein -uninstallSkript. Windows Server-basierte verwaltete Knoten erfordern PowerShell Skripts (Skripts mit den Namen `install.ps1` und `uninstall.ps1`). Linux-basierte verwaltete Knoten erfordern Shell-Skripts (Skripts mit den Namen `install.sh` und `uninstall.sh`). AWS Systems Manager SSM Agent liest und führt die Anweisungen in den uninstallSkripten install und aus.
  - Eine ausführbare Datei. Diese muss SSM Agent finden, um das Paket auf den anvisierten verwalteten Knoten installieren zu können.
- Eine Manifestdatei im JSON-Format, die den Paketinhalt beschreibt. Das Manifest ist nicht in der ZIP-Datei enthalten, aber im selben Amazon Simple Storage Service (Amazon S3)-Bucket gespeichert wie die ZIP-Dateien, aus denen das Paket besteht. Das Manifest identifiziert die Paketversion und ordnet die ZIP-Dateien im Paket den Attributen des anvisierten verwalteten Knotens zu (z. B. die Version oder Architektur des Betriebssystems). Informationen zum Erstellen des Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.

Wenn Sie Einfache Paketerstellung in der Distributor-Konsole wählen, generiert Distributor die Installations- und Deinstallationsskripts, sowie die Datei-Hashes und das Manifest des JSON-Pakets, basierend auf dem Dateinamen der ausführbaren Software sowie den Zielplattformen und -Architekturen.

Unterstützte Paketplattformen und -architekturen

Sie können Distributor verwenden, um Pakete auf den folgenden Plattformen verwalteter Knoten von Systems Manager zu veröffentlichen. Ein Versionswert muss genau mit der Version des Betriebssystems-Amazon Machine Image (AMI) übereinstimmen. Weitere Informationen zum Ermitteln dieser Version finden Sie in Schritt 4 unter Schritt 2: Erstellen des JSON-Paketmanifests.

> **Note**
>
> Systems Manager unterstützt nicht alle der folgenden Betriebssysteme für - AWS IoT Greengrass Core-Geräte. Weitere Informationen finden Sie unter Einrichten von AWS IoT Greengrass -Core-Geräten im AWS IoT Greengrass Version 2 -Entwicklerhandbuch.

Plattform	Codewert in der Manifestdatei	Architektur
Windows Server	`windows`	`x86_64` oder `386`
Debian Server	`debian`	`x86_64` oder `386`
Ubuntu Server	`ubuntu`	`x86_64` oder `386` `arm64` (Ubuntu Server 16 und neuer, A1-Instance-Typen)
Red Hat Enterprise Linux (RHEL)	`redhat`	`x86_64` oder `386` `arm64` (RHEL 7.6 und neuer, A1-Instance-Typen)
CentOS	`centos`	`x86_64` oder `386`
Amazon Linux 1, Amazon Linux 2 und Amazon Linux 2023	`amazon`	`x86_64` oder `386` `arm64` (Amazon Linux 2 und AL2023, A1-Instance-Typen)
SUSE Linux Enterprise Server (SLES)	`suse`	`x86_64` oder `386`
openSUSE	`opensuse`	`x86_64` oder `386`
openSUSE Leap	`opensuseleap`	`x86_64` oder `386`
Oracle Linux	`oracle`	`x86_64`

Themen

- Einrichten von Distributor
- Arbeiten mit Distributor
- Prüfen und Protokollieren von Distributor-Aktivitäten
- Fehlerbehebung für AWS Systems ManagerDistributor

Einrichten von Distributor

Bevor Sie Distributor, eine Funktion von AWS Systems Manager, zum Erstellen, Verwalten und Bereitstellen von Softwarepaketen verwenden, führen Sie die folgenden Schritte aus.

Themen

- Schritt 1: Erfüllen der Distributor-Voraussetzungen
- Schritt 2: Überprüfen oder Erstellen eines IAM-Instance-Profils mit Distributor-Berechtigungen
- Schritt 3: Kontrollieren des Benutzerzugriffs auf Pakete
- Schritt 4: Erstellen oder Auswählen eines Amazon S3-Buckets

Schritt 1: Erfüllen der Distributor-Voraussetzungen

Bevor Sie Distributor verwenden, eine Funktion von AWS Systems Manager, stellen Sie sicher, dass Ihre Umgebung den folgenden Anforderungen entspricht:

Distributor-Voraussetzungen

Anforderung	Beschreibung
SSM Agent	Auf den verwalteten Knoten, auf denen Pakete bereitgestellt bzw. aus denen Pakete entfernt werden sollen, muss AWS Systems Manager SSM Agent Version 2.3.274.0 oder höher installiert sein. Informationen zum Installieren oder Aktualisieren von SSM Agent finden Sie unter Arbeiten mit SSM Agent.
AWS CLI	(Optional) Um die AWS Command Line Interface (AWS CLI) anstelle der Systems Manager-Konsole zum Erstellen und Verwalten von Paketen zu verwenden, installieren Sie die neueste Version der AWS CLI auf Ihrem lokalen Computer.

Anforderung	Beschreibung
	Weitere Informationen zum Installieren oder Upgraden der CLI finden Sie unter Installieren der AWS Command Line Interface um AWS Command Line Interface-Benutzerhandbuch.
AWS Tools for PowerShell	(Optional) Um die Tools for PowerShell anstelle der Systems Manager-Konsole zum Erstellen und Verwalten von Paketen zu verwenden, installieren Sie die neueste Version der Tools for PowerShell auf Ihrem lokalen Computer. Weitere Informationen zum Installieren oder Aktualisieren der Tools für PowerShel l finden Sie unter Einrichten der AWS Tools for Windows PowerShell oder AWS Tools for PowerShell Core im AWS Tools for Windows PowerShell-Benutzerhandbuch.

> **Note**
>
> Systems Manager unterstützt nicht die Verteilung von Paketen an von Oracle Linux verwaltete Knoten mit Distributor.

Schritt 2: Überprüfen oder Erstellen eines IAM-Instance-Profils mit Distributor-Berechtigungen

Hat standardmäßig AWS Systems Manager keine Berechtigung, Aktionen auf Ihren Instances durchzuführen. Sie müssen den Zugriff mithilfe eines AWS Identity and Access Management (IAM-) Instanzprofils gewähren. Ein Instance-Profil ist ein Container, der Informationen zur IAM-Rolle beim Start an eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance übergibt. Diese Anforderung gilt für Berechtigungen für alle Systems Manager Manager-FunktionenDistributor, nicht nur für, welche Fähigkeit von AWS Systems Manager.

> **Note**
>
> Wenn Sie Ihre Edge-Geräte für die Ausführung der AWS IoT Greengrass Core-Software und konfigurierenSSM Agent, geben Sie eine IAM-Servicerolle an, die es Systems Manager ermöglicht, Aktionen darauf auszuführen. Sie müssen keine verwalteten Edge-Geräte mit einem Instance-Profil konfigurieren.

Wenn Sie bereits andere Systems Manager-Funktionen verwenden, wie z. B. Run Command und State Manager, ist ein Instance-Profil mit den erforderlichen Berechtigungen für Distributor bereits Ihren Instances zugeordnet. Die einfachste Methode, um sicherzustellen, dass Sie über die erforderlichen Berechtigungen zur Ausführung von Distributor Aufgaben verfügen, besteht darin, Ihrem Instance-Profil die ManagedInstanceAmazonSSM-Core-Richtlinie anzuhängen. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

Schritt 3: Kontrollieren des Benutzerzugriffs auf Pakete

Mit AWS Identity and Access Management-(IAM)-Richtlinien können Sie steuern, wer Pakete erstellen, bereitstellen und verwalten kann. Sie können auch steuern, welche Run Command- und State Manager-API-Operationen auf verwalteten Knoten ausgeführt werden können. Wie Distributor sind Run Command und State Manager ebenfalls Funktionen von AWS Systems Manager.

ARN-Format

Benutzerdefinierte Pakete sind Dokument-Amazon Resource Names (ARNs) zugeordnet und haben das folgende Format.

```
arn:aws:ssm:region:account-id:document/document-name
```

Im Folgenden wird ein Beispiel gezeigt.

```
arn:aws:ssm:us-west-1:123456789012:document/ExampleDocumentName
```

Sie können zwei von AWS bereitgestellte IAM-Standardrichtlinien verwenden, eine für Endbenutzer und eine für Administratoren, um Berechtigungen für Distributor-Aktivitäten zu erteilen. Sie können auch benutzerdefinierte IAM-Richtlinien erstellen, die an Ihre Berechtigungsanforderungen angepasst sind.

Weitere Informationen zur Verwendung von Variablen in IAM-Richtlinien finden Sie unter IAM-Richtlinienelemente: Variablen.

Informationen zum Erstellen von Richtlinien und zum Anfügen an Benutzer oder Gruppen finden Sie unter Erstellen von IAM-Richtlinien und Hinzufügen und Entfernen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Schritt 4: Erstellen oder Auswählen eines Amazon S3-Buckets

Wenn Sie ein Paket erstellen, indem Sie die in der AWS Systems Manager-Konsole den Workflow Simple (Einfach) auswählen, wählen Sie einen vorhandenen Amazon Simple Storage Service (Amazon S3)-Bucket aus, an den Distributor Ihre Software hochlädt. Distributor ist eine Funktion von AWS Systems Manager. Wenn Sie den Workflow Advanced (Erweitert) auswählen, müssen Sie Ihre Software oder Komponenten als ZIP-Dateien in einen Amazon-S3-Bucket hochladen, bevor Sie beginnen. Unabhängig davon, ob Sie ein Paket mit dem Workflow Simple (Einfach) oder Advanced (Erweitert) in der Konsole erstellen, oder aber ob Sie die API verwenden, Sie benötigen einen Amazon-S3-Bucket, bevor Sie mit der Erstellung Ihres Paket beginnen. Bei der Paketerstellung kopiert Distributor Ihre installierbare Software und die Komponenten aus diesem Bucket in einen internen Systems Manager-Speicher. Da die Komponenten in einen internen Speicher kopiert werden, können Sie Ihren Amazon-S3-Bucket löschen oder wiederverwenden, wenn die Paketerstellung abgeschlossen ist.

Weitere Informationen zur Erstellung eines Buckets finden Sie unter Erstellen eines Buckets im Amazon Simple Storage Service Getting Started Guide. Weitere Informationen zum Ausführen eines AWS CLI-Befehls zum Erstellen eines Buckets finden Sie unter mb in der AWS CLI-Befehlsreferenz.

Arbeiten mit Distributor

Sie können die AWS Systems Manager-Konsole, die AWS-Befehlszeilen-Tools (AWS CLI und AWS Tools for PowerShell) und AWS-SDKs zum Hinzufügen, Verwalten oder Bereitstellen von Paketen in Distributor verwenden. Distributor ist eine Funktion von AWS Systems Manager. Vor dem Hinzufügen eines Pakets zu Distributor:

- Erstellen und zippen Sie die zu installierbaren Komponenten.
- (Optional) Erstellen Sie eine JSON-Manifestdatei für das Paket. Dies ist nicht erforderlich, um den Prozess der einfachen Paketerstellung in der Distributor-Konsole zu nutzen. Bei der einfachen Paketerstellung wird die JSON-Manifestdatei automatisch generiert.

Sie können zum Erstellen der Manifestdatei die AWS Systems Manager-Konsole oder einen Text- oder JSON-Editor verwenden.

- Halten Sie einen Amazon Simple Storage Service (Amazon S3)-Bucket bereit, um Ihre installierbaren Komponenten oder Software zu speichern. Wenn Sie die den Advanced (Erweitert)-Workflow zur Paketerstellung verwenden, laden Sie Ihre Komponenten an den Amazon-S3-Bucket herunter, bevor Sie beginnen.

> **Note**
>
> Sie können diesen Bucket nach Abschluss der Erstellung Ihres Pakets löschen oder anderweitig verwenden, weil Distributor im Rahmen des Paketerstellungsprozesses die Paketinhalte in einen internen Systems Manager-Bucket verschiebt.

Von AWS veröffentlichte Pakete sind bereits verpackt und können sofort bereitgestellt werden. Informationen zum Bereitstellen eines von AWS veröffentlichten Pakets an verwalteten Knoten finden Sie unter Installieren oder Aktualisieren von Paketen.

Sie können Distributor-Pakete zwischen AWS-Konten tauschen. Wenn Sie ein Paket verwenden, das von einem anderen Konto in AWS CLI-Befehlen geteilt wird, verwenden Sie das Paket Amazon Resource Name (ARN) anstelle des Paketnamens.

Themen

- Pakete anzeigen
- Erstellen eines Pakets
- Bearbeiten von Paketberechtigungen (Konsole)
- Bearbeiten von Paket-Tags (Konsole)
- Hinzufügen einer Paketversion zu Distributor
- Installieren oder Aktualisieren von Paketen
- Deinstallieren eines Pakets
- Löschen eines Pakets

Pakete anzeigen

Um Pakete anzuzeigen, die für die Installation verfügbar sind, können Sie die AWS Systems Manager-Konsole oder Ihr bevorzugtes AWS-Befehlszeilen-Tool verwenden. Distributor ist eine Funktion von AWS Systems Manager. Um auf Distributor zuzugreifen, öffnen Sie die AWS Systems Manager-Konsole und wählen Sie im linken Navigationsbereich Distributor aus. Sie werden alle Pakete sehen, die Ihnen zur Verfügung stehen.

Im folgenden Abschnitt wird beschrieben, wie Sie Ihre Distributor-Pakete mithilfe Ihrer bevorzugten Befehlszeilen-Tools anzeigen.

Anzeigen von Paketen (Befehlszeile)

Dieser Abschnitt enthält Informationen dazu, wie Sie Ihr bevorzugtes Befehlszeilen-Tool verwenden können, um Distributor-Pakete mit den bereitgestellten Befehlen anzuzeigen.

Linux & macOS

Anzeigen von Paketen mit AWS CLI unter Linux

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, mit der Ausnahme freigegebener Pakete.

```
aws ssm list-documents \
    --filters Key=DocumentType,Values=Package
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Amazon gehören.

```
aws ssm list-documents \
    --filters Key=DocumentType,Values=Package Key=Owner,Values=Amazon
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Drittanbietern gehören.

```
aws ssm list-documents \
    --filters Key=DocumentType,Values=Package Key=Owner,Values=ThirdParty
```

Windows

Anzeigen von Paketen mit AWS CLI unter Windows

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, mit der Ausnahme freigegebener Pakete.

```
aws ssm list-documents ^
    --filters Key=DocumentType,Values=Package
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Amazon gehören.

```
aws ssm list-documents ^
    --filters Key=DocumentType,Values=Package Key=Owner,Values=Amazon
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Drittanbietern gehören.

```
aws ssm list-documents ^
    --filters Key=DocumentType,Values=Package Key=Owner,Values=ThirdParty
```

PowerShell

Anzeigen von Packages mit Tools for PowerShell

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, mit der Ausnahme freigegebener Pakete.

```
$filter = New-Object Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "DocumentType"
$filter.Values = "Package"

Get-SSMDocumentList `
    -Filters @($filter)
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Amazon gehören.

```
$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)
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Drittanbietern gehören.

```
$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)
```

Erstellen eines Pakets

Um ein Paket zu erstellen, bereiten Sie Ihre installierbare Software oder Komponenten vor, immer eine Datei pro Betriebssystem-Plattform. Zur Erstellung eines Pakets ist mindestens eine Datei erforderlich.

Manchmal verwenden unterschiedliche Plattformen dieselbe Datei. Alle Ihrem Paket hinzugefügten Dateien müssen jedoch im Abschnitt `Files` des Manifests aufgelistet sein. Wenn Sie ein Paket in der Konsole über den einfachen Workflow erstellen, wird das Manifest automatisch generiert. Die maximale Anzahl von Dateien, die Sie einem einzelnen Dokument anfügen können, beträgt 20. Die maximale Größe der einzelnen Dateien beträgt 1 GB. Weitere Informationen zu unterstützten Plattformen finden Sie unter Unterstützte Paketplattformen und -architekturen.

Wenn Sie ein neues Paket erstellen, erstellt das System ein neues SSM-Dokument. Mit diesem Dokument können Sie das Paket an verwaltete Knoten bereitstellen.

Nur zu Demonstrationszwecken steht Ihnen ein Beispielpaket, ExamplePackage.zip, zum Herunterladen von unserer Website zur Verfügung. Das Beispielpaket enthält ein vollständiges

JSON-Manifest und drei .zip-Dateien mit Installationsprogrammen für Version 7.0.0. PowerShell Die Installations- und Deinstallationsskripts enthalten keine gültigen Befehle. Wenn Sie ein Paket im Workflow Advanced (Erweitert) erstellen, müssen Sie alle installierbaren Softwaredateien und Skripts in einer ZIP-Datei komprimieren, beim Workflow Simple (Einfach) ist es jedoch nicht notwendig, installierbare Komponenten zu zippen.

Themen

- Erstellen eines Pakets (einfach)
- Erstellen eines Pakets (erweitert)

Erstellen eines Pakets (einfach)

In diesem Abschnitt wird beschrieben, wie Sie ein Paket erstellen, Distributor indem Sie in der Konsole den Workflow Einfache Paketerstellung auswählen. Distributor Distributorist eine Fähigkeit von AWS Systems Manager. Um ein Paket zu erstellen, bereiten Sie Ihre zu installierenden Komponenten vor, eine Datei pro Betriebssystemplattform. Zur Erstellung eines Pakets ist mindestens eine Datei erforderlich. Bei der einfachen Paketerstellung werden die Installations- und Deinstallationsskripts generiert, sowie die Datei-Hashes und eine Manifest-Datei im JSON-Format. Der Simple (Einfach)-Workflow übernimmt das Hochladen und Komprimieren Ihrer installierbaren Dateien sowie das Erstellen eines neuen Pakets und des zugehörigen SSM-Dokuments. Weitere Informationen zu unterstützten Plattformen finden Sie unter Unterstützte Paketplattformen und -architekturen.

Wenn Sie die Methode „Simple (Einfach)“ verwenden, um ein Paket zu erstellen, erstellt Distributor die `install`- und `uninstall`-Skripts für Sie. Wenn Sie jedoch ein Paket für ein direktes Update erstellen, müssen Sie Ihren eigenen `update`-Skript-Inhalt in der Registerkarte Update script bereitstellen.. Wenn Sie Eingabebefehle für ein `update`-Skript hinzufügen, schließt Distributor dieses Skript zusammen mit den `install`- und `uninstall`-Skripts in das für Sie erstellte ZIP-Paket ein.

> **Note**
>
> Verwenden Sie die Aktualisierungsoption `In-place` zum Hinzufügen neuer oder aktualisierter Dateien einer vorhandenen Paketinstallation, ohne die zugehörige Anwendung offline zu schalten.

So erstellen Sie ein Paket (einfaches Verfahren)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite die Option Create package (Paket erstellen) aus und wählen Sie dann die Option Simple (Einfach).
4. Geben Sie auf der Seite Create package (Paket erstellen) einen Namen für Ihr Paket ein. Paketnamen können Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten. Der Name sollte allgemein genug sein, um auf alle Versionen der Paketanhänge angewendet werden zu können, jedoch spezifisch genug, um den Zweck des Pakets zu identifizieren.
5. (Optional) Geben Sie unter Version name (Versionsname) einen Versionsnamen ein. Versionsnamen dürfen maximal 512 Zeichen lang sein und dürfen keine Sonderzeichen enthalten.
6. Wählen Sie unter Location (Speicherort) einen Bucket aus. Verwenden Sie dazu den Namen und das Präfix des Buckets oder die Bucket-URL.
7. Wählen Sie unter Upload software (Software hochladen) die Option Add software (Software hinzufügen) aus und navigieren Sie dann zu installierbaren Softwaredateien mit den Erweiterungen `.rpm`, `.msi`, oder `.deb`. Wenn der Dateiname Leerzeichen enthält, schlägt der Upload fehl. Sie können mehrere Softwaredateien in einer einzigen Aktion hochladen.
8. Überprüfen Sie unter Für Target platform (Ziel-Plattform für jede Plattform, ob das Ziel-Betriebssystem für die installierbare Datei korrekt ist. Wenn das angezeigte Betriebssystem nicht korrekt ist, wählen Sie das richtige Betriebssystem aus der Dropdown-Liste aus.

   Beim Workflow Simple (Einfach) zur Paketerstellung sind zusätzliche Schritte erforderlich, wenn Distributor nur eine Datei für mehrere Betriebssysteme verwenden soll, da installierbare Dateien nur einmal hochgeladen werden. Wenn Sie zum Beispiel eine installierbare Softwaredatei mit dem Namen `Logtool_v1.1.1.rpm` hochladen, müssen Sie im Simple-Workflow einige Standardeinstellungen ändern, um als Zielplattform für die Software Amazon Linux- und Ubuntu-Betriebssysteme anzugeben. Führen Sie bei Verwendung mehrerer Zielplattformen einen der folgenden Schritte aus.

   - Verwenden Sie stattdessen den Workflow Advanced (Erweitert), zippen Sie jede installierbare Datei, bevor Sie beginnen, und richten Sie das Manifest manuell so ein, dass eine installierbare Datei für mehrere Betriebssystemplattformen oder -versionen verwendet werden kann. Weitere Informationen finden Sie unter Erstellen eines Pakets (erweitert).

- Bearbeiten Sie die Manifestdatei im Workflow Simple (Einfach) so, dass Ihre ZIP-Datei für mehrere Betriebssystemplattformen oder -versionen verwendet wird. Weitere Informationen zu diesem Verfahren finden Sie am Ende von Schritt 4 in Schritt 2: Erstellen des JSON-Paketmanifests.

9. Stellen Sie unter Platform version (Plattformversion)sicher, dass als Betriebssystem-Plattformversion **_any**, eine Hauptversionsnummer, gefolgt von einem Platzhalter (7.*), angezeigt wird, oder genau die spezifische Betriebssystemversion, die Sie als Plattformversion für Ihre Softwareinstallation verwenden möchten. Weitere Informationen zur Angabe der Betriebssystem-Plattformversion finden Sie unter Schritt 4 in Schritt 2: Erstellen des JSON-Paketmanifests.
10. Wählen Sie unter Architecture (Architektur) für jeden installierbare Datei die richtige Prozessorarchitektur aus der Dropdown-Liste aus. Weitere Informationen zu unterstützten Prozessorarchitekturen finden Sie unter Unterstützte Paketplattformen und -architekturen.
11. (Optional) Erweitern Sie Scripts (Skripts) und überprüfen Sie die Skripts, die Distributor für Ihre installierbare Software generiert hat.
12. (Optional) Um ein Aktualisierungsskript für direkte Aktualisierungen bereitzustellen, erweitern Sie Scripts (Skripts), wählen Sie die Registerkarte Update script (Aktualisierungsskript) aus und geben Sie die Befehle für das Aktualisierungsskript ein.

    Systems Manager generiert keine Aktualisierungsskripts für Sie.
13. Zum Hinzufügen weiterer installierbaren Softwaredateien wählen Sie Add Software (Software hinzufügen). Andernfalls fahren Sie mit dem nächsten Schritt fort.
14. (Optional) Erweitern Sie Manifest und überprüfen Sie das JSON-Manifest des Pakets, das Distributor für Ihre installierbare Software generiert hat. Wenn Sie Informationen über Ihre Software geändert haben, nachdem Sie mit dieser Prozedur begonnen haben, beispielsweise die Plattformversion oder die Zielplattform, wählen Sie Generate Manifest (Manifest erzeugen), um das Paketmanifest zu aktualisieren.

    Sie können das Manifest manuell bearbeiten, wenn Sie möchten, dass für eine installierbare Software mehr als ein Betriebssystem als Ziel festgelegt wird, wie in Schritt 8 beschrieben. Weitere Informationen zum Bearbeiten des Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.
15. Wählen Sie Create package (Paket erstellen) aus.

Warten Sie, bis Distributor das Hochladen Ihrer Software und das Erstellen Ihres Pakets abgeschlossen hat. Distributor zeigt den Uploadstatus für jede installierbare Datei einzeln an. Abhängig von der Anzahl und Größe der Pakete, die Sie hinzufügen, kann dies einige Minuten dauern. Distributor leitet Sie automatisch auf die Seite Package details (Paketdetails) für das neue Paket weiter, Sie können diese Seite aber jederzeit selbst öffnen, nachdem die Software hochgeladen ist. Auf der Seite Package details (Paketdetails) werden erst dann alle Informationen zu Ihrem Paket angezeigt, wenn Distributor den Paketerstellungsprozess abgeschlossen hat. Um den Upload- und Paketerstellungsprozess abzubrechen, wählen Sie Cancel (Abbrechen).

Wenn Distributor keine installierbaren Softwaredateien hochladen kann, wird eine Fehlermeldung Upload failed (Upload fehlgeschlagen) angezeigt. Um den Uploadversuch zu wiederholen, wählen Sie Retry Upload (Uploadversuch wiederholen). Weitere Informationen zur Fehlerbehebung bei der Paketerstellung finden Sie unter Fehlerbehebung für AWS Systems ManagerDistributor.

Erstellen eines Pakets (erweitert)

In diesem Abschnitt erfahren Sie, wie fortgeschrittene Benutzer ein Paket in Distributor erstellen können, nachdem sie die installierbaren Komponenten mit den Skripts zur Installation und Deinstallation, sowie eine JSON-Manifestdatei als ZIP-Archiv an einen Amazon S3-Bucket hochgeladen haben.

Um ein Paket zu erstellen, bereiten Sie Ihre ZIP-Dateien mit den zu installierenden Komponenten vor (eine ZIP-Datei pro Betriebssystemplattform). Zur Erstellung eines Pakets ist mindestens eine ZIP-Datei erforderlich. Erstellen Sie als Nächstes ein JSON-Manifest. Das Manifest enthält Verweise auf Ihre Paketcodedateien. Wenn Sie die erforderlichen Codedateien zu einem Ordner oder Verzeichnis hinzugefügt haben und das Manifest mit den korrekten Werten ausgefüllt ist, laden Sie Ihr Paket an einen S3-Bucket hoch.

Ein Beispielpaket, ExamplePackage.zip, steht Ihnen auf unserer Website zum Herunterladen zur Verfügung. Das Beispielpaket enthält ein fertiges JSON-Manifest und drei ZIP-Dateien.

Themen

- Schritt 1: Erstellen der ZIP-Dateien
- Schritt 2: Erstellen des JSON-Paketmanifests
- Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket
- Schritt 4: Hinzufügen eines Pakets zu Distributor

Schritt 1: Erstellen der ZIP-Dateien

Die Grundlage Ihres Pakets ist mindestens eine ZIP-Datei mit Softwaredateien oder zu installierenden Komponenten. Ein Paket enthält eine ZIP-Datei pro Betriebssystem, das Sie unterstützen möchten, es sei denn, eine ZIP-Datei kann auf mehreren Betriebssystemen installiert werden. Beispielsweise können Red Hat Enterprise Linux- und Amazon Linux-Instances in der Regel dieselben ausführbaren RPM-Dateien ausführen. Daher müssen Sie Ihrem Paket nur eine ZIP-Datei anfügen, um beide Betriebssysteme zu unterstützen.

Erforderliche Dateien

Die folgenden Elemente müssen in jeder ZIP-Datei enthalten sein:

- Ein install und ein uninstall Skript. Windows Serverbasierte verwaltete Knoten benötigen PowerShell Skripten (Skripten mit dem Namen `install.ps1` und`uninstall.ps1`). Linux-basierte verwaltete Knoten benötigen Shell-Skripts (Skripts mit dem Namen `install.sh` und `uninstall.sh`). SSM Agent führt die Anweisungen in den install- und uninstall-Skripts aus.

  Ihre Installationsskripts können beispielsweise ein Installationsprogramm ausführen (z. B. eine RPM- oder MSI-Datei), Dateien kopieren oder Konfigurationseinstellungen festlegen.
- Eine ausführbare Datei, Installationsprogrammpakete (.rpm, .deb, .msi usw.), weitere Skripts oder Konfigurationsdateien.

Optionale Dateien

Die folgenden Elemente können optional in jeder ZIP-Datei enthalten sein:

- Ein update-Skript. Die Angabe eines Aktualisierungsskripts ermöglicht es Ihnen, die Option `In-place update` zum Installieren eines Pakets zu verwenden. Wenn Sie einer vorhandenen Paketinstallation neue oder aktualisierte Dateien hinzufügen möchten, wird die Paketanwendung bei dieser `In-place update` Option nicht offline geschaltet, während das Update ausgeführt wird. Windows Serverbasierte verwaltete Knoten benötigen ein PowerShell Skript (mit dem Namen`update.ps1`). Linux-basierte verwaltete Knoten benötigen ein Shell-Skript (Skript mit dem Namen `update.sh`). SSM Agent führt die Anweisungen im update-Skript aus.

Weitere Informationen zum Installieren oder Aktualisieren von Paketen finden Sie unter Installieren oder Aktualisieren von Paketen.

Für Beispiele für ZIP-Dateien, einschließlich Beispiel install - und uninstall Skripten, laden Sie das Beispielpaket (ExamplePackage.zip) herunter.

Schritt 2: Erstellen des JSON-Paketmanifests

Nachdem Sie die zu installierenden Dateien vorbereitet und gezippt haben, erstellen Sie ein JSON-Manifest. Im Folgenden finden Sie eine Vorlage. Die einzelnen Teile der Manifestvorlage werden im Verfahren in diesem Abschnitt beschrieben. Sie können einen JSON-Editor verwenden, um dieses Manifest in einer eigenen Datei zu erstellen. Alternativ können Sie das Manifest in der AWS Systems Manager Konsole erstellen, wenn Sie ein Paket erstellen.

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

So erstellen Sie ein JSON-Paket-Manifest

1. Fügen Sie Ihrem Schema die Schemaversion hinzu. In dieser Version ist die Schemaversion stets `2.0`.

   ```
   { "schemaVersion": "2.0",
   ```

2. Fügen Sie Ihrem Manifest eine benutzerdefinierte Paketversion hinzu. Dies ist auch der Wert in Version name (Versionsname), den Sie angeben, wenn Sie Ihr Paket zu Distributor hinzufügen. Er wird Teil des AWS Systems Manager -Dokuments, das Distributor erstellt, wenn Sie Ihr Paket hinzufügen. Sie stellen diesen Wert auch als Eingabewert im Dokument `AWS-ConfigureAWSPackage` bereit, um eine andere als die aktuelle Version des Pakets zu installieren. Ein `version`-Wert kann Buchstaben, Zahlen, Unterstriche, Bindestriche und Punkte enthalten. Er darf jedoch höchstens 128 Zeichen enthalten. Sie sollten eine von Menschen lesbare Paketversion verwenden, um bei Bereitstellungen die Angabe der genauen Paketversionen für Sie und andere Administratoren einfacher zu machen. Im Folgenden wird ein Beispiel gezeigt.

   ```
   "version": "1.0.1",
   ```

3. (Optional) Fügen Sie den Namen des Herausgebers hinzu. Im Folgenden wird ein Beispiel gezeigt.

   ```
   "publisher": "MyOrganization",
   ```

4. Fügen Sie Pakete hinzu. Der Abschnitt `"packages"` beschreibt die von den ZIP-Dateien in Ihrem Paket unterstützten Plattformen, Versionen und Architekturen. Weitere Informationen finden Sie unter Unterstützte Paketplattformen und -architekturen.

   Beim Wert in *`Plattformversion`* kann es sich um den Platzhalterwert `_any` handeln. Sie verwenden den Platzhalterwert, um anzugeben, dass eine ZIP-Datei eine beliebige Version der

Plattform unterstützt. Sie können auch eine Hauptversion und gefolgt von einem Platzhalter angeben, sodass alle Nebenversionen unterstützt werden, z. B. 7.*. Wenn Sie einen Wert für *Plattformversion* für eine bestimmte Betriebssystemversion angeben, stellen Sie sicher, dass er genau mit der Version des Betriebssystem-AMI übereinstimmt, auf das Sie abzielen. Im Folgenden werden Ressourcen empfohlen, mit denen Sie den richtigen Wert für das Betriebssystem ermitteln können.

- Auf einem Windows Server-basierten verwalteten Knoten ist die Version in Form von Windows Management Instrumentation (WMI)-Daten verfügbar. Sie können den folgenden Befehl in der Eingabeaufforderung ausführen, um Versionsinformationen zu erhalten. Anschließend müssen Sie die Ergebnisse nach `version` durchsuchen. Dieser Befehl zeigt die Version für Windows Server Nano nicht an. Der Versionswert für Windows Server Nano ist `nano`.

  ```
  wmic OS get /format:list
  ```

- Auf einem Linux-basierten verwalteten Knoten erhalten Sie die Version, indem Sie zunächst nach der Betriebssystemversion scannen (der folgende Befehl). Suchen Sie den Wert von `VERSION_ID`.

  ```
  cat /etc/os-release
  ```

  Wenn die ausgegebene Zeichenfolge nicht die benötigten Informationen enthält, führen Sie den folgenden Befehl aus, um die LSB-Versionsinformationen aus der Datei `/etc/lsb-release` abzurufen, und suchen den Wert von `DISTRIB_RELEASE`.

  ```
  lsb_release -a
  ```

  Wenn diese Methoden nicht zum Erfolg führen, finden Sie die Version in der Regel anhand der verwendeten Distribution. In Debian Server können Sie beispielsweise die Datei `/etc/debian_version` und in Red Hat Enterprise Linux die Datei `/etc/redhat-release` scannen.

  ```
  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"
        }
      }
    }
  }
```

Im Folgenden wird ein Beispiel gezeigt. In diesem Beispiel ist die Betriebssystemplattform `amazon`, die unterstützte Version `2016.09`, die Architektur `x86_64` und die ZIP-Datei, die diese Plattform unterstützt, `test.zip`.

```
{
    "amazon": {
        "2016.09": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    }
},
```

Sie können mit dem Platzhalterwert (`_any`) angeben, dass das Paket alle Versionen des übergeordneten Elements unterstützt. Um beispielsweise anzugeben, dass das Paket in jeder Version von Amazon Linuxs unterstützt wird, sollte Ihre Paketanweisung ähnlich wie folgt aussehen. Sie können den Platzhalter `_any` auf Versions- oder Architekturebene verwenden, um alle Versionen einer Plattform, alle Architekturen in einer Version oder alle Versionen und alle Architekturen einer Plattform zu unterstützen.

```
{
    "amazon": {
        "_any": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    }
},
```

Das folgende Beispiel fügt `_any` hinzu, um zu zeigen, dass das erste Paket `data1.zip` für alle Architekturen von Amazon Linux 2016.09 unterstützt wird. Das zweite Paket (`data2.zip`) wird für alle Versionen von Amazon Linux unterstützt, jedoch nur für verwaltete Knoten mit `x86_64`-Architektur. Sowohl die Version mit `2016.09` als auch die Version mit `_any` sind Einträge unter `amazon`. Es handelt sich um eine einzige Plattform (Amazon Linux), aber verschiedene unterstützte Versionen und Architekturen sowie zugehörige ZIP-Dateien.

```
{
    "amazon": {
        "2016.09": {
            "_any": {
                "file": "data1.zip"
            }
        },
        "_any": {
            "x86_64": {
                "file": "data2.zip"
            }
        }
    }
}
```

Sie können im Abschnitt `"packages"` des Manifests mehrmals auf eine ZIP-Datei verweisen, wenn diese mehrere Plattformen unterstützt. Wenn Ihre ZIP-Datei beispielsweise sowohl Red Hat Enterprise Linux 7.x Versionen als auch Amazon Linux unterstützt, enthält sie wie im folgenden Beispiel gezeigt im Abschnitt `"packages"` zwei Einträge, die auf dieselbe ZIP-Datei verweisen.

```
{
```

```
    "amazon": {
        "2018.03": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    },
    "redhat": {
        "7.*": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    }
},
```

5. Fügen Sie die Liste mit den zu diesem Paket gehörenden ZIP-Dateien aus Schritt 4 hinzu. Für jeden Dateieintrag sind der Dateiname und die Prüfsumme des `sha256`-Hash-Werts erforderlich. Die Prüfsummenwerte im Manifest müssen mit dem `sha256`-Hash-Wert in den gezippten Ressourcen übereinstimmen, um zu verhindern, dass die Paketinstallation fehlschlägt.

   Um die korrekte Prüfsumme aus den zu installierenden Dateien zu erhalten, können Sie die folgenden Befehle ausführen. In Linux führen Sie `shasum -a 256 file-name.zip` oder `openssl dgst -sha256 file-name.zip` aus. Führen Sie unter Windows das `Get-FileHash -Path path-to-.zip-file` Cmdlet in aus. PowerShell

   Der Abschnitt `"files"` des Manifests enthält einen Verweis auf jede ZIP-Datei in Ihrem Paket.

```
"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. Nachdem Sie die Paketinformationen hinzugefügt haben, speichern und schließen Sie die Manifestdatei.

Im Folgenden finden Sie ein Beispiel für ein fertiges Manifest. In diesem Beispiel haben Sie eine ZIP-Datei `NewPackage_LINUX.zip`, die mehrere Plattformen unterstützt, jedoch nur einmal im Abschnitt `"files"` referenziert wird.

```
{
    "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"
            }
```

```
        }
    }
}
```

Beispiel für ein Paket

Ein Beispielpaket, ExamplePackage.zip, steht Ihnen auf unserer Website zum Herunterladen zur Verfügung. Das Beispielpaket enthält ein fertiges JSON-Manifest und drei ZIP-Dateien.

Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket

Bereiten Sie Ihr Paket vor, indem Sie alle ZIP-Dateien in einen Ordner oder ein Verzeichnis kopieren oder verschieben. Ein Paket ist nur gültig, wenn es das von Ihnen in Schritt 2: Erstellen des JSON-Paketmanifests erstellte Manifest und alle ZIP-Dateien enthält, die in der Dateiliste des Manifests angegeben sind.

So laden Sie das Paket und das Manifest in Amazon S3 hoch

1. Kopieren oder verschieben Sie alle von Ihnen im Manifest angegebenen ZIP-Archivdateien in einen Ordner oder ein Verzeichnis. Komprimieren Sie nicht den Ordner oder das Verzeichnis, in das/den Sie die ZIP-Archivdateien und die Manifestdatei verschieben.
2. Erstellen Sie einen Bucket, oder wählen Sie einen vorhandenen Bucket aus. Weitere Informationen finden Sie unter Create a Bucket (Bucket erstellen) im Amazon Simple Storage Service Getting Started Guide (Amazon Simple Storage Service Erste-Schritte-Leitfaden). Weitere Informationen zum Ausführen eines AWS CLI Befehls zum Erstellen eines Buckets finden Sie mbin der AWS CLI Befehlsreferenz.
3. Laden Sie den Ordner oder das Verzeichnis zum Bucket hoch. Anleitungen finden Sie unter Hinzufügen eines Objekts zu einem Bucket im Erste Schritte-Handbuch zu Amazon Simple Storage Service. Wenn Sie Ihr JSON-Manifest in die AWS Systems Manager Konsole einfügen möchten, laden Sie das Manifest nicht hoch. Weitere Informationen zum Ausführen eines AWS CLI Befehls zum Hochladen von Dateien in einen Bucket finden Sie mvin der AWS CLI Befehlsreferenz.
4. Wählen Sie auf der Startseite des Buckets den von Ihnen hochgeladenen Ordner oder das Verzeichnis aus. Wenn Sie Ihre Dateien in einen Unterordner in einem Bucket hochgeladen haben, stellen Sie sicher, dass Sie sich den Namen des Unterordner notieren (wird auch als Präfix bezeichnet). Sie benötigen das Präfix, um Ihr Paket zu Distributor hinzuzufügen.

Schritt 4: Hinzufügen eines Pakets zu Distributor

Sie können die AWS Systems Manager Konsole, AWS Befehlszeilentools (AWS CLI und AWS Tools for PowerShell) oder AWS SDKs verwenden, um ein neues Paket hinzuzufügen. Distributor Wenn Sie ein Paket hinzufügen, fügen Sie ein neues SSM-Dokument hinzu. Mit dem Dokument können Sie das Paket an verwaltete Knoten bereitstellen.

Themen

- Hinzufügen eines Pakets (Konsole)
- Hinzufügen eines Pakets (AWS CLI)

Hinzufügen eines Pakets (Konsole)

Sie können die AWS Systems Manager Konsole verwenden, um ein Paket zu erstellen. Halten Sie den Namen des Buckets bereit, auf den Sie in Ihr Paket in Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket hochgeladen haben.

So fügen Sie Distributor ein Paket hinzu (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählenen Sie auf der Distributor-Startseite die Option Create package (Paket erstellen) und klicken Sie dann auf Advanced (Erweitert).
4. Geben Sie auf der Seite Create package (Paket erstellen) einen Namen für Ihr Paket ein. Paketnamen können Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten. Der Name sollte allgemein genug sein, um auf alle Versionen der Paketanhänge angewendet werden zu können, jedoch spezifisch genug, um den Zweck des Pakets zu identifizieren.
5. Geben Sie unter Version name (Versionsname) den exakten Wert des Eintrags `version` in Ihrer Manifestdatei ein.
6. Wählen Sie unter S3 bucket name (S3-Bucketname) den Namen des Buckets aus, in den Sie Ihre ZIP-Dateien und das Manifest in the section called “Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket” hochgeladen haben.
7. Geben Sie unter S3 key prefix (S3-Schlüsselpräfix) den Unterordner des Buckets ein, in dem Ihre ZIP-Dateien und das Manifest gespeichert sind.

8. Wählen Sie unter Manifest die Option Extract from package (Aus Paket extrahieren) aus, um ein Manifest zu verwenden, das Sie mit Ihren ZIP-Dateien in den Amazon S3-Bucket hochgeladen haben.

   (Optional) Wenn Sie Ihr JSON-Manifest nicht in den S3-Bucket hochgeladen haben, in dem Ihre ZIP-Dateien gespeichert sind, wählen Sie New Manifest (Neues Manifest) aus. Sie können das gesamte Manifest in dem JSON-Editor erstellen oder in ihn hineinkopieren. Weitere Informationen zum Erstellen des JSON-Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.
9. Wenn das Manifest fertiggestellt ist, wählen Sie Create package (Paket erstellen).
10. Warten Sie, bis Distributor die Erstellung des Pakets aus den ZIP-Dateien und dem Manifest abgeschlossen hat. Abhängig von der Anzahl und Größe der Pakete, die Sie hinzufügen, kann dies einige Minuten dauern. Distributor leitet Sie automatisch auf die Seite Package details (Paketdetails) für das neue Paket weiter, Sie können diese Seite aber jederzeit selbst öffnen, nachdem die Software hochgeladen ist. Auf der Seite Package details (Paketdetails) werden erst dann alle Informationen zu Ihrem Paket angezeigt, wenn Distributor den Paketerstellungsprozess abgeschlossen hat. Um den Upload- und Paketerstellungsprozess abzubrechen, wählen Sie Cancel (Abbrechen).

Hinzufügen eines Pakets (AWS CLI)

Sie können das verwenden AWS CLI , um ein Paket zu erstellen. Halten Sie die URL für den Bucket bereit, zu dem Sie Ihr Paket in Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket hochgeladen haben.

So fügen Sie ein Paket in Amazon S3 hinzu (AWS CLI)

1. Um das zum Erstellen eines Pakets AWS CLI zu verwenden, führen Sie den folgenden Befehl aus und ersetzen Sie dabei *`package-name`* durch den Namen Ihres Pakets und *`path-to-manifest-file`* durch den Dateipfad für Ihre JSON-Manifestdatei. DOC-EXAMPLE-BUCKET ist die URL des Amazon S3-Buckets, in dem das gesamte Paket gespeichert ist. Wenn Sie den Befehl create-document in Distributor ausführen, geben Sie den `Package`-Wert für `--document-type` an.

   Wenn Sie Ihre Manifestdatei nicht dem Amazon S3-Bucket hinzugefügt haben, ist der `--content`-Parameterwert der Dateipfad zur JSON-Manifestdatei.

   ```
   aws ssm create-document \
   ```

```
    --name "package-name" \
    --content file://path-to-manifest-file \
    --attachments Key="SourceUrl",Values="DOC-EXAMPLE-BUCKET" \
    --version-name version-value-from-manifest \
    --document-type Package
```

Im Folgenden wird ein Beispiel gezeigt.

```
aws ssm create-document \
    --name "ExamplePackage" \
    --content file://path-to-manifest-file \
    --attachments Key="SourceUrl",Values="https://s3.amazonaws.com/DOC-EXAMPLE-
BUCKET/ExamplePackage" \
    --version-name 1.0.1 \
    --document-type Package
```

2. Überprüfen Sie, ob Ihr Paket hinzugefügt wurde. Zeigen Sie hierzu das Paketmanifest an, indem Sie den folgenden Befehl ausführen, wobei Sie *package-name* durch den Namen Ihres Pakets ersetzen. Um eine spezifische Version des Dokuments (nicht identisch mit der Version eines Pakets) zu erhalten, können Sie den Parameter `--document-version` hinzufügen.

```
aws ssm get-document \
    --name "package-name"
```

Informationen zu anderen Optionen, die Sie mit dem Befehl create-document verwenden können, finden Sie unter create-document im Abschnitt AWS Systems Manager der AWS CLI -Command Reference. Informationen zu anderen Optionen, die Sie mit dem Befehl get-document verwenden können, finden Sie unter get-document.

Bearbeiten von Paketberechtigungen (Konsole)

Nachdem Sie ein Paket zu Distributor, eine Funktion von AWS Systems Manager, hinzugefügt haben, können Sie die Berechtigungen des Pakets in der Systems Manager-Konsole bearbeiten. Sie können den Berechtigungen eines Pakets weitere AWS-Konten hinzufügen. Pakete können nur für andere Konten in derselben AWS-Region freigegeben werden. Die Freigabe über Regionsgrenzen hinweg wird nicht unterstützt. Standardmäßig sind Pakete auf Private (Privat) festgelegt. Das bedeutet, dass nur Benutzer mit Zugriff auf das AWS-Konto des Paketerstellers Paketinformationen anzeigen und das Paket aktualisieren oder löschen können. Wenn Private (Privat)-Berechtigungen akzeptabel sind, können Sie dieses Verfahren überspringen.

> **Note**
>
> Sie können die Berechtigungen von Paketen aktualisieren, die mit 20 oder weniger Konten freigegeben werden.

So bearbeiten Sie Paketberechtigungen (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Seite Packages (Pakete) das Paket aus, für das Sie Berechtigungen bearbeiten möchten.
4. Wählen Sie auf der Registerkarte Package details (Paketdetails) die Option Edit permissions (Berechtigungen bearbeiten) aus, um Berechtigungen zu ändern.
5. Wählen Sie unter Edit permissions (Berechtigungen bearbeiten) die Option Shared with specific accounts (Für bestimmte Konten freigegeben) aus.
6. Fügen Sie in Shared with specific accounts (Für bestimmte Konten freigegeben) nacheinander AWS-Konto hinzu. Wenn Sie fertig sind, wählen Sie Speichern.

Bearbeiten von Paket-Tags (Konsole)

Nachdem Sie ein Paket zu Distributor, eine Funktion von AWS Systems Manager, hinzugefügt haben, können Sie die Tags des Pakets in der Systems Manager-Konsole bearbeiten. Diese Tags werden auf das Paket angewendet. Sie haben keine Verbindung zu Tags in dem verwalteten Knoten, auf dem Sie das Paket bereitstellen möchten. Tags unterscheiden nach Groß- und Kleinschreibung. Es handelt sich um Schlüssel-Wert-Paare, die Ihnen helfen können, Ihre Pakete nach für Ihre Organisation relevanten Kriterien zu gruppieren und zu filtern. Wenn Sie keine Tags hinzufügen möchten, können Sie Ihr Paket installieren oder eine neue Version hinzufügen.

So bearbeiten Sie Paket-Tags (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.

3. Wählen Sie auf der Seite Packages (Pakete) das Paket aus, für das Sie Tags bearbeiten möchten.
4. Wählen Sie auf der Registerkarte Package details (Paketdetails) in Tags (Tags) die Option Edit (Bearbeiten) aus.
5. Geben Sie unter Add tags (Tags hinzufügen) einen Schlüssel oder ein Schlüssel-Wert-Paar für das Tag ein. Klicken Sie anschließend auf Add (Hinzufügen). Wiederholen Sie dies, wenn Sie weitere Tags hinzufügen möchten. Um Tags zu löschen, wählen Sie unten im Fenster für das Tag X aus.
6. Wenn Sie Ihrem Paket keine weiteren Tags mehr hinzufügen möchten, wählen Sie Save (Speichern) aus.

Hinzufügen einer Paketversion zu Distributor

Um eine Paketversion hinzuzufügen, erstellen Sie ein Paket und verwenden Sie es dann, Distributor um eine Paketversion hinzuzufügen, indem Sie dem AWS Systems Manager (SSM-) Dokument, das bereits für ältere Versionen vorhanden ist, einen Eintrag hinzufügen. Distributorist eine Fähigkeit von. AWS Systems Manager Um Zeit zu sparen, aktualisieren Sie das Manifest für eine ältere Version des Pakets, ändern den Wert des Eintrags `version` im Manifest (z. B. von `Test_1.0` in `Test_2.0`) und speichern das Manifest als Manifest für die neue Version. Bei dem einfachen Add Version (Version hinzufügen)-Workflow in der Distributor-Konsole wird das Manifest automatisch aktualisiert.

Eine neue Paketversion kann:

- Mindestens eine der installierbaren Dateien ersetzen, die der aktuellen Version angefügt sind.
- Neue installierbare Dateien hinzufügen, um zusätzliche Plattformen zu unterstützen
- Dateien löschen, um die Unterstützung für bestimmte Plattformen zu beenden

Eine neuere Version kann denselben Amazon Simple Storage Service (Amazon S3)-Bucket verwenden, muss jedoch eine URL mit einem anderen Dateinamen am Ende besitzen. Sie können die Systems Manager-Konsole oder das AWS Command Line Interface (AWS CLI) verwenden, um die neue Version hinzuzufügen. Beim Hochladen einer installierbaren Datei mit demselben Namen wie eine vorhandene installierbare Datei in dem Amazon S3-Bucket wird die vorhandene Datei überschrieben. Es werden keine Dateien aus der älteren Version in die neue Version hineinkopiert. Sie müssen installierbare Dateien aus der älteren Version erneut hochladen, damit sie in die neue Version aufgenommen werden. Nachdem Distributor Ihre neue Paketversion erstellt hat, können Sie

den Amazon S3-Bucket löschen oder wiederverwenden, da Distributor Ihre Software im Rahmen der Versioning in einen internen Systems Manager-Bucket kopiert.

> **Note**
>
> Jedes Paket ist auf maximal 25 Versionen beschränkt. Sie können Versionen löschen, die nicht mehr benötigt werden.

Themen

- Hinzufügen einer Paketversion (Konsole)
- Hinzufügen einer Paketversion (AWS CLI)

Hinzufügen einer Paketversion (Konsole)

Führen Sie vor der Ausführung der folgenden Schritte die Anweisungen unter Erstellen eines Pakets aus, um ein neues Paket für die Version zu erstellen. Verwenden Sie anschließend die Systems Manager-Konsole, um Distributor eine neue Paketversion hinzuzufügen.

Hinzufügen einer Paketversion (einfach)

Um eine Paketversion mithilfe des einfachen Workflows hinzuzufügen, bereiten Sie aktualisierte installierbare Dateien vor oder fügen Sie installierbare Dateien hinzu, um weitere Plattformen und Architekturen zu unterstützen. Verwenden Sie anschließend Distributor zum Hochladen neuer und aktualisierter installierbarer Dateien und fügen Sie eine Paketversion hinzu. Die vereinfachte Add version (Version hinzufügen)-Workflow in der Distributor Konsole aktualisiert die Manifestdatei und das zugehörige SSM-Dokument automatisch.

So fügen Sie eine Paketversion hinzu (einfacher Workflow)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, dem Sie eine weitere Version hinzufügen möchten.
4. Wählen Sie auf der Seite Add version (Version hinzufügen) die Option Simple (Einfach).

5. Geben Sie unter Version name (Versionsname) einen Versionsnamen ein. Der Versionsname für die neue Version muss sich von der älteren Version unterscheiden. Versionsnamen dürfen maximal 512 Zeichen lang sein und dürfen keine Sonderzeichen enthalten.
6. Wählen Sie für S3 bucket name (S3-Bucketname), einen vorhandenen S3-Bucket aus der Liste aus. Dabei kann es sich um den Bucket handeln, den Sie zum Speichern installierbarer Dateien für ältere Versionen verwendet haben, aber die installierbaren Dateinamen müssen unterschiedlich sein, damit das Überschreiben vorhandener installierbaren Dateien in dem Bucket vermieden wird.
7. Geben Sie unter S3 key prefix (S3-Schlüsselpräfix) den Unterordner des Buckets ein, in dem Ihre installierbaren Komponenten gespeichert sind.
8. Navigieren Sie unter Upload Software (Software hochladen) zu den installierbaren Softwaredateien, die Sie für die neue Version anfügen möchten. Installierbare Versionen von vorhandenen Dateien werden nicht automatisch in eine neue Version herüberkopiert. Sie müssen alle installierbaren Dateien aus älteren Versionen des Pakets erneut hochladen, wenn Sie diese in die neue Version übernehmen möchten. Sie können mehrere Softwaredateien in einer einzigen Aktion hochladen.
9. Überprüfen Sie unter Für Target platform (Ziel-Plattform für jede Plattform, ob das Ziel-Betriebssystem für die installierbare Datei korrekt ist. Wenn das angezeigte Betriebssystem nicht korrekt ist, wählen Sie das richtige Betriebssystem aus der Dropdown-Liste aus.

   Bei dem Workflow Simple (Einfach) zur Versioning sind zusätzliche Schritte erforderlich, wenn nur eine Datei für mehrere Betriebssysteme als Ziel verwendet werden soll, da installierbare Dateien nur einmal hochgeladen werden. Wenn Sie zum Beispiel eine installierbare Softwaredatei mit dem Namen `Logtool_v1.1.1.rpm` hochladen, müssen Sie einige Standardeinstellungen für den Simple-Workflow ändern, um Distributor anzuweisen, für dieselbe Software sowohl das Amazon Linux- als auch das Ubuntu-Betriebssystem als Ziel festzulegen. Um dieses Problem zu beheben, können Sie eine der folgenden Aktionen ausführen.

   - Verwenden Sie stattdessen den Workflow Advanced (Erweitert) zur Versioning, zippen Sie jede installierbare Datei, bevor Sie beginnen, und richten Sie das Manifest manuell so ein, dass eine installierbare Datei für mehrere Betriebssystemplattformen oder -versionen verwendet werden kann. Weitere Informationen finden Sie unter Hinzufügen einer Paketversion (erweitert).
   - Bearbeiten Sie die Manifestdatei im Workflow Simple (Einfach) so, dass Ihre ZIP-Datei für mehrere Betriebssystemplattformen oder -versionen verwendet wird. Weitere Informationen

zu diesem Verfahren finden Sie am Ende von Schritt 4 in Schritt 2: Erstellen des JSON-Paketmanifests.

10. Stellen Sie unter Platform version (Plattformversion)sicher, dass als Betriebssystem-Plattformversion **_any**, eine Hauptversionsnummer, gefolgt von einem Platzhalter (7.*), angezeigt wird, oder genau die spezifische Betriebssystemversion, die Sie als Plattformversion für Ihre Softwareinstallation verwenden möchten. Weitere Informationen zum Festlegen einer Plattformversion finden Sie unter Schritt 4 in Schritt 2: Erstellen des JSON-Paketmanifests.
11. Wählen Sie unter Architecture (Architektur) für jeden installierbare Datei die richtige Prozessorarchitektur aus der Dropdown-Liste aus. Weitere Informationen zu unterstützten Architekturen finden Sie unter Unterstützte Paketplattformen und -architekturen.
12. (Optional) Erweitern Sie Scripts (Skripts) und überprüfen Sie die Installations- und Deinstallationsskripts, die Distributor für Ihre installierbare Software generiert hat.
13. Zum Hinzufügen weiterer installierbarer Softwaredateien zu der neuen Version wählen Sie Add Software (Software hinzufügen). Andernfalls fahren Sie mit dem nächsten Schritt fort.
14. (Optional) Erweitern Sie Manifest und überprüfen Sie das JSON-Manifest des Pakets, das Distributor für Ihre installierbare Software generiert hat. Wenn Sie Informationen über Ihre installierbare Software geändert haben, nachdem Sie mit dieser Prozedur begonnen haben, beispielsweise die Plattformversion oder die Zielplattform, wählen Sie Generate Manifest (Manifest erzeugen), um das Paketmanifest zu aktualisieren.

    Sie können das Manifest manuell bearbeiten, wenn Sie möchten, dass für eine installierbare Software mehr als ein Betriebssystem als Ziel festgelegt wird, wie in Schritt 9 beschrieben. Weitere Informationen zum Bearbeiten des Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.
15. Wählen Sie nach dem Hinzufügen der Software und der Überprüfung der Daten zur Zielplattform, zur Version und zur Architektur Sie Add version (Version hinzufügen).
16. Warten Sie, bis Distributor das Hochladen Ihrer Software und das Erstellen Ihres neuen Pakets abgeschlossen Hat. Distributor zeigt den Uploadstatus für jede installierbare Datei einzeln an. Abhängig von der Anzahl und Größe der Pakete, die Sie hinzufügen, kann dies einige Minuten dauern. Distributor leitet Sie automatisch auf die Seite Package details (Paketdetails) für das neue Paket weiter, Sie können diese Seite aber jederzeit selbst öffnen, nachdem die Software hochgeladen ist. Auf der Seite Package details (Paketdetails) werden erst dann alle Informationen zu Ihrem Paket angezeigt, wenn Distributor den Erstellungsprozess für die neue Paketversion abgeschlossen hat. Um den Uploadvorgang bzw. den prozess zur Erstellung der Paketversion anzuhalten, wählen Sie Stop upload (Upload anhalten).

17. Wenn Distributor keine installierbaren Softwaredateien hochladen kann, wird eine Fehlermeldung Upload failed (Upload fehlgeschlagen) angezeigt. Um den Uploadversuch zu wiederholen, wählen Sie Retry Upload (Uploadversuch wiederholen). Weitere Informationen zur Fehlerbehebung bei der Paketversionserstellung finden Sie unter Fehlerbehebung für AWS Systems ManagerDistributor.
18. Wenn Distributor die neue Paketversion erstellt hat. zeigen Sie auf der Seite Details (Details) auf der Registerkarte Versions (Versionen) die neue Version in der Liste der verfügbaren Paketversionen an. Legen Sie die Standardversion des Pakets fest, indem Sie eine Version auswählen. Wählen Sie anschließend Set default version (Als Standardversion festlegen) aus.

    Wenn Sie keine Standardversion festlegen, ist die neueste Paketversion die Standardversion.

Hinzufügen einer Paketversion (erweitert)

Um eine Paketversion hinzuzufügen, erstellen Sie ein Paket und verwenden Sie anschließend Distributor, um eine Paketversion hochzuladen, indem Sie dem für ältere Versionen vorhandenen SSM-Dokument einen Eintrag hinzufügen. Um Zeit zu sparen, aktualisieren Sie das Manifest für eine ältere Version des Pakets, ändern den Wert des Eintrags `version` im Manifest (z. B. von `Test_1.0` in `Test_2.0`) und speichern das Manifest als Manifest für die neue Version. Sie müssen das Manifest so ändern, dass eine neue Version des Pakets hinzugefügt wird. Dies führen Sie mit dem Advanced-Workflow durch.

So fügen Sie eine Paketversion hinzu (erweitert)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, dem Sie eine weitere Version hinzufügen möchten, an wählen Sie dann Add version (Version hinzufügen).
4. Geben Sie unter Version name (Versionsname) den exakten Wert des Eintrags `version` Ihrer Manifestdatei ein.
5. Wählen Sie für S3 bucket name (S3-Bucketname), einen vorhandenen S3-Bucket aus der Liste aus. Dabei kann es sich um den Bucket handeln, den Sie zum Speichern installierbarer Dateien für ältere Versionen verwendet haben, aber die installierbaren Dateinamen müssen unterschiedlich sein, damit das Überschreiben vorhandener installierbaren Dateien in dem Bucket vermieden wird.

6. Geben Sie unter S3 key prefix (S3-Schlüsselpräfix) den Unterordner des Buckets ein, in dem Ihre installierbaren Komponenten gespeichert sind.
7. Wählen Sie unter Manifest die Option Extract from package (Aus Paket extrahieren) aus, um ein Manifest zu verwenden, das Sie mit Ihren ZIP-Dateien in den S3-Bucket hochgeladen haben.

   (Optional) Wenn Sie kein aktualisiertes JSON-Manifest in den Amazon S3-Bucket mit Ihren ZIP-Dateien hochgeladen haben, wählen Sie New manifest (Neues Manifest) aus. Sie können das gesamte Manifest in dem JSON-Editor erstellen oder in ihn hineinkopieren. Weitere Informationen zum Erstellen des JSON-Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.
8. Wenn das Manifest fertiggestellt ist, wählen Sie Add package version (Paketversion hinzufügen).
9. Zeigen Sie auf der Seite Details (Details) auf der Registerkarte Versions (Versionen) die neue Version in der Liste der verfügbaren Paketversionen an. Legen Sie die Standardversion des Pakets fest, indem Sie eine Version auswählen. Wählen Sie anschließend Set default version (Als Standardversion festlegen) aus.

   Wenn Sie keine Standardversion festlegen, ist die neueste Paketversion die Standardversion.

Hinzufügen einer Paketversion (AWS CLI)

Sie können das verwenden AWS CLI , um eine neue Paketversion hinzuzufügenDistributor. Vor der Ausführung der folgenden Befehle müssen Sie eine neue Paketversion erstellen und diese zu S3 hochladen wie am Anfang dieses Themas beschrieben.

So fügen Sie eine Paketversion hinzu (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um das AWS Systems Manager Dokument mit einem Eintrag für eine neue Paketversion zu bearbeiten. Ersetzen Sie *`document-name`* durch den Namen Ihres Dokuments. Ersetzen Sie *`DOC-EXAMPLE-BUCKET`* mit der URL des JSON-Manifests, das Sie in Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket kopiert haben. *`S3-bucket-URL-of-package`* ist die URL des Amazon S3-Buckets, in dem das gesamte Paket gespeichert ist. Ersetzen Sie *`version-name-from-updated-manifest`* durch den Wert für `version` im Manifest. Legen Sie den Parameter `--document-version` auf `$LATEST` fest, um das Dokument für diese Paketversion als aktuelle Version des Dokuments festzulegen.

   ```
   aws ssm update-document \
       --name "document-name" \
   ```

```
    --content "S3-bucket-URL-to-manifest-file" \
    --attachments Key="SourceUrl",Values="DOC-EXAMPLE-BUCKET" \
    --version-name version-name-from-updated-manifest \
    --document-version $LATEST
```

Im Folgenden wird ein Beispiel gezeigt.

```
aws ssm update-document \
    --name ExamplePackage \
    --content "https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/ExamplePackage/
manifest.json" \
    --attachments Key="SourceUrl",Values="https://s3.amazonaws.com/DOC-EXAMPLE-
BUCKET/ExamplePackage" \
    --version-name 1.1.1 \
    --document-version $LATEST
```

2. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Ihr Paket aktualisiert wurde, und das Paketmanifest anzuzeigen. Ersetzen Sie *package-name* durch den Namen Ihres Pakets. Optional können Sie *document-version* durch die Versionsnummer des von Ihnen aktualisierten Dokuments (nicht identisch mit der Paketversion) ersetzen. Wenn diese Paketversion der aktuellen Version des Dokuments zugeordnet ist, können Sie `$LATEST` als Wert des optionalen Parameters `--document-version` angeben.

```
aws ssm get-document \
    --name "package-name" \
    --document-version "document-version"
```

Informationen zu anderen Optionen, die Sie mit dem update-document Befehl verwenden können, finden Sie update-documentim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Installieren oder Aktualisieren von Paketen

Sie können Pakete auf Ihren AWS Systems Manager verwalteten Knoten bereitstellenDistributor, indem Sie eine Funktion von verwenden AWS Systems Manager. Um die Pakete bereitzustellen, verwenden Sie entweder AWS Management Console oder AWS Command Line Interface (AWS CLI). Sie können pro Befehl eine Version eines Pakets bereitstellen. Sie können neue Pakete installieren oder vorhandene Installationen direkt aktualisieren. Sie können wählen, ob Sie eine bestimmte Version oder stets die aktuelle Version eines Pakets bereitstellen möchten. Wir empfehlenState Manager, für die Installation von Paketen eine Funktion von AWS Systems Manager zu verwenden.

Durch die Verwendung wird State Manager sichergestellt, dass auf Ihren verwalteten Knoten immer die neueste up-to-date Version Ihres Pakets ausgeführt wird.

Präferenz	AWS Systems Manager Aktion	Weitere Informationen
Installieren oder aktualisieren Sie ein Paket sofort.	Run Command	• Einmaliges Installieren oder Aktualisieren eines Pakets (Konsole) • Einmaliges Installieren eines Pakets (AWS CLI) • Einmaliges Aktualisieren eines Pakets (AWS CLI)
Installieren Sie ein Paket nach einem Zeitplan, sodass die Installation immer die Standardversion enthält.	State Manager	• Planen einer Paketinstallation oder -aktualisierung (Konsole) • Planen einer Paketinstallation (AWS CLI) • Planen einer Paketaktualisierung (AWS CLI)
Installieren Sie ein Paket automatisch auf neuen verwalteten Knoten, die ein bestimmtes Tag oder einen bestimmten Satz von Tags besitzen. Zum Beispiel die Installation des CloudWatch Amazon-Agenten auf neuen Instances.	State Manager	Eine Möglichkeit besteht in der Anwendung von Tags auf neue verwaltete Knoten und die anschließende Auflistung der Tags als Ziele in der State Manager-Zuordnung. State Manager installiert automatisch das Paket in einer Zuordnung auf verwalteten Knoten, deren Tags übereinstimmen. Siehe Informationen zu Zielen und Ratensteuerungen in State Manager Zuordnungen.

Themen

- Einmaliges Installieren oder Aktualisieren eines Pakets (Konsole)
- Planen einer Paketinstallation oder -aktualisierung (Konsole)
- Einmaliges Installieren eines Pakets (AWS CLI)
- Einmaliges Aktualisieren eines Pakets (AWS CLI)
- Planen einer Paketinstallation (AWS CLI)
- Planen einer Paketaktualisierung (AWS CLI)

Einmaliges Installieren oder Aktualisieren eines Pakets (Konsole)

Sie können die AWS Systems Manager Konsole verwenden, um ein Paket einmal zu installieren oder zu aktualisieren. Wenn Sie eine einmalige Installation konfigurieren, verwendet Distributor AWS Systems Manager Run Command, eine Funktion von AWS Systems Manager, zum Ausführen der Installation.

So installieren oder aktualisieren Sie ein Paket einmal (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, das Sie installieren möchten.
4. Wählen Sie Install one time (Einmal installieren) aus.

   Dieser Befehl öffnet Run Command mit dem Befehlsdokument `AWS-ConfigureAWSPackage`. Ihr Distributor-Paket ist vorausgewählt.
5. Wählen Sie unter Document version (Dokumentversion) die Version des `AWS-ConfigureAWSPackage`-Dokuments aus, das Sie ausführen möchten.
6. Wählen Sie für Action (Aktion) die Option Install (Installieren).
7. Wählen Sie unter Installation type (Installationstyp) eine der folgenden Optionen aus:

   - Uninstall and reinstall (Deinstallieren und neu installieren): Das Paket wird vollständig deinstalliert und dann neu installiert. Die Anwendung ist bis zum Abschluss der Neuinstallation nicht verfügbar.
   - In-place update (Direkte Aktualisierung): Der vorhandenen Installation werden entsprechend den Anweisungen, die Sie in einem `update`-Skript angeben, nur neue oder geänderte Dateien

hinzugefügt. Die Anwendung ist während des Aktualisierungsprozesses weiterhin verfügbar. Diese Option wird für AWS veröffentlichte Pakete außer dem `AWSEC2Launch-Agent` Paket nicht unterstützt.

8. Überprüfen Sie, ob unter Name der Name des ausgewählten Pakets angegeben ist.
9. (Optional) Geben Sie unter Version den Versionsnamen des Pakets ein. Wenn Sie dieses Feld leer lassen, installiert Run Command die von Ihnen in Distributor ausgewählte Standardversion.
10. Wählen Sie im Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Geräte manuell auswählen oder eine Ressourcengruppe angeben.

    > **Note**
    >
    > Wenn kein verwalteter Knoten in der Liste angezeigt wird, lesen Sie Problembehandlung bei der Verfügbarkeit verwalteter Knoten.

11. Für Other parameters (Weitere Parameter):

    - Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
    - Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

12. Für Rate control (Temposteuerung):

    - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz von Zielen an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

      > **Note**
      >
      > Wenn Sie Ziele ausgewählt haben, indem Sie Tags oder Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen können, beschränken, indem Sie einen Prozentsatz angeben.

    - Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen Zielen beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von verwalteten Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der

vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

13. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

14. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

15. Wenn Sie bereit sind, das Paket zu installieren, klicken Sie auf Run (Ausführen).
16. Im Bereich Command status (Befehlsstatus) wird der Fortschritt der Installation angezeigt. Wenn der Befehl noch ausgeführt wird, klicken Sie oben links in der Konsole auf das Aktualisierungssymbol, bis in der Spalte Overall status (Gesamtstatus) oder Detailed status (Detailstatus) der Status Success (Erfolgreich) oder Failed (Fehlgeschlagen) angezeigt wird.
17. Klicken Sie im Bereich Targets and outputs (Ziele und Ausgaben) auf die Schaltfläche neben dem Namen eines verwalteten Knotens und wählen Sie dann View output (Ausgabe anzeigen).

    Der Befehlsausgabeseite zeigt die Ergebnisse der Befehlsausführung an.

18. (Optional) Wenn Sie die Befehlsausgabe in einen Amazon S3-Bucket schreiben möchten, wählen Sie Amazon S3, um die Ausgabeprotokolldaten anzuzeigen.

Planen einer Paketinstallation oder -aktualisierung (Konsole)

Sie können die AWS Systems Manager Konsole verwenden, um die Installation oder Aktualisierung eines Pakets zu planen. Wenn Sie die Paketinstallation oder -aktualisierung planen, verwendet Distributor AWS Systems Manager State Manager zum Installieren oder Aktualisieren.

So planen Sie eine Paketinstallation (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, das Sie installieren oder aktualisieren möchten.
4. Wählen Sie unter Package (Paket) die Option Install on a schedule (Nach Plan installieren) aus.

   Dieser Befehl öffnet State Manager zu einer neuen Zuordnung, die für Sie erstellt wurde.
5. Geben Sie unter Name einen Namen ein (z. B. **`Deploy-test-agent-package`**). Dies ist zwar optional, wird aber empfohlen. Der Name darf keine Leerzeichen enthalten.
6. In der Liste Document (Dokument) ist der Dokumentname `AWS-ConfigureAWSPackage` bereits ausgewählt.
7. Überprüfen Sie unter Action (Aktion), ob Install (Installieren) ausgewählt ist.
8. Wählen Sie unter Installation type (Installationstyp) eine der folgenden Optionen aus:

   - Uninstall and reinstall (Deinstallieren und neu installieren): Das Paket wird vollständig deinstalliert und dann neu installiert. Die Anwendung ist bis zum Abschluss der Neuinstallation nicht verfügbar.
   - In-place update (Direkte Aktualisierung): Der vorhandenen Installation werden entsprechend den Anweisungen, die Sie in einem `update`-Skript angeben, nur neue oder geänderte Dateien hinzugefügt. Die Anwendung ist während des Aktualisierungsprozesses weiterhin verfügbar.
9. Überprüfen Sie unter Name, ob der Name Ihres Pakets angegeben ist.
10. Geben Sie unter Version die Versionskennung ein, wenn Sie eine andere Paketversion als die zuletzt veröffentlichte Version installieren möchten.
11. Wählen Sie unter Targets (Ziele) die Optionen Selecting all managed instances in this account (Alle verwalteten Instances in diesem Konto auswählen), Specifying tags (Tags angeben) oder Manually Selecting Instance (Instance manuell auswählen) aus. Wenn Sie die Zielressourcen

mithilfe von Tags ausgewählt haben, geben Sie einen Tag-Schlüssel und einen Tag-Wert in die entsprechenden Felder ein.

> **Note**
>
> Sie können verwaltete AWS IoT Greengrass Kerngeräte auswählen, indem Sie entweder Alle verwalteten Instanzen in diesem Konto auswählen oder Instanz manuell auswählen wählen.

12. Wählen Sie unter Specify schedule (Plan angeben) die Option On Schedule (Nach Plan) aus, um die Zuordnung nach einem regelmäßigen Zeitplan auszuführen, oder No Schedule (Kein Plan), um die Zuordnung einmalig auszuführen. Weitere Informationen zu diesen Optionen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager. Verwenden Sie die Steuerelemente, um einen `cron`- oder Rate-Zeitplan für die Zuordnung zu erstellen.
13. Wählen Sie Create Association.
14. Klicken Sie auf der Seite Association (Zuordnung) auf die Schaltfläche neben der von Ihnen erstellten Zuordnung und wählen Sie dann Apply association now (Zuordnung jetzt anwenden) aus.

    State Manager erstellt die Zuordnung und führt sie sofort auf den angegebenen Zielen aus. Weitere Informationen zu den Ergebnissen der Ausführung von Zuordnungen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager in diesem Handbuch.

Weitere Informationen zur Verwendung der Optionen unter Advanced Options (Erweiterte Optionen), Rate control (Ratensteuerung) und Output options (Ausgabeoptionen) finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

Einmaliges Installieren eines Pakets (AWS CLI)

Sie können send-command den ausführen AWS CLI , um ein Distributor Paket einmal zu installieren. Wenn das Paket bereits installiert ist, wird die Anwendung offline geschaltet, während das Paket deinstalliert und stattdessen die neue Version installiert wird.

So installieren Sie ein Paket einmalig (AWS CLI)

- Führen Sie in der AWS CLI den folgenden aus.

```
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**
>
> Das Standardverhalten für `installationType` ist `Uninstall and reinstall`. Sie können `"installationType":["Uninstall and reinstall"]` im Befehl weglassen, wenn Sie ein komplettes Paket installieren.

Im Folgenden wird ein Beispiel gezeigt.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-00000000000000" \
    --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["ExamplePackage"]}'
```

Informationen zu anderen Optionen, die Sie mit dem send-command Befehl verwenden können, finden Sie send-commandim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Einmaliges Aktualisieren eines Pakets (AWS CLI)

Sie können das Programm ausführen send-command AWS CLI , um ein Distributor Paket zu aktualisieren, ohne die zugehörige Anwendung offline zu schalten. Nur neue oder aktualisierte Dateien im Paket werden ersetzt.

So aktualisieren Sie ein Paket einmal (AWS CLI)

- Führen Sie in der AWS CLI den folgenden aus.

```
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**
>
> Wenn Sie neue oder geänderte Dateien hinzufügen, müssen Sie `"installationType":["In-place update"]` in den Befehl einschließen.

Im Folgenden wird ein Beispiel gezeigt.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-02573cafcfEXAMPLE" \
    --parameters '{"action":["Install"],"installationType":["In-place
 update"],"name":["ExamplePackage"]}'
```

Informationen zu anderen Optionen, die Sie mit dem send-command Befehl verwenden können, finden Sie send-commandim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Planen einer Paketinstallation (AWS CLI)

Sie können create-association den ausführen AWS CLI , um ein Distributor Paket nach einem Zeitplan zu installieren. Der Wert für `--name`, d. h. der Name des Dokuments, ist stets `AWS-ConfigureAWSPackage`. Der folgende Befehl verwendet den Schlüssel `InstanceIds` zur Angabe von verwalteten Knoten als Ziel. Wenn das Paket bereits installiert ist, wird die Anwendung offline geschaltet, während das Paket deinstalliert und stattdessen die neue Version installiert wird.

```
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**
>
> Das Standardverhalten für `installationType` ist `Uninstall and reinstall`. Sie können `"installationType":["Uninstall and reinstall"]` im Befehl weglassen, wenn Sie ein komplettes Paket installieren.

Im Folgenden wird ein Beispiel gezeigt.

```
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\"]}]
```

Informationen zu anderen Optionen, die Sie mit dem create-association Befehl verwenden können, finden Sie create-associationim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Planen einer Paketaktualisierung (AWS CLI)

Sie können den ausführen create-association AWS CLI , um ein Distributor Paket nach einem Zeitplan zu aktualisieren, ohne die zugehörige Anwendung offline zu nehmen. Nur neue oder aktualisierte Dateien im Paket werden ersetzt. Der Wert für `--name`, d. h. der Name des Dokuments, ist stets `AWS-ConfigureAWSPackage`. Der folgende Befehl verwendet den Schlüssel `InstanceIds` zur Angabe von Ziel-Instances.

```
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**
>
> Wenn Sie neue oder geänderte Dateien hinzufügen, müssen Sie `"installationType": ["In-place update"]` in den Befehl einschließen.

Im Folgenden wird ein Beispiel gezeigt.

```
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\"]}]
```

Informationen zu anderen Optionen, die Sie mit dem create-association Befehl verwenden können, finden Sie create-associationim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Deinstallieren eines Pakets

Sie können die AWS Management Console oder die AWS Command Line Interface (AWS CLI) verwenden, um Distributor-Pakete aus Ihren von AWS Systems Manager verwalteten Knoten mithilfe von Run Command zu entfernen. Distributor und Run Command sind Funktionen von AWS Systems Manager. In dieser Version können Sie pro Befehl eine Version eines Pakets deinstallieren. Sie können eine bestimmte Version oder die Standardversion deinstallieren.

Themen

- Deinstallieren eines Pakets (Konsole)
- Deinstallieren eines Pakets (AWS CLI)

Deinstallieren eines Pakets (Konsole)

Sie können Run Command in der Systems Manager-Konsole verwenden, um ein Paket einmalig zu deinstallieren. Distributor verwendet AWS Systems Manager Run Command zur Deinstallation von Paketen.

So deinstallieren Sie ein Paket (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie auf der Run Command-Startseite Run command (Befehl ausführen) aus.
4. Wählen Sie das Befehlsdokument `AWS-ConfigureAWSPackage` aus.
5. Wählen Sie in Action (Aktion) die Option Uninstall (Deinstallieren) aus.

6. Geben Sie in Name (Name) den Namen des Pakets ein, das Sie deinstallieren möchten.
7. Für Targets (Ziele) wählen Sie aus, wie Sie Ihre verwaltete Knoten anvisieren möchten. Sie können einen Tag-Schlüssel und Werte angeben, die von den Zielen geteilt werden. Sie können Ziele auch angeben, indem Sie Attribute wie ID, Plattform und SSM Agent-Version auswählen.
8. Sie können in den erweiterten Optionen Kommentare zur Operation hinzufügen, die Werte für Concurrency (Gleichzeitigkeit) und Error threshold (Fehlerschwellenwert) in Rate control (Ratenkontrolle) ändern, Ausgabeoptionen angeben oder Amazon Simple Notification Service (Amazon SNS)-Benachrichtigungen konfigurieren. Weitere Informationen finden Sie unter Ausführen von Befehlen über die Konsole in diesem Handbuch.
9. Wenn Sie zur Deinstallation des Pakets bereit sind, wählen Sie Run (Ausführen) und dann View results (Ergebnisse anzeigen) aus.
10. Wählen Sie in der Befehlsliste den `AWS-ConfigureAWSPackage` aus, den Sie ausgeführt haben. Wenn der Befehl noch in Bearbeitung ist, wählen Sie das Aktualisierungssymbol oben rechts in der Konsole aus.
11. Wenn die Spalte Status Success (Erfolg) oder Failed (Fehlgeschlagen) anzeigt, wählen Sie die Registerkarte Output (Ausgabe).
12. Wählen Sie View output (Ausgabe anzeigen) aus. Der Befehlsausgabeseite zeigt die Ergebnisse der Befehlsausführung an.

Deinstallieren eines Pakets (AWS CLI)

Sie können die AWS CLI verwenden, um ein Distributor-Paket mittels Run Command von Ihren verwalteten Knoten zu deinstallieren.

So deinstallieren Sie ein Paket (AWS CLI)

- Führen Sie in der AWS CLI den folgenden aus.

```
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)"]}'
```

  Im Folgenden wird ein Beispiel gezeigt.

```
aws ssm send-command \
```

```
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-02573cafcfEXAMPLE" \
    --parameters '{"action":["Uninstall"],"name":["Test-ConfigureAWSPackage"]}'
```

Informationen zu anderen Optionen, die Sie mit dem Befehl send-command verwenden können, finden Sie unter send-command im Abschnitt AWS Systems Manager der AWS CLI-Command Reference.

Löschen eines Pakets

In diesem Abschnitt wird beschrieben, wie Sie ein Paket löschen. Sie können eine Version eines Pakets nicht löschen, sondern nur das gesamte Paket.

Löschen eines Pakets (Konsole)

Sie können die AWS Systems Manager-Konsole verwenden, um ein Paket oder eine Paketversion vonDistributor, eine Funktion von AWS Systems Manager, löschen. Durch das Löschen eines Pakets werden alle Versionen des Pakets aus Distributor gelöscht.

So löschen Sie ein Paket (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, das Sie löschen möchten.
4. Wählen Sie auf der Detailseite des Pakets Delete package (Paket löschen) aus.
5. Wenn Sie zum Bestätigen des Löschvorgangs aufgefordert werden, wählen Sie Delete package (Paket löschen) aus.

Löschen einer Paketversion (Konsole)

Sie können die Systems Manager-Konsole verwenden, um eine Paketversion aus Distributor zu löschen.

So löschen Sie eine Paketversion (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, von dem Sie eine Version löschen möchten.
4. Wählen Sie auf der Versionsseite für das Paket die zu löschende Version und anschließend die Option Delete version (Version löschen) aus.
5. Wenn Sie zum Bestätigen des Löschvorgangs aufgefordert werden, wählen Sie Delete package version (Paketversion löschen) aus.

Löschen eines Pakets (Befehlszeile)

Sie können Ihr bevorzugtes Befehlszeilen-Tool verwenden, um ein Paket aus Distributor zu löschen.

Linux & macOS

So löschen Sie ein Paket (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um Dokumente für spezifische Pakete aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls das Paket, das Sie löschen möchten.

```
aws ssm list-documents \
    --filters Key=Name,Values=package-name
```

2. Führen Sie den folgenden Befehl aus, um ein Paket zu löschen. Ersetzen Sie *package-name* durch den Namen des Pakets.

```
aws ssm delete-document \
    --name "package-name"
```

3. Führen Sie den Befehl list-documents erneut aus, um zu überprüfen, ob das Paket gelöscht wurde. Das Paket, das Sie gelöscht haben, sollte nicht in die Liste aufgenommen werden.

```
aws ssm list-documents \
    --filters Key=Name,Values=package-name
```

Windows

So löschen Sie ein Paket (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um Dokumente für spezifische Pakete aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls das Paket, das Sie löschen möchten.

```
aws ssm list-documents ^
    --filters Key=Name,Values=package-name
```

2. Führen Sie den folgenden Befehl aus, um ein Paket zu löschen. Ersetzen Sie *package-name* durch den Namen des Pakets.

```
aws ssm delete-document ^
    --name "package-name"
```

3. Führen Sie den Befehl list-documents erneut aus, um zu überprüfen, ob das Paket gelöscht wurde. Das Paket, das Sie gelöscht haben, sollte nicht in die Liste aufgenommen werden.

```
aws ssm list-documents ^
    --filters Key=Name,Values=package-name
```

PowerShell

Löschen eines Pakets (Tools for PowerShell)

1. Führen Sie den folgenden Befehl aus, um Dokumente für spezifische Pakete aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls das Paket, das Sie löschen möchten.

```
$filter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Name"
$filter.Values = "package-name"

Get-SSMDocumentList `
    -Filters @($filter)
```

2. Führen Sie den folgenden Befehl aus, um ein Paket zu löschen. Ersetzen Sie *package-name* durch den Namen des Pakets.

```
Remove-SSMDocument `
    -Name "package-name"
```

3. Führen Sie den Befehl Get-SSMDocumentList erneut aus, um zu überprüfen, ob das Paket gelöscht wurde. Das Paket, das Sie gelöscht haben, sollte nicht in die Liste aufgenommen werden.

```
$filter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Name"
$filter.Values = "package-name"

Get-SSMDocumentList `
    -Filters @($filter)
```

Löschen einer Paketversion (Befehlszeile)

Sie können Ihr bevorzugtes Befehlszeilen-Tool verwenden, um eine Paketversion aus Distributor zu löschen.

Linux & macOS

So löschen Sie eine Paketversion (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um die Versionen Ihres Pakets aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls die Paketversion, die Sie löschen möchten.

```
aws ssm list-document-versions \
    --name "package-name"
```

2. Führen Sie den folgenden Befehl aus, um eine Paketversion zu löschen. Ersetzen Sie *package-name* durch den Paketnamen und *version* durch die Versionsnummer.

```
aws ssm delete-document \
    --name "package-name" \
    --document-version version
```

3. Führen Sie den Befehl list-document-versions aus, um zu überprüfen, ob die Version des Pakets gelöscht wurde. Die von Ihnen gelöschte Paketversion sollte nicht gefunden werden.

```
aws ssm list-document-versions \
    --name "package-name"
```

Windows

So löschen Sie eine Paketversion (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um die Versionen Ihres Pakets aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls die Paketversion, die Sie löschen möchten.

```
aws ssm list-document-versions ^
    --name "package-name"
```

2. Führen Sie den folgenden Befehl aus, um eine Paketversion zu löschen. Ersetzen Sie *package-name* durch den Paketnamen und *version* durch die Versionsnummer.

```
aws ssm delete-document ^
    --name "package-name" ^
    --document-version version
```

3. Führen Sie den Befehl list-document-versions aus, um zu überprüfen, ob die Version des Pakets gelöscht wurde. Die von Ihnen gelöschte Paketversion sollte nicht gefunden werden.

```
aws ssm list-document-versions ^
    --name "package-name"
```

PowerShell

Löschen einer Paketversion (Tools for PowerShell)

1. Führen Sie den folgenden Befehl aus, um die Versionen Ihres Pakets aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls die Paketversion, die Sie löschen möchten.

```
Get-SSMDocumentVersionList `
    -Name "package-name"
```

2. Führen Sie den folgenden Befehl aus, um eine Paketversion zu löschen. Ersetzen Sie *package-name* durch den Paketnamen und *version* durch die Versionsnummer.

```
Remove-SSMDocument `
    -Name "package-name" `
    -DocumentVersion version
```

3. Führen Sie den Befehl Get-SSMDocumentVersionList aus, um zu überprüfen, ob die Version des Pakets gelöscht wurde. Die von Ihnen gelöschte Paketversion sollte nicht gefunden werden.

```
Get-SSMDocumentVersionList `
    -Name "package-name"
```

Informationen zu anderen Optionen, die Sie mit dem Befehl list-documents verwenden können, finden Sie unter list-documents im Abschnitt AWS Systems Manager der AWS CLI-Command Reference. Informationen zu anderen Optionen, die Sie mit dem Befehl delete-document verwenden können, finden Sie unter delete-document.

Prüfen und Protokollieren von Distributor-Aktivitäten

Sie können AWS CloudTrail zur Prüfung der Aktivität im Zusammenhang mit Distributor, eine Funktion von AWS Systems Manager, verwenden. Weitere Informationen zu den Prüfungs- und Protokollierungsoptionen für Systems Manager finden Sie unter Überwachung AWS Systems Manager.

Distributor-Aktivität mithilfe von CloudTrail prüfen

CloudTrail erfasst über die AWS Systems Manager-Konsole, die AWS Command Line Interface (AWS CLI) und das Systems Manager SDK ausgeführte API-Aufrufe. Sie können die Informationen in der CloudTrail-Konsole oder in einem Amazon Simple Storage Service (Amazon S3)-Bucket anzeigen. Für alle CloudTrail-Protokolle in Ihrem Konto wird nur ein Bucket benötigt.

Protokolle von Run Command- und State Manager-Aktionen zeigen Aktivitäten im Zusammenhang mit Dokumenterstellung, Paketinstallation und Paketdeinstallation an. Run Command und State Manager sind Funktionen von AWS Systems Manager. Weitere Informationen zum Anzeigen und Verwenden von CloudTrail-Protokollen von Systems Manager-Aktivitäten finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.

Fehlerbehebung für AWS Systems ManagerDistributor

Die folgenden Informationen können Ihnen bei der Behebung von Problemen helfen, die auftreten könnenDistributor, wenn Sie eine Funktion von verwenden AWS Systems Manager.

Themen

- Falsches Paket mit identischem Namen installiert
- Fehler: Abruf des Manifests fehlgeschlagen: Aktuelle Version des Pakets wurde nicht gefunden
- Fehler: Fehler beim Abrufen des Manifests: Validierungsausnahme
- Paket wird nicht unterstützt (dem Paket fehlt Installationsaktion)
- Fehler: Manifest konnte nicht heruntergeladen werden: Dokument mit dem Namen ist nicht vorhanden
- Hochladen fehlgeschlagen.

Falsches Paket mit identischem Namen installiert

Problem: Sie haben ein Paket installiert, Distributor hat stattdessen ein anderes Paket installiert.

Ursache: Während der Installation findet Systems Manager von AWS veröffentlichte Pakete als Ergebnisse, bevor benutzerdefinierte externe Pakete gefunden werden. Wenn Ihr benutzerdefinierter Paketname mit dem Namen eines AWS veröffentlichten Pakets identisch ist, wird das AWS Paket anstelle Ihres Pakets installiert.

Lösung: Um dieses Problem zu vermeiden, geben Sie Ihrem Paket einen anderen Namen als den Namen eines AWS veröffentlichten Pakets.

Fehler: Abruf des Manifests fehlgeschlagen: Aktuelle Version des Pakets wurde nicht gefunden

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
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
```

Ursache: Sie verwenden eine SSM Agent-Version mit einer früheren Distributor-Version als Version 2.3.274.0.

Lösung: Aktualisieren Sie die SSM Agent-Version auf Version 2.3.274.0 oder höher. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command oder Anleitung: Automatische Aktualisierung von SSM Agent (CLI).

Fehler: Fehler beim Abrufen des Manifests: Validierungsausnahme

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
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
```

Ursache: Sie verwenden eine SSM Agent-Version mit einer früheren Distributor-Version als Version 2.3.274.0.

Lösung: Aktualisieren Sie die SSM Agent-Version auf Version 2.3.274.0 oder höher. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command oder Anleitung: Automatische Aktualisierung von SSM Agent (CLI).

Paket wird nicht unterstützt (dem Paket fehlt Installationsaktion)

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Package is not supported (package is missing install action)
```

Ursache: Die Paketverzeichnisstruktur ist falsch.

Lösung: Zipen Sie kein übergeordnetes Verzeichnis, das die Software und die erforderlichen Skripte enthält. Erstellen Sie stattdessen eine `.zip`-Datei aller erforderlichen Inhalte direkt im absoluten Pfad. Um zu überprüfen, ob die `.zip`-Datei korrekt erstellt wurde, entpacken Sie das Zielplattformverzeichnis und überprüfen Sie die Verzeichnisstruktur. Der absolute Pfad für das Installationsskript sollte beispielsweise */ExamplePackage_targetPlatform*`/install.sh` sein.

Fehler: Manifest konnte nicht heruntergeladen werden: Dokument mit dem Namen ist nicht vorhanden

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Failed to download manifest - failed to retrieve package document description:
 InvalidDocument: Document with name filename does not exist.
```

Ursache: Distributor kann das Paket nicht anhand des Paketnamens finden, wenn ein Distributor-Paket von einem anderen Konto geteilt wird.

Lösung: Wenn Sie ein Paket von einem anderen Konto freigeben, verwenden Sie den vollständigen Amazon-Ressourcennamen (ARN) für das Paket und nicht nur den Namen.

Hochladen fehlgeschlagen.

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Upload failed. At least one of your files was not successfully uploaded to your S3
 bucket.
```

Ursache: Der Name Ihres Softwarepakets enthält ein Leerzeichen. Zum Beispiel würde bei `Hello World.msi` der Upload fehlschlagen.

AWS Systems Manager Gemeinsam genutzte Ressourcen

Systems Manager verwendet die folgenden freigegebenen Ressourcen für die Verwaltung und Konfiguration Ihrer AWS -Ressourcen.

Themen

- AWS Systems Manager-Documents

AWS Systems Manager-Documents

Ein AWS Systems Manager-Dokument (SSM-Dokument) definiert die Aktionen, die Systems Manager auf Ihren verwalteten Instances durchführt. Systems Manager umfasst mehr als 100 vorkonfigurierter Dokumente, die Sie verwenden können, indem Sie zur Laufzeit Parameter angeben. Vorkonfigurierte Dokumente finden Sie in der Systems-Manager-Dokumentenkonsole, indem Sie die Registerkarte Owned by Amazon (Eigentum von Amazon) auswählen. Alternativ können Sie beim Aufrufen des API-Vorgangs `Owner` für den Filter `ListDocuments` Amazon angeben. Die Dokumente liegen im JavaScript Object Notation (JSON)- oder YAML-Format vor und enthalten die von Ihnen angegebenen Schritte und Parameter. Um mit SSM-Dokumenten zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.

Wie kann meine Organisation von der Documents-Funktion profitieren?

Documents, eine Funktion von AWS Systems Manager, bietet die folgenden Vorteile:

- Kategorien von Dokumenten

  Zur einfacheren Suche nach den benötigten Dokumenten wählen Sie je nach Typ des gesuchten Dokuments eine Kategorie aus. Um die Suche zu erweitern, können Sie mehrere Kategorien desselben Dokumenttyps auswählen. Die Auswahl von Kategorien verschiedener Dokumenttypen wird nicht unterstützt. Kategorien werden nur für Dokumente im Besitz von Amazon unterstützt.

- Dokumentversionen

  Sie können unterschiedliche Versionen von Dokumenten erstellen und speichern. Anschließend können Sie eine Standardversion für jedes Dokument angeben. Die Standardversion eines Dokuments kann auf eine neuere Version aktualisiert und wieder auf eine ältere Version

zurückgesetzt werden. Wenn Sie den Inhalt eines Dokuments ändern, inkrementiert Systems Manager automatisch die Versionsnummer des Dokuments. Sie können eine beliebige Version eines Dokuments abrufen oder verwenden, indem Sie die Dokumentversion in der Konsole, in AWS CLI-Befehlen (AWS Command Line Interface) oder API-Aufrufen angeben.

- Anpassen von Dokumenten an die eigenen Bedürfnisse

  Wenn Sie die Schritte und Aktionen in einem Dokument anpassen möchten, können Sie Ihre eigenen Dokumente erstellen. Das System speichert das Dokument mit Ihrem AWS-Konto in der AWS-Region, in der Sie es erstellen. Weitere Informationen zum Erstellen eines SSM-Dokuments finden Sie unter Erstellen von SSM-Dokumentinhalten.

- Markieren von Dokumenten

  Sie können Ihre Dokumente markieren, um sie später anhand der zugewiesenen Tags schnell zu identifizieren. Beispielsweise können Sie Dokumente für bestimmte Umgebungen, Abteilungen, Benutzer, Gruppen oder Zeiträume markieren. Sie können auch den Zugriff auf Dokumente einschränken, indem Sie eine AWS Identity and Access Management (IAM)-Richtlinie erstellen, die festlegt, auf welche Tags Benutzer oder Gruppen zugreifen können. Weitere Informationen finden Sie unter Markierungen von Systems Manager-Dokumenten.

- Freigeben von Dokumenten

  Sie können Ihre Dokumente öffentlich zugänglich machen oder für bestimmte AWS-Konten in derselben AWS-Region freigeben. Das Freigeben von Dokumenten für mehrere Konten kann sinnvoll sein, wenn z. B. alle Amazon Elastic Compute Cloud (Amazon EC2)-Instances, die Sie Kunden oder Mitarbeitenden zur Verfügung stellen, die gleiche Konfiguration aufweisen sollen. Möglicherweise möchten Sie nicht nur Anwendungen oder Patches auf den Instances auf dem neuesten Stand halten, sondern auch bestimmte Aktivitäten von Kunden-Instances beschränken. Oder Sie möchten sicherstellen, dass Instances, die von Mitarbeiterkonten organisationsweit genutzt werden, auf bestimmte interne Ressourcen zugreifen können. Weitere Informationen finden Sie unter Freigeben von SSM-Dokumenten.

Wer sollte Documents verwenden?

- AWS-Kunden, die mithilfe von Systems-Manager-Funktionen ihre betriebliche Effizienz im großen Maßstab verbessern, Fehler im Zusammenhang mit manuellen Eingriffen reduzieren und die Zeit bis zur Lösung häufig auftretender Probleme verkürzen möchten.
- Infrastrukturexperten, die Bereitstellungs- und Konfigurationsaufgaben automatisieren möchten.

- Administratoren, die häufig auftretende Probleme zuverlässig lösen, die Effizienz bei der Fehlerbehebung verbessern und die Anzahl sich wiederholender Vorgänge reduzieren möchten.
- Benutzer, die eine Aufgabe automatisieren möchten, die sie normalerweise manuell ausführen.

Welche Typen von SSM-Dokumenten gibt es?

Die folgende Tabelle beschreibt die verschiedenen Arten von SSM-Dokumenten und ihre jeweilige Nutzung.

Typ	Verwendet mit	Details
ApplicationConfiguration ApplicationConfigurationSchema	AWS AppConfig	AWS AppConfig, eine Funktion von AWS Systems Manager, ermöglicht es Ihnen, Anwendungskonfigurationen zu erstellen, zu verwalten und schnell bereitzustellen. Sie können Konfigurationsdaten in einem SSM-Dokument speichern, indem Sie ein Dokument erstellen, das den Dokumenttyp `ApplicationConfiguration` verwendet. Weitere Informationen finden Sie unter Freeform configurations (Freiform-Konfigurationen) im AWS AppConfig Benutzerhandbuch. Wenn Sie eine Konfiguration in einem SSM-Dokument erstellen, müssen Sie ein entsprechendes JSON-Schema angeben. Das Schema verwendet den `ApplicationConfigurationSch`

Typ	Verwendet mit	Details
		`ema` -Dokumenttyp und definiert wie ein Regelsatz die zulässigen Eigenschaften für jede Anwendungskonfigurationseinstellung. Weitere Informationen finden Sie unter About validators (Informationen zu Validatoren) im AWS AppConfig-Benutzerhandbuch.

Typ	Verwendet mit	Details
Automation-Runbook	Automation State Manager Maintenance Windows	Verwenden Sie Automation-Runbooks, wenn Sie allgemeine Wartungs- und Bereitstellungsaufgaben durchführen, wie z. B. das Erstellen oder Aktualisieren eines Amazon Machine Image (AMI). State Manager verwendet Automation-Runbooks, um eine Konfiguration anzuwenden. Diese Aktionen können während des Lebenszyklus einer Instance jederzeit für ein oder mehrere Ziele ausgeführt werden. Maintenance Windows verwendet Automation-Runbooks, um basierend auf dem angegebenen Zeitplan allgemeine Wartungs- und Bereitstellungsaufgaben durchzuführen. Alle Automation-Runbooks, die für Linux-basierte Betriebssysteme unterstützt werden, werden auch auf EC2-Instances für macOS unterstützt.

Typ	Verwendet mit	Details
Kalenderdokument ändern	Change Calendar	Change Calendar, eine Funktion von AWS Systems Manager, verwendet den `ChangeCalendar` -Dokumenttyp. Ein Change Calendar-Dokument speichert einen Kalendereintrag und zugehörige Ereignisse, die es ermöglichen oder verhindern können, dass Automation-Aktionen Ihre Umgebung verändern. In Change Calendar speichert ein Dokument iCalendar 2.0-Daten im Klartextformat. Change Calendar wird auf EC2-Instances für macOS nicht unterstützt.

Typ	Verwendet mit	Details
AWS CloudFormation-Vorlage	AWS CloudFormation	Diese AWS CloudFormation-Vorlagen beschreiben die Ressourcen, die Sie in Ihren CloudFormation-Stacks bereitstellen möchten. Wenn Sie CloudFormation-Vorlagen als Systems-Manager-Dokumente speichern, können Sie von den Dokumentfeatures von Systems Manager profitieren. Dazu gehören das Erstellen und Vergleichen mehrerer Versionen Ihrer Vorlage und das Freigeben Ihrer Vorlage für andere Konten in derselben AWS-Region. Sie können CloudFormation-Vorlagen und Stacks erstellen und bearbeiten, indem Sie Application Manager, eine Funktion von Systems Manager, verwenden. Weitere Informationen finden Sie unter Arbeiten mit AWS CloudFormation-Vorlagen und Stacks in Application Manager.

Typ	Verwendet mit	Details
Befehlsdokument	Run Command State Manager Maintenance Windows	Run Command, eine Funktion von AWS Systems Manager, verwendet Befehlsdokumente, um Befehle auszuführen. State Manager, eine Funktion von AWS Systems Manager, verwendet Befehlsdokumente, um eine Konfiguration zu übernehmen. Diese Aktionen können während des Lebenszyklus einer Instance jederzeit für ein oder mehrere Ziele ausgeführt werden. Maintenance Windows, eine Funktion von AWS Systems Manager, verwendet Befehlsdokumente, um basierend auf dem angegeben en Zeitplan eine Konfiguration zu übernehmen. Die meisten Befehlsdo kumente werden unter allen Linux- und Windows Server-Betriebssystemen von Systems Manager unterstüt zt. Die folgenden Befehlsdo kumente werden auf EC2-Instances für macOS unterstüt zt: • `AWS-ConfigureAWSPa ckage` • `AWS-RunPatchBaseli ne`

Typ	Verwendet mit	Details
		• `AWS-RunPatchBaselineAssociation` • `AWS-RunShellScript`
AWS Config-Konformität spaketvorlage	AWS Config	AWS Config-Konformität spaketvorlagen sind Dokumente im YAML-Format, die zum Erstellen von Konformitätspakete n verwendet werden, welche die Liste der AWS Config-verwalteten oder benutzerdefinierten Regeln und Korrekturmaßnahmen enthalten. Weitere Informationen finden Sie unter Konformitätspakete.
Paketdokument	Distributor	In Distributor, eine Funktion von AWS Systems Manager, wird ein Paket wird durch ein SSM-Dokument dargestellt. Ein Paketdokument enthält angefügte ZIP-Archivdateien mit Software oder Ressourcen zur Installation auf verwalteten Instances. Durch die Erstellun g eines Pakets in Distribut or wird das Paketdokument erstellt. Distributor wird unter Oracle Linux und macOS verwalteten Instances nicht unterstützt.

Typ	Verwendet mit	Details
Richtliniendokument	State Manager	Inventory, eine Funktion von AWS Systems Manager, verwendet das `AWS-GatherSoftwareInventory` -Richtliniendokument mit einer State Manager-Zuordnung, um Bestandsdaten von verwalteten Instances zu erfassen. Beim Erstellen eigener SSM-Dokumente sind Automation-Runbooks und Command-Dokumente die bevorzugte Methode zum Durchsetzen einer Richtlinie auf einer verwalteten Instance. Systems Manager Inventory und `AWS-GatherSoftwareInventory` -Richtliniendokument werden auf allen Betriebssystemen unterstützt, die von Systems Manager unterstützt werden.

Typ	Verwendet mit	Details
Vorlage für die Analyse nach einem Vorfall	Incident Manager-Analyse nach einem Vorfall	Incident Manager verwendet die Analysevorlage nach einem Vorfall, um eine Analyse basierend auf AWS bewährten Methoden der Betriebsverwaltung zu erstellen. Erstellen Sie mithilfe der Vorlage eine Analyse, mit der Ihr Team Verbesserungen für die Reaktion auf Vorfälle ermitteln kann.

Typ	Verwendet mit	Details
Sitzungsdokument	Session Manager	Session Manager, eine Funktion von AWS Systems Manager, verwendet Sitzungsdokumente, um zu bestimmen, welche Art von Sitzung gestartet werden soll, z. B. eine Port-Weiterleitung ssitzung, eine Sitzung zum Ausführen eines interaktiven Befehls oder eine Sitzung zum Erstellen eines SSH-Tunnels. Befehlsdokumente werden unter allen Linux- und Windows Server-Betriebssys temen von Systems Manager unterstützt. Die folgenden Befehlsdokumente werden auf EC2-Instances für macOS unterstützt: • `AWS-PasswordReset` • `AWS-StartInteracti veCommand` • `AWS-StartPortForwa rdingSession` • `AWS-StartPortForwa rdingSessionToSock et` • `AWS-StartSSHSession`

SSM-Dokumentkontingente

Informationen zu SSM-Dokumentkontingenten finden Sie unter Systems Manager Service Quotas in der Allgemeine Amazon Web Services-Referenz.

Themen

- Dokument-Komponenten
- Erstellen von SSM-Dokumentinhalten
- Arbeiten mit Dokumenten

Dokument-Komponenten

Dieser Bereich enthält Informationen zu den Komponenten, aus denen sich SSM-Dokumente zusammensetzen.

Inhalt

- Schemata, Features und Beispiele
- Datenelemente und Parameter
- Referenz für Befehlsdokument-Plug-ins

Schemata, Features und Beispiele

AWS Systems Manager (SSM)-Dokumente verwenden die folgenden Schema-Versionen.

- Dokumente des Typs `Command` können die Schema-Versionen 1.2, 2.0 und 2.2 verwenden. Wenn Sie Schema 1.2-Dokumente verwenden, empfehlen wir, dass Sie Dokumente erstellen, die Schema-Version 2.2 verwenden.
- Dokumente des Typs `Policy` müssen Schema-Version 2.0 oder höher verwenden.
- Dokumente des Typs `Automation` müssen Schema-Version 0.3 verwenden.
- Sie können Dokumente im JSON- oder YAML-Format erstellen.

Durch die Verwendung der neuesten Schema-Version `Command`- und `Policy`-Dokumente können Sie die folgenden Features nutzen.

Features für Schema-Version 2.2-Dokumente

Feature	Details
Dokumentbearbeitung	Dokumente können jetzt aktualisiert werden. Bei Version 1.2 mussten aktualisierte Dokument unter einem anderen Namen gespeichert werden.
Automatisches Versioning	Bei jeder Änderung an einem Dokument wird eine neue Version erstellt. Dies ist kein Schema-Version, sondern eine Version des Dokuments.
Standardversion	Wenn Sie mehrere Versionen eines Dokuments haben, können Sie festlegen, welche Version das Standarddokument ist.
Sequenzierung	Plug-ins oder Schritte in einem Dokument werden in der Reihenfolge ausgeführt, die Sie angegeben haben.
Unterstützung für plattformübergreifende Anweisungen	Die Unterstützung für plattformübergreifende Anweisungen ermöglicht die Angabe unterschi edlicher Betriebssysteme für verschiedene Plugins innerhalb desselben SSM-Dokum ents. Plattformübergreifende Anweisungen verwenden in einem Schritt den Parameter `precondition` .

> **Note**
>
> Sie müssen AWS Systems Manager SSM Agent auf Ihren Instances immer auf die neueste Version aktualisieren, um die neuen Systems Manager-Features und SSM-Dokumentfunktionen nutzen zu können. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

In der folgenden Tabelle finden Sie die Unterschiede zwischen de Schema-Hauptversionen.

Version 1.2	Version 2.2 (neueste Version)	Details
runtimeConfig	mainSteps	In Version 2.2 ersetzt der Abschnitt `mainSteps` `runtimeConfig` . Im Abschnitt `mainSteps` erlaubt Systems Manager das Ausführen von nacheinander folgenden Schritten.
Eigenschaften	inputs	In Version 2.2 ersetzt der Abschnitt `inputs` den Abschnitt `properties` . Der Abschnitt `inputs` nimmt Parameter für Schritte entgegen.
commands	runCommand	In Version 2.2 ersetzt im Abschnitt `inputs` der Parameter `runCommand` den Parameter `commands`.
ID	action	In Version 2.2 ersetzt `Action ID`. Dies ist lediglich eine Umbenennung.
n.v.	Name	In Version 2.2 ist `name` ein benutzerdefinierter Name für einen Schritt.

Verwenden des Parameters „precondition“

Bei Schema-Version 2.2 oder neuer können Sie mithilfe des Parameters `precondition` das Zielbetriebssystem für jedes Plug-in angeben oder um Eingabeparameter zu validieren, die Sie in Ihrem SSM-Dokument definiert haben. Der `precondition`-Parameter unterstützt die Referenzierung der Eingabeparameter Ihres SSM-Dokuments und `platformType` unter

Verwendung von Werten von `Linux`, `MacOS` und `Windows`. Nur der `StringEquals`-Operator wird unterstützt.

Wenn bei Dokumenten in Schema-Version 2.2 oder höher `precondition` nicht angegeben ist, werden Plug-ins entweder ausgeführt oder übersprungen, je nachdem, ob das Plug-in mit dem jeweiligen Betriebssystem kompatibel ist. Plugin-Kompatibilität mit dem Betriebssystem wird vor der `precondition` ausgewertet. Bei Dokumenten, die Schema-Version 2.0 oder eine frühere Version verwenden, wird bei nicht kompatiblen Plug-ins ein Fehler ausgelöst.

Wenn beispielsweise in einem Schema-Version 2.2-Dokument `precondition` nicht angegeben ist und das `aws:runShellScript`-Plugin zur Ausführung aufgelistet ist, wird der Schritt auf Linux-Instances ausgeführt, aber auf Windows Server-Instances übersprungen, da `aws:runShellScript` nicht kompatibel mit Windows Server-Instances ist. Bei Schema-Version 2.0 Dokumenten schlägt jedoch die Ausführung fehl, wenn Sie das `aws:runShellScript`-Plugin angeben und dann das Dokument auf einer Windows Server-Instance ausführen. Weiter hinten in diesem Abschnitt finden Sie ein Beispiel der Vorbedingungsparameter in SSM-Dokumenten.

Schema der Version 2.2

Top-Level-Elemente

Das folgende Beispiel zeigt die Elemente der obersten Ebene eines SSM-Dokuments bei Verwendung von Schema-Version 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 }}"
         }
      }
   ]
}
```

Schema-Version 2.2 -Beispiel

Im folgenden Beispiel wird das `aws:runPowerShellScript`-Plugin verwendet, um einen PowerShell-Befehl auf den Ziel-Instances auszuführen.

YAML

```
---
schemaVersion: "2.2"
description: "Example document"
parameters:
  Message:
```

```
    type: "String"
    description: "Example parameter"
    default: "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"
      }
   },
   "mainSteps": [
      {
         "action": "aws:runPowerShellScript",
         "name": "example",
         "inputs": {
            "timeoutSeconds": "60",
            "runCommand": [
               "Write-Output {{Message}}"
            ]
         }
      }
   ]
}
```

Schema der Version 2.2 – Vorbedingungsparameterbeispielen

Schema-Version 2.2 bietet Unterstützung für plattformübergreifende Aktionen. Dies bedeutet, dass Sie in einem SSM-Dokument unterschiedliche Betriebssysteme für verschiedene Plugins angeben können. Plattformübergreifende Aktionen werden durch den Parameter `precondition` in einem

Schritt aufgerufen, wie in dem folgenden Beispiel dargestellt. Sie können auch den `precondition`-Parameter verwenden, um Eingabeparameter zu validieren, die Sie in Ihrem SSM-Dokument definiert haben. Dies sehen Sie im zweiten der folgenden Beispiele.

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"
            ]
         }
      }
   ]
}
```

Schema-Version 2.2 State Manager-Beispiel

Sie können das folgende SSM-Dokument mit State Manager, eine Funktion von Systems Manager, nutzen, um die ClamAV-Antivirensoftware herunterzuladen und zu installieren. State Manager erzwingt eine bestimmte Konfiguration, d. h. jedes Mal, wenn die State Manager-Zuordnung ausgeführt wird, prüft das System, ob die ClamAV-Software installiert ist. Ist dies nicht der Fall, führt State Manager dieses Dokument erneut aus.

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"
            ]
         }
      }
   ]
}
```

Schema Version 2.2 - Bestandsbeispiel

Sie können das folgende SSM-Dokument mit State Manager verwenden, um Bestandsmetadaten zu Ihren Instances zu erfassen.

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 }}"
         }
      }
   ]
```

```
}
```

Schema-Version 2.2 **AWS-ConfigureAWSPackage**-Beispiel

Das folgende Beispiel zeigt das `AWS-ConfigureAWSPackage`-Dokument. Der Abschnitt `mainSteps` enthält das `aws:configurePackage`-Plugin im Schritt `action`.

> **Note**
>
> In Linux-Betriebssystemen werden nur die `AmazonCloudWatchAgent`- und `AWSSupport-EC2Rescue`-Pakete unterstützt.

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 }}"
         }
      }
   ]
}
```

Schema der Version 1.2

Das folgende Beispiel zeigt die Elemente der obersten Ebene eines Dokuments in Schema-Version 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"
            }
         ]
      }
   }
}
```

Schema-Version 1.2 **`aws:runShellScript`**-Beispiel

Das folgende Beispiel zeigt das `AWS-RunShellScript` SSM-Dokument. Der Abschnitt runtimeConfig bindet das Plugin `aws:runShellScript` ein.

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

Schema der Version 0.3

Top-Level-Elemente

Im folgenden Beispiel werden die Elemente der obersten Ebene eines Automation-Runbook der Schema-Version 0.3im JSON-Format gezeigt.

```
{
    "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
        }
    }
}
```

Beispiel für YAML-Automation-Runbook

Das folgende Beispiel zeigt den Inhalt eines Automation-Runbooks im YAML-Format. In diesem funktionierenden Beispiel der Version 0.3 des Dokumentschemas wird auch die Verwendung von Markdown zur Formatierung von Dokumentbeschreibungen veranschaulicht.

```
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
```

Datenelemente und Parameter

In diesem Thema werden die in SSM-Dokumenten verwendeten Datenelemente beschrieben. Die Schemaversion, die zum Erstellen eines Dokuments verwendet wird, definiert die Syntax und die Datenelemente, die das Dokument akzeptiert. Es wird empfohlen, Schema-Version 2.2 oder höher für Befehlsdokumente zu verwenden. Automation-Runbooks verwenden die Schema-Version 0.3. Automation-Runbooks unterstützen darüber hinaus die Verwendung von Markdown, einer Markup-Sprache, mit der Sie Wiki-Beschreibungen zu Dokumenten und einzelnen Schritten innerhalb des Dokuments hinzufügen können. Weitere Informationen zur Verwendung von Markdown finden Sie unter Verwenden von Markdown in der Konsole im AWS Management Console -Handbuch „Erste Schritte“.

Im folgenden Abschnitt werden die Datenelemente beschrieben, die Sie in ein SSM-Dokument aufnehmen können.

Top-Level-Datenelemente

schemaVersion

Die zu verwendende Schema-Version.

Typ: Version

Erforderlich: Ja

description

Von Ihnen angegebene Informationen, um den Zweck des Dokuments zu beschreiben. Sie können dieses Feld auch verwenden, um anzugeben, ob ein Parameter einen Wert für die Ausführung eines Dokuments benötigt oder ob die Bereitstellung eines Werts für den Parameter optional ist. Erforderliche und optionale Parameter sind in den Beispielen dieses Themas zu sehen.

Typ: Zeichenfolge

Erforderlich: Nein

Parameter

Eine Struktur, die die Parameter definiert, die das Dokument akzeptiert.

Für Parameter, die Sie häufig verwenden, empfehlen wir, diese Parameter in Parameter Store einer Fähigkeit von zu speichern AWS Systems Manager. Anschließend können Sie Parameter in Ihrem Dokument definieren, die Parameter Store-Parameter als Standardwert referenzieren. Um auf einen Parameter Store-Parameter zu verweisen, verwenden Sie die folgende Syntax.

```
{{ssm:parameter-name}}
```

Sie können einen Parameter, der auf einen Parameter Store-Parameter verweist, auf die gleiche Weise wie alle anderen Dokumentparameter verwenden. Im folgenden Beispiel ist der Standardwert für den `commands`-Parameter der Parameter Store-Parameter `myShellCommands`. Durch Angabe des `commands`-Parameters als `runCommand`-Zeichenfolge führt das Dokument die im `myShellCommands`-Parameter gespeicherten Befehle aus.

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 }}"]
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 }}"]
      }
    },
    "mainSteps": [
      {
        "action": "aws:runShellScript",
        "name": "runShellScriptDefaultParams",
        "inputs": {
            "runCommand": [
              "{{ commands }}"
          ]
        }
      }
    ]
  }
```

> **Note**
>
> Im `parameters`-Abschnitt Ihres Dokuments können Sie die -Parameter `String` und `StringList` Parameter Store referenzieren. Sie können nicht die Parameter `SecureString` Parameter Store referenzieren.

Mehr über Parameter Store erfahren Sie unter AWS Systems Manager Parameter Store.

Typ: Struktur

Die `parameters`-Struktur akzeptiert die folgenden Felder und Werte:

- `type`: (Erforderlich) Zulässige Werte umfassen die Folgenden: `String`, `StringList`, `Integer`, `Boolean`, `MapList` und `StringMap`. Beispiele für jeden Typ finden Sie Beispiele für den Parameter `type` in SSM-Dokumenten im nächsten Abschnitt.

  > **Note**
  >
  > Befehlstyp-Dokumente unterstützen nur die Parametertypen `String` und `StringList`.

- `description`: (Optional) Eine Beschreibung der Parametergruppe.

- `default`: (Optional) Der Standardwert des Parameters oder eine Referenz bezüglich eines Parameters in Parameter Store.
- `allowedValues`: (Optional) Ein Array von Werten, die für den Parameter zulässig sind. Durch das Definieren zulässiger Werte für den Parameter wird die Benutzereingabe überprüft. Wenn ein Benutzer einen Wert eingibt, der nicht zulässig ist, kann die Ausführung nicht gestartet werden.

  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`: (Optional) Ein regulärer Ausdruck, der überprüft, ob die Benutzereingabe mit dem definierten Muster für den Parameter übereinstimmt. Wenn die Benutzereingabe nicht mit dem zulässigen Muster übereinstimmt, kann die Ausführung nicht gestartet werden.

  > **Note**
  >
  > Systems Manager führt zwei Validierungen für `allowedPattern` aus. Die erste Validierung erfolgt unter Verwendung der Java regex library (Java-Regex-Bibliothek) auf API-Ebene, wenn Sie ein Dokument verwenden. Die zweite Validierung wird am SSM

Agent ausgeführt, indem Sie die GO regexp library (GO-regexp-Bibliothek) verwenden, bevor Sie das Dokument bearbeiten.

YAML

```
InstanceId:
  type: String
  description: "(Required) The instance ID to target."
  allowedPattern: "^i-[a-z0-9]{8,17}$"
  default: ''
```

JSON

```
"InstanceId": {
  "type": "String",
  "description": "(Required) The instance ID to target.",
  "allowedPattern": "^i-[a-z0-9]{8,17}$",
  "default": ""
}
```

- `displayType`: (Optional) Wird verwendet, um entweder a `textfield` oder a `textarea` in der anzuzeigen AWS Management Console. `textfield`dist ein einzeiliges Textfeld. `textarea`ist ein mehrzeiliger Textbereich.
- `minItems`: (Optional) Die minimal zulässige Anzahl von Elementen.
- `maxItems`: (Optional) Die maximal zulässige Anzahl von Elementen.
- `minChars`: (Optional) Die minimal zulässige Anzahl an Parameterzeichen.
- `maxChars`: (Optional) Die maximal zulässige Anzahl an Parameterzeichen.

Erforderlich: Nein

variables

(Nur Schemaversion 0.3) Werte, auf die Sie während der einzelnen Schritte in einem Automation-Runbook verweisen oder diese aktualisieren können. Variablen ähneln Parametern, unterscheiden sich jedoch in einem sehr wichtigen Punkt. Parameterwerte sind im Kontext eines Runbooks statisch, aber die Werte von Variablen können im Kontext des Runbooks geändert werden. Beim Aktualisieren des Werts einer Variable muss der Datentyp dem definierten Datentyp

entsprechen. Hinweise zum Aktualisieren von Variablenwerten in einer Automatisierung finden Sie unter `aws:updateVariable` – Aktualisiert einen Wert für eine Runbook-Variable.

Typ: Boolean | Integer | | Zeichenfolge | MapList | StringList StringMap

Erforderlich: Nein

YAML

```
variables:
    payload:
        type: StringMap
        default: "{}"
```

JSON

```
{
    "variables": [
        "payload": {
            "type": "StringMap",
            "default": "{}"
        }
    ]
}
```

runtimeConfig

(Nur für Schemaversion 1.2) Die Konfiguration für die Instance, wie sie von mindestens einem Systems Manager-Plug-In verwendet wird. Es wird nicht garantiert, dass Plug-Ins nacheinander ausgeführt werden.

Typ: Dictionary<String, > PluginConfiguration

Erforderlich: Nein

mainSteps

(Nur Schema-Version 0.3, 2.0 und 2.2) Ein Objekt, das mehrere Schritte (Plugins) enthalten kann. Plugins werden innerhalb von Schritten definiert. Die Schritte werden in der Reihenfolge ausgeführt, in der sie im Dokument aufgeführt sind.

Typ: Dictionary<String, > PluginConfiguration

Erforderlich: Ja

outputs

(Nur Schema-Version 0.3) Daten, die durch die Ausführung dieses Dokuments generiert werden, die in anderen Prozessen verwendet werden können. Wenn Ihr Dokument beispielsweise ein neues erstelltAMI, können Sie "angeben. CreateImage ImageId"als Ausgabewert und verwenden Sie diese Ausgabe dann, um in einer nachfolgenden Automatisierungsausführung neue Instanzen zu erstellen. Weitere Informationen zu Ausgaben finden Sie unter Verwenden von Aktionsausgaben als Eingaben.

Geben Sie ein: Dictionary<String, > OutputConfiguration

Erforderlich: Nein

files

(Nur Schema-Version 0.3) Die Skriptdateien (und ihre Prüfsummen), die dem Dokument zugeordnet sind und während einer Automatisierungsausführung ausgeführt werden. Gilt nur für Dokumente, die die `aws:executeScript` Aktion enthalten und für die Anfügungen in einem oder mehreren Schritten angegeben wurden.

Für die Unterstützung der Skript-Laufzeit unterstützen Automation-Runbooks Skripts für Python 3.7, Python 3.8, PowerShell Core 6.0 und PowerShell 7.0. Weitere Informationen zum Einbinden von Skripten in Automation-Runbooks finden Sie unter Verwenden von Skripten in Runbooks und Verwenden von Document Builder zur Erstellung von Runbooks.

Wenn Sie ein Automatisierungs-Runbook mit Anlagen erstellen, müssen Sie auch Anhangsdateien mit der `--attachments` Option (für AWS CLI) oder `Attachments` (für API und SDK) angeben. Sie können den Dateispeicherort sowohl für lokale Dateien als auch für Dateien festlegen, die in Amazon Simple Storage Service (Amazon S3)-Buckets gespeichert sind. Weitere Informationen finden Sie in der AWS Systems Manager API-Referenz unter Anlagen.

YAML

```
---
files:
  launch.py:
    checksums:
      sha256: 18871b1311b295c43d0f...
[truncated]...772da97b67e99d84d342ef4aEXAMPLE
```

JSON

```
"files": {
    "launch.py": {
        "checksums": {
            "sha256": "18871b1311b295c43d0f...
[truncated]...772da97b67e99d84d342ef4aEXAMPLE"
        }
    }
}
```

Geben Sie ein: Dictionary<String, > FilesConfiguration

Erforderlich: Nein

Beispiele für den Parameter **`type`** in SSM-Dokumenten

Parametertypen in SSM-Dokumenten sind statisch. Dies bedeutet, dass der Parametertyp nicht geändert werden kann, nachdem er definiert wurde. Bei der Verwendung von Parametern mit SSM-Dokumenten-Plug-ins kann der Typ eines Parameters innerhalb der Eingabe eines Plug-ins nicht dynamisch geändert werden. Beispielsweise können Sie nicht auf einen `Integer`-Parameter innerhalb der Eingabe `runCommand` des Plug-ins `aws:runShellScript` verweisen, da diese Eingabe eine Zeichenfolge oder eine Liste von Zeichenfolgen akzeptiert. Um einen Parameter für eine Plug-in-Eingabe verwenden zu können, muss der Parametertyp mit dem akzeptierten Typ übereinstimmen. Sie müssen beispielsweise einen Parameter des Typs `Boolean` für die Eingabe `allowDowngrade` des Plug-ins `aws:updateSsmAgent` angeben. Wenn der Parametertyp nicht mit dem Eingabetyp für ein Plug-in übereinstimmt, kann das SSM-Dokument nicht validiert werden und das System erstellt das Dokument nicht. Dies gilt auch, wenn Parameter nachgeschaltet in Eingaben für andere Plugins oder Automatisierungsaktionen verwendet werden. AWS Systems Manager Sie können beispielsweise keinen `StringList`-Parameter in der `documentParameters`-Eingabe des Plug-Ins `aws:runDocument` referenzieren. Die `documentParameters`-Eingabe akzeptiert eine Zuordnung von Zeichenfolgen, auch wenn der nachgelagerte Parametertyp des SSM-Dokuments ein `StringList`-Parameter ist und dem referenzierten Parameter entspricht.

Wenn Sie Parameter mit -Automation-Aktionen verwenden, werden Parametertypen bei der Erstellung des SSM-Dokuments in den meisten Fällen nicht validiert. Nur wenn Sie die Aktion `aws:runCommand` verwenden, werden Parametertypen bei der Erstellen des SSM-Dokuments validiert. In allen anderen Fällen erfolgt die Parametervalidierung während der Automatisierungsausführung, wenn die Eingabe einer Aktion überprüft wird, bevor die Aktion

ausgeführt wird. Wenn der Eingabeparameter beispielsweise ein `String` ist und Sie auf ihn als Wert für die Eingabe `MaxInstanceCount` der Aktion `aws:runInstances` verweisen, wird das SSM-Dokument erstellt. Beim Ausführen des Dokuments schlägt die Automatisierung jedoch fehl, wenn die Aktion `aws:runInstances` validiert wird, da für die Eingabe `MaxInstanceCount` ein `Integer` erforderlich ist.

Im Folgenden finden Sie für jeden Parameter`type` ein Beispiel.

String

Eine Abfolge von null oder mehr Unicode-Zeichen in Anführungszeichen. Zum Beispiel "i-1234567890abcdef0". Verwenden Sie umgekehrte Schrägstriche als Escapezeichen.

YAML

```
---
InstanceId:
  type: String
  description: "(Optional) The target EC2 instance ID."
```

JSON

```
"InstanceId":{
  "type":"String",
  "description":"(Optional) The target EC2 instance ID."
}
```

StringList

Eine Liste von String-Elementen, die durch Kommas getrennt sind. Zum Beispiel ["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"
}
```

Boolesch

Akzeptiert nur `true` oder `false`. Akzeptiert nicht „true“ oder 0.

YAML

```
---
canRun:
  type: Boolean
  description: ''
  default: true
```

JSON

```
"canRun": {
  "type": "Boolean",
  "description": "",
  "default": true
}
```

Ganzzahl

Ganze Zahlen. Akzeptiert keine Dezimalzahlen, z. B. 3,14159, oder Zahlen in Anführungszeichen, z. B. "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

Ein Mapping von Schlüsseln zu Werten. Schlüssel und Werte müssen Zeichenfolgen sein. Zum Beispiel {"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

Eine Liste von StringMap Objekten.

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
}
```

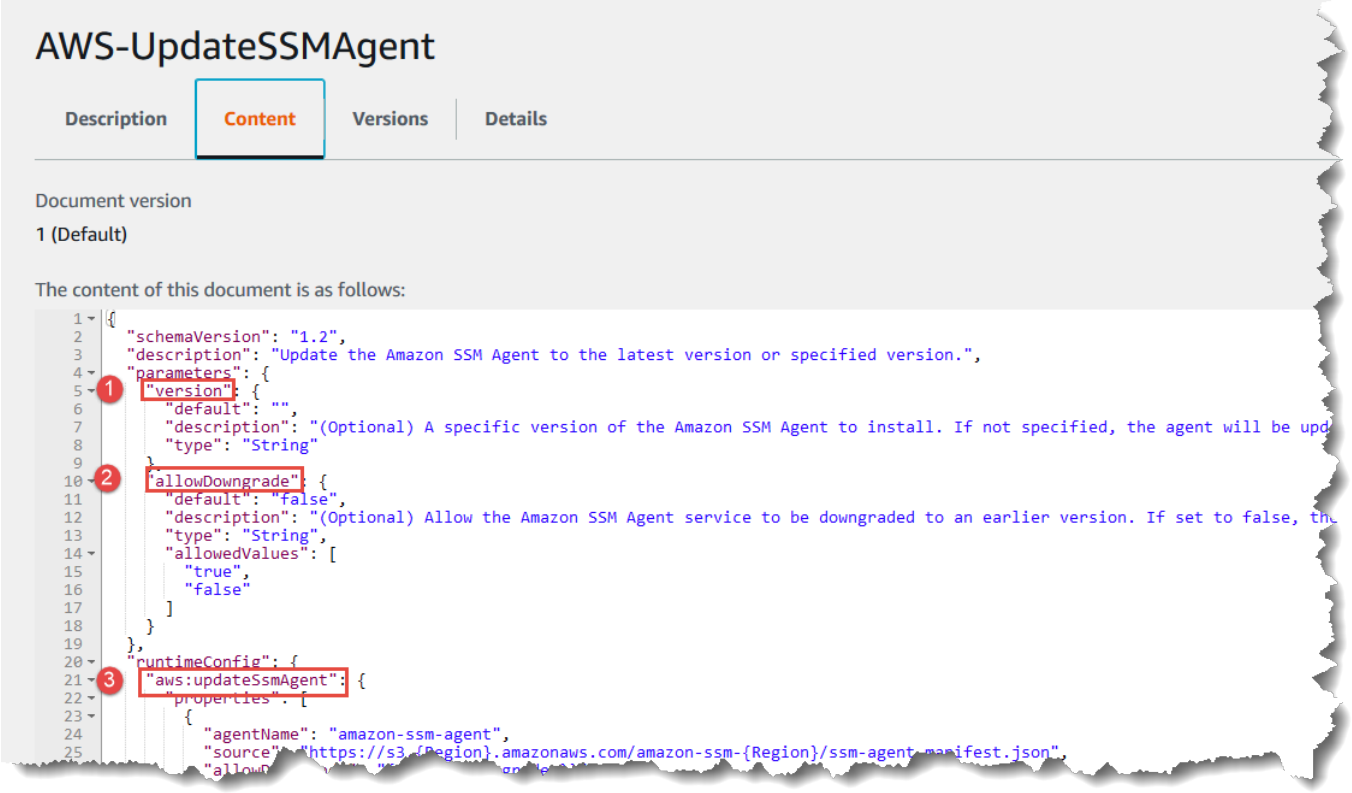
Inhalte von SSM-Befehlsdokument anzeigen

Um eine Vorschau der erforderlichen und optionalen Parameter für ein AWS Systems Manager (SSM-) Befehlsdokument anzuzeigen, können Sie zusätzlich zu den Aktionen, die das Dokument ausführt, den Inhalt des Dokuments in der Systems Manager Manager-Konsole anzeigen.

Inhalte von SSM-Befehlsdokument anzeigen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie im Suchfeld Dokumenttyp und wählen Sie danach Befehl.
4. Wählen Sie den Namen eines Dokuments und dann die Registerkarte Content (Inhalt) aus.
5. Überprüfen Sie im Inhaltsfeld die verfügbaren Parameter und Aktionsschritte für das Dokument.

   Das folgende Image zeigt beispielsweise, dass (1) `version` und (2) `allowDowngrade` optionale Parameter für das `AWS-UpdateSSMAgent`-Dokument sind, und dass die erste Aktion, die vom Dokument ausgeführt wird, (3) `aws:updateSsmAgent` ist.



Referenz für Befehlsdokument-Plug-ins

Diese Referenz beschreibt die Plugins, die Sie in einem Dokument vom Typ AWS Systems Manager (SSM) Command angeben können. Diese Plug-ins können nicht in SSM-Automation-Runbooks verwendet werden, die Automation-Aktionen verwenden. Informationen zu AWS Systems Manager Automatisierungsaktionen finden Sie unterSystems Manager Automation Aktionen-Referenz.

Systems Manager bestimmt die Aktionen, die auf einer verwalteten Instance ausgeführt werden sollen, durch Lesen der Inhalte eines SSM-Dokuments. Jedes Dokument enthält einen Abschnitt zur Ausführung von Code. Abhängig von der Schemaversion des Dokuments umfasst dieser Abschnitt zu Codeausführung ein oder mehrere Plug-Ins oder Schritte. Im Rahmen dieses Hilfethemas werden die Plug-Ins und Schritte als Plug-Ins bezeichnet. Dieser Abschnitt enthält Informationen zu allen Systems Manager-Plug-Ins. Weitere Informationen zu Dokumenten, einschließlich Informationen zum Erstellen von Dokumenten und zu den Unterschiede zwischen Schemaversionen finden Sie unter AWS Systems Manager-Documents.

> **Note**
>
> Manche der hier beschriebenen Plug-Ins funktionieren nur auf Windows Server-Instances oder Linux-Instances. Für alle Plug-Ins werden Plattformabhängigkeiten angegeben.
> Die folgenden Dokumentenplugins werden auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances für macOS unterstützt:
>
> - `aws:refreshAssociation`
> - `aws:runShellScript`
> - `aws:runPowerShellScript`
> - `aws:softwareInventory`
> - `aws:updateSsmAgent`

Inhalt

- Gemeinsame Eingaben
- 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

Gemeinsame Eingaben

Mit SSM Agent Version 3.0.502 und höher, können alle Plugins die folgenden Eingaben verwenden:

finallyStep

Der letzte Schritt, in dem das Dokument ausgeführt werden soll. Wenn diese Eingabe für einen Schritt definiert ist, hat sie Vorrang vor einem `exit`-Wert, der `onFailure`- oder `onSuccess`-Eingängen definiert ist. Damit ein Schritt mit dieser Eingabe erwartungsgemäß ausgeführt wird, muss der Schritt der letzte sein, der in den `mainSteps` Ihres Dokuments ausgeführt wird.

Typ: Boolesch

Zulässige Werte: `true` | `false`

Erforderlich: Nein

onFailure

Wenn Sie diese Eingabe für ein Plugin mit dem Wert `exit` angeben und der Schritt fehlschlägt, spiegelt der Schrittstatus den Fehler wider und das Dokument führt keine weiteren Schritte aus, es sei denn, es wurde ein `finallyStep` definiert. Wenn Sie diese Eingabe für ein Plugin mit dem Wert `successAndExit` angeben und der Schritt fehlschlägt, zeigt der Schrittstatus Erfolg an und das Dokument führt keine weiteren Schritte aus, es sei denn, es wurde ein `finallyStep` definiert.

Typ: Zeichenfolge

Zulässige Werte: `exit` | `successAndExit`

Erforderlich: Nein

onSuccess

Wenn Sie diese Eingabe für ein Plugin angeben und der Schritt erfolgreich ausgeführt wird, führt das Dokument keine weiteren Schritte durch, es sei denn, es wurde ein `finallyStep`definiert.

Typ: Zeichenfolge

Zulässige Werte: `exit`

Erforderlich: Nein

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`

Installieren, Reparieren oder Deinstallieren von Anwendungen auf einer EC2-Instance. Dieses Plug-In läuft nur unter Windows Server-Betriebssystemen.

Syntax

Schema 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 }}"
```

```
      }
    }
  ]
}
```

Schema 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 }}"
            }
         ]
      }
   }
}
```

Eigenschaften

action

Die zu ergreifende Maßnahme.

Type: Zähler

Zulässige Werte: `Install` | `Repair` | `Uninstall`

Erforderlich: Ja

Parameter

Die Parameter für das Installationsprogramm.

Typ: Zeichenfolge

Erforderlich: Nein

Quelle

Die URL der `.msi`-Datei der Anwendung.

Typ: Zeichenfolge

Erforderlich: Ja

sourceHash

Der SHA256-Hashwert der `.msi`-Datei.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:cloudWatch`

Exportieren Sie Daten aus Windows Server Amazon CloudWatch oder Amazon CloudWatch Logs und überwachen Sie die Daten anhand von CloudWatch Metriken. Dieses Plug-In läuft nur unter Windows Server-Betriebssystemen. Weitere Informationen zur Konfiguration der CloudWatch Integration mit Amazon Elastic Compute Cloud (Amazon EC2) finden Sie unter Erfassung von Metriken, Protokollen und Traces mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.

> **Important**
>
> Der Unified CloudWatch Agent wurde SSM Agent als Tool zum Senden von Protokolldaten an Amazon CloudWatch Logs ersetzt. Das SSM Agent-aws:cloudWatch-Plugin wird nicht unterstützt. Wir empfehlen, nur den Unified CloudWatch Agent für Ihre Protokollerfassungsprozesse zu verwenden. Weitere Informationen finden Sie unter den folgenden Themen:
>
> - Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent)
> - Migrieren Sie die Erfassung von Windows Server-Knotenprotokollen auf den CloudWatch Agenten
> - Erfassung von Metriken, Protokollen und Traces mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.

Sie können die folgenden Datentypen exportieren und überwachen:

ApplicationEventProtokollieren

Sendet Daten aus dem Anwendungsereignisprotokoll an CloudWatch Logs.

CustomLogs

Sendet jede textbasierte Protokolldatei an Amazon CloudWatch Logs. Das CloudWatch Plugin erstellt einen Fingerabdruck für Protokolldateien. Anschließend verknüpft das System einen Datenversatz mit jedem Fingerabdruck. Das Plug-In lädt Dateien hoch, wenn Änderungen vorliegen, erfasst den Versatz und verknüpft ihn mit einem Fingerabdruck. Diese Methode wird verwendet, um zu verhindern, dass ein Benutzer das Plug-In aktiviert, den Service mit einem Verzeichnis verknüpft, in dem sich eine große Anzahl von Dateien befindet, und das System alle Dateien hochlädt.

> **Warning**
>
> Hinweis: Falls Ihre Anwendung während der Abfrage Protokolle kürzt oder zu säubern versucht, besteht die Möglichkeit, dass alle Protokolle, die für `LogDirectoryPath` angegeben wurden, Einträge verlieren. Wenn Sie beispielsweise die Größe der Protokolldatei einschränken möchten, erstellen Sie eine neue Protokolldatei, wenn diese Beschränkung erreicht ist, und lassen Sie neue Daten dann in die neue Datei schreiben.

ETW

Sendet ETW-Daten (Event Tracing for Windows) an CloudWatch Logs.

IIS

Sendet IIS-Protokolldaten an CloudWatch Logs.

PerformanceCounter

Sendet Windows-Leistungsindikatoren an CloudWatch. Sie können verschiedene Kategorien auswählen, in die Daten hochgeladen CloudWatch werden sollen. Erstellen Sie für jeden Leistungsindikator, den Sie hochladen möchten, einen PerformanceCounterAbschnitt mit einer eindeutigen ID (z. B. "PerformanceCounter2"," PerformanceCounter 3" usw.) und konfigurieren Sie dessen Eigenschaften.

> **Note**
>
> Wenn das AWS Systems Manager SSM Agent oder das CloudWatch Plugin gestoppt ist, werden die Leistungsindikatordaten nicht protokolliert CloudWatch. Diese Verhaltensweise unterscheidet sich von der von benutzerdefinierten oder von Windows-Event-Protokollen.. In benutzerdefinierten Protokollen und Windows-Ereignisprotokollen werden die SSM Agent Leistungsindikatordaten gespeichert und CloudWatch nach der Verfügbarkeit des CloudWatch Plug-ins hochgeladen.

SecurityEventProtokollieren

Sendet Protokolldaten von Sicherheitsereignissen an CloudWatch Logs.

SystemEventProtokoll

Sendet Daten aus dem Systemereignisprotokoll an CloudWatch Logs.

Sie können die folgenden Ziele für die Daten definieren:

CloudWatch

Das Ziel, an das die Leistungsindikatormetrikdaten gesendet werden. Sie können weitere Abschnitte mit eindeutigen IDs hinzufügen (z. B. "CloudWatch2", CloudWatch 3" usw.) und für jede neue ID eine andere Region angeben, um dieselben Daten an verschiedene Speicherorte zu senden.

CloudWatchLogs

Das Ziel, an das die Protokolldaten gesendet werden. Sie können weitere Abschnitte mit eindeutigen IDs hinzufügen (z. B. "CloudWatchLogs2", CloudWatchLogs 3" usw.) und für jede neue ID eine andere Region angeben, um dieselben Daten an verschiedene Standorte zu senden.

Syntax

```
"runtimeConfig":{
        "aws:cloudWatch":{
            "settings":{
                "startType":"{{ status }}"
            },
            "properties":"{{ properties }}"
        }
    }
```

Einstellungen und Eigenschaften

AccessKey

Ihre -Zugriffsschlüssel-ID Diese Eigenschaft ist erforderlich, wenn Sie die Instance mithilfe einer IAM-Rolle gestartet haben. Diese Eigenschaft kann nicht mit SSM verwendet werden.

Typ: Zeichenfolge

Erforderlich: Nein

CategoryName

Die Leistungsindikatorkategorie von Performance Monitor.

Typ: Zeichenfolge

Erforderlich: Ja

CounterName

Der Name des Leistungsindikators von Performance Monitor.

Typ: Zeichenfolge

Erforderlich: Ja

CultureName

Das Gebietsschema, unter dem der Zeitstempel protokolliert wird. Wenn dieses Feld leer CultureNameist, wird standardmäßig dasselbe Gebietsschema verwendet, das von Ihrer Instanz verwendet wird. Windows Server

Typ: Zeichenfolge

Gültige Werte: Eine Liste der unterstützten Werte finden Sie unter National Language Support (NLS) auf der Microsoft-Website. Die Werte div, div-MV, hu und hu-HU werden nicht unterstützt.

Erforderlich: Nein

DimensionName

Eine Dimension für Ihre CloudWatch Amazon-Metrik. Wenn Sie `DimensionName` angeben, müssen Sie auch `DimensionValue` angeben. Diese Parameter bieten eine andere Ansicht bei der Auflistung von Metriken. Sie können eine Dimension auch für mehrere Metriken verwenden, sodass Sie alle Metriken anzeigen können, die zu einer bestimmten Dimension gehören.

Typ: Zeichenfolge

Erforderlich: Nein

DimensionValue

Ein Dimensionswert für Ihre CloudWatch Amazon-Metrik.

Typ: Zeichenfolge

Erforderlich: Nein

Codierung

Die zu verwendende Dateikodierung (z. B: UTF-8). Verwenden Sie den Kodierungsnamen, nicht den Anzeigenamen.

Typ: Zeichenfolge

Gültige Werte: Eine Liste der unterstützten Werte finden Sie unter Encoding Class in der Microsoft Learn Bibliothek.

Erforderlich: Ja

Filter

Das Präfix des Protokollnamens. Lassen Sie diesen Parameter leer, um alle Dateien zu überwachen.

Typ: Zeichenfolge

Gültige Werte: Eine Liste der unterstützten Werte finden Sie unter „FileSystemWatcherFilter Eigenschaft“ in der MSDN-Bibliothek.

Erforderlich: Nein

Flows

Jeder Datentyp, der hochgeladen werden soll, zusammen mit dem Ziel für die Daten (CloudWatch oder CloudWatch Protokolle). Um beispielsweise einen unter definierten Leistungsindikator an das unter definierte CloudWatch Ziel `"Id": "PerformanceCounter"` zu senden`"Id": "CloudWatch"`, geben Sie "PerformanceCounter,CloudWatch" ein. Um das benutzerdefinierte Protokoll, das ETW-Protokoll und das Systemprotokoll an das unter definierte CloudWatch Protokollziel zu senden`"Id": "ETW"`, geben Sie ebenfalls „(ETW), CloudWatch Logs“ ein. Außerdem können Sie dieselbe Leistungsindikator- oder Protokolldatei an mehrere Ziele senden. Um beispielsweise das Anwendungsprotokoll an zwei verschiedene Ziele zu senden, die Sie unter `"Id": "CloudWatchLogs"` und definiert haben`"Id": "CloudWatchLogs2"`, geben Sie "ApplicationEventLog, (CloudWatchLogs, CloudWatchLogs 2)“ ein.

Typ: Zeichenfolge

Gültige Werte (Quelle): `ApplicationEventLog` | `CustomLogs` | `ETW` | `PerformanceCounter` | `SystemEventLog` | `SecurityEventLog`

Gültige Werte (Ziel): `CloudWatch` | `CloudWatchLogs` | `CloudWatch`*n* | `CloudWatchLogs`*n*

Erforderlich: Ja

FullName

Der vollständige Name der Komponente.

Typ: Zeichenfolge

Erforderlich: Ja

Id

Identifiziert die Datenquelle bzw. das Ziel. Die ID muss innerhalb der Konfigurationsdatei eindeutig sein.

Typ: Zeichenfolge

Erforderlich: Ja

InstanceName

Der Name der Leistungsindikator-Instance. Verwenden Sie kein Sternchen (*) für alle Instances, da jede Leistungszählerkomponente nur eine Metrik unterstützt. Sie können jedoch _Total verwenden.

Typ: Zeichenfolge

Erforderlich: Ja

Levels

Die Arten von Nachrichten, die an Amazon gesendet werden sollen CloudWatch.

Typ: Zeichenfolge

Zulässige Werte:

- 1 – Nur Fehlermeldungen werden hochgeladen.
- 2 – Nur Warnmeldungen werden hochgeladen.
- 4 – Nur Informationsmeldungen werden hochgeladen.

Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispiel: 3 bedeutet, dass Fehlermeldungen (1) und Warnmeldungen (2) enthalten sind. Wenn Sie den Wert 7 eingeben, werden Fehlermeldungen (1), Warnmeldungen (2) und Informationsmeldungen (4) einbezogen.

Erforderlich: Ja

> **Note**
>
> Windows-Sicherheitsprotokolle müssen für „Levels“ den Wert „7“ festlegen.

LineCount

Die Anzahl der Zeilen im Header zur Identifikation der Protokolldatei. Beispielsweise haben IIS-Protokolldateien praktisch identische Header. Sie können 3 eingeben; dann würden die ersten drei Zeilen des Headers der Protokolldatei gelesen, um diese zu identifizieren. In IIS-Protokolldateien ist die dritte Zeile Datum und Zeitstempel, die sich zwischen Protokolldateien unterscheiden.

Typ: Ganzzahl

Erforderlich: Nein

LogDirectoryPfad

Für den Pfad CustomLogs, in dem Protokolle auf Ihrer EC2-Instance gespeichert werden. *Bei IIS-Protokollen der Ordner, in dem IIS-Protokolle für eine einzelne Site gespeichert werden (z. B.* ***C:\\ inetpub\\ logs\\\LogFiles\ W3SVC*** *n).* Hinsichtlich IIS-Protokolle wird nur das Protokollformat W3C unterstützt. IIS, NCSA und benutzerdefinierte Formate werden nicht unterstützt.

Typ: Zeichenfolge

Erforderlich: Ja

LogGroup

Der Name für Ihre Protokollgruppe. Dieser Name wird auf dem Bildschirm Protokollgruppen in der Konsole angezeigt. CloudWatch

Typ: Zeichenfolge

Erforderlich: Ja

LogName

Der Name der Protokolldateien.

1. Zum Suchen des Protokollnamens klicken Sie in der Ereignisanzeige im Navigationsbereich auf Applications and Services Logs.
2. Klicken Sie in der Liste der Protokolle mit der rechten Maustaste auf das Protokoll, das Sie hochladen möchten (z. B. Microsoft > Windows > Backup > Operational), und klicken Sie dann auf Create Custom View.

3. Klicken Sie im Dialogfeld Create Custom View auf die Registerkarte XML. Der LogNamebefindet sich im Tag <Select Path=> (zum Beispiel`Microsoft-Windows-Backup`). Kopieren Sie diesen Text in den LogNameParameter.

Typ: Zeichenfolge

Zulässige Werte: `Application` | `Security` | `System` | `Microsoft-Windows-WinINet/Analytic`

Erforderlich: Ja

LogStream

Der Zielprotokollstream. Wenn Sie {instance_id}, verwenden, also den Standard, wird die Instance-ID dieser Instance als Name des Protokollstreams verwendet.

Typ: Zeichenfolge

Gültige Werte: `{instance_id}` | `{hostname}` | `{ip_address}` *`<log_stream_name>`*

Wenn Sie einen Log-Stream-Namen eingeben, der noch nicht existiert, erstellt CloudWatch Logs ihn automatisch für Sie. Sie können den Protokollstream mit einer Literalzeichenfolge, einer vordefinierten Variablen ({instance_id}, {hostname}, {ip_address}) oder einer Kombination aus allen drei Variablen definieren.

Der in diesem Parameter angegebene Log-Stream-Name wird auf dem Bildschirm Log Groups > Streams for ***`< YourLog Stream >`*** in der CloudWatch Konsole angezeigt.

Erforderlich: Ja

MetricName

Die CloudWatch Metrik, unter der Leistungsdaten enthalten sein sollen.

> **Note**
>
> Verwenden Sie keine Sonderzeichen in dem Namen. Andernfalls funktionieren die Metrik und die zugehörigen Alarme möglicherweise nicht.

Typ: Zeichenfolge

Erforderlich: Ja

NameSpace

Der Metrik-Namespace, in dem die Leistungsindikatordaten geschrieben werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

PollInterval

Die Anzahl der Sekunden, die vergehen muss, bevor neue Leistungsindikator- und Protokolldaten hochgeladen werden.

Typ: Ganzzahl

Gültige Werte: Legen Sie für diesen Wert 5 oder mehr Sekunden fest. Fünfzehn Sekunden (00:00:15) sind empfohlen.

Erforderlich: Ja

Region

Der AWS-Region Ort, an den Sie Protokolldaten senden möchten. Obwohl Sie Leistungszähler an eine andere Region als die, an die Sie Ihre Protokolldaten senden, senden können, empfehlen wir, diesen Parameter auf dieselbe Region zu setzen, in der Ihre Instance ausgeführt wird.

Typ: Zeichenfolge

Gültige Werte: Regions-IDs der sowohl von Systems Manager als auch von CloudWatch Logs AWS-Regionen unterstützten Bereich`eus-east-2`, wie`eu-west-1`, und`ap-southeast-1`. Eine Liste der von den einzelnen Services AWS-Regionen unterstützten Services finden Sie unter Amazon CloudWatch Logs Service Endpoints und Systems Manager Service Endpoints in der. Allgemeine Amazon Web Services-Referenz

Erforderlich: Ja

SecretKey

Ihr geheimer -Zugriffsschlüssel Diese Eigenschaft ist erforderlich, wenn Sie die Instance mithilfe einer IAM-Rolle gestartet haben.

Typ: Zeichenfolge

Erforderlich: Nein

startType

Schalten Sie die Instance ein oder aus CloudWatch .

Typ: Zeichenfolge

Zulässige Werte: `Enabled` | `Disabled`

Erforderlich: Ja

TimestampFormat

Das Zeitstempelformat, das Sie verwenden möchten. Eine Liste der unterstützten Werte finden Sie unter Custom Date and Time Format Strings in der MSDN-Bibliothek.

Typ: Zeichenfolge

Erforderlich: Ja

TimeZoneFreundlich

Stellt Zeitzoneninformationen bereit, wenn der Zeitstempel Ihres Protokolls keine Zeitzoneninformationen enthält. Wenn dieser Parameter leer gelassen wird und Ihr Zeitstempel keine Zeitzoneninformationen enthält, verwendet CloudWatch Logs standardmäßig die lokale Zeitzone. Dieser Parameter wird ignoriert, wenn der Zeitstempel bereits Zeitzoneninformationen enthält.

Typ: Zeichenfolge

Zulässige Werte: `Local` | `UTC`

Erforderlich: Nein

Einheit

Die korrekte Maßeinheit für die Metrik.

Typ: Zeichenfolge

Gültige Werte: Sekunden | Mikrosekunden | Millisekunden | Bytes | KB | MB | GB | TB | Bits | Kilobits | Megabits | Gigabits | Terabits | Prozent | Anzahl | Byte/Sekunde | KB/Sekunde | MB/Sekunde | GB/Sekunde | TB/Sekunde | Bits/Sekunde | Kilobit/Sekunde | Megabit/Sekunde | Gigabit/Sekunde | Terabit/Sekunde | Anzahl/Sekunde | Keine.

Erforderlich: Ja

`aws:configureDocker`

(Schemaversion 2.0 oder höher) Konfigurieren Sie eine Instance für die Arbeit mit Container und Docker. Dieses Plug-In wird unter Linux- und Windows Server-Betriebssystemen unterstützt.

Syntax

Schema 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 }}"
      }
    }
  ]
}
```

Eingaben

action

Der Typ der Aktion, die durchgeführt werden soll.

Type: Zähler

Zulässige Werte: `Install` | `Uninstall`

Erforderlich: Ja

`aws:configurePackage`

(Schemaversion 2.0 oder höher) Installieren oder deinstallieren Sie ein AWS Systems Manager Distributor Paket. Sie können die neueste Version, die Standardversion oder eine Version des angegebenen Pakets installieren. Pakete, die von bereitgestellt AWS werden, werden ebenfalls unterstützt. Dieses Plug-in läuft unter Windows Server- und Linux-Betriebssystemen, wobei jedoch in Linux-Betriebssystemen nicht alle verfügbaren Pakete unterstützt werden.

Zu den verfügbaren AWS Paketen für Windows Server gehören: `AWSPVDriverAWSNVMe`,`AwsEnaNetworkDriver`,`AwsVssComponents`,`AmazonCloudWatchAgent`,`Code` und `AWSSupport-EC2Rescue.`

Zu den verfügbaren AWS Paketen für Linux-Betriebssysteme gehören die folgenden: `AmazonCloudWatchAgentCodeDeployAgent`, und`AWSSupport-EC2Rescue`.

Syntax

Schema 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\"}"
            }
         }
      ]
}
```

Eingaben

Name

Der Name des zu installierenden oder zu deinstallierenden AWS Pakets. In verfügbaren Paketen ist Folgendes enthalten: `AWSPVDriver`, `AwsEnaNetworkDriver`, `AwsVssComponents` und `AmazonCloudWatchAgent`.

Typ: Zeichenfolge

Erforderlich: Ja

action

Installieren oder deinstallieren Sie ein Paket.

Type: Zähler

Zulässige Werte: `Install` | `Uninstall`

Erforderlich: Ja

installationType

Der Typ der auszuführenden Installation. Wenn Sie `Uninstall and reinstall` angeben, wird das Paket vollständig deinstalliert und anschließend neu installiert. Die Anwendung ist bis zum Abschluss der Neuinstallation nicht verfügbar. Wenn Sie `In-place update` angeben, werden der vorhandenen Installation nur neue oder geänderte Dateien hinzugefügt, entsprechend den Anweisungen, die Sie in einem Update-Skript bereitstellen. Die Anwendung ist während des Aktualisierungsprozesses weiterhin verfügbar. Die `In-place update` Option wird für Pakete mit dem Namen AWS-published nicht unterstützt. `Uninstall and reinstall`ist der Standardwert.

Type: Zähler

Zulässige Werte: `Uninstall and reinstall` | `In-place update`

Erforderlich: Nein

additionalArguments

Eine JSON-Zeichenkette mit den zusätzlichen Parametern, die Sie Ihren Installations-, Deinstallations- oder Update-Skripten bereitstellen müssen. Jedem Parameter muss das Präfix `SSM_` angefügt werden. Sie können in Ihren zusätzlichen Argumenten auf einen Parameter Store-Parameter verweisen, indem Sie die Konvention `{{ssm:`*`parameter-name`*`}}` verwenden. Um den zusätzlichen Parameter in Ihrem Installations-, Deinstallations- oder Updateskript zu verwenden, müssen Sie den Parameter mithilfe der für das Betriebssystem geeigneten Syntax als Umgebungsvariable referenzieren. In verweisen Sie PowerShell beispielsweise auf das `SSM_arg` Argument als`$Env:SSM_arg`. Es gibt keine Begrenzung für die Anzahl der von Ihnen definierten Argumente, aber die Eingabe von zusätzlichen Argumenten hat eine Begrenzung von 4096 Zeichen. Dieser Grenzwert umfasst alle von Ihnen definierten Schlüssel und Werte.

Geben Sie ein: StringMap

Erforderlich: Nein

version

Installieren oder deinstallieren Sie eine bestimmte Version des Pakets. Wenn Sie ein Installation vornehmen, installiert das System standardmäßig die neueste veröffentlichte Version. Wenn Sie eine Deinstallation vornehmen, deinstalliert das System standardmäßig die derzeit installierte Version. Wenn keine installierte Version gefunden wird, wird die neueste veröffentlichte Version heruntergeladen und die Deinstallationsaktion ausgeführt.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:domainJoin`

Verbinden Sie eine EC2-Instance mit einer Domain. Dieses Plug-In läuft unter Linux- und Windows Server-Betriebssystemen. Dieses Plugin ändert den Hostnamen für Linux-Instances in das Format EC2AMAZ-*XXXXXXX*. Weitere Informationen zum Beitreten von EC2-Instances finden Sie unter Join an EC2 Instance to Your AWS Managed Microsoft AD Directory im AWS Directory Service Administrationshandbuch.

Syntax

Schema 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
mainSteps:
- action: aws:domainJoin
  name: domainJoin
  inputs:
    directoryId: "{{ directoryId }}"
    directoryName: "{{ directoryName }}"
    directoryOU: "{{ directoryOU }}"
    dnsIpAddresses: "{{ dnsIpAddresses }}"
```

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"
    },
  },
  "mainSteps": [
    {
      "action": "aws:domainJoin",
      "name": "domainJoin",
      "inputs": {
        "directoryId": "{{ directoryId }}",
```

```
        "directoryName": "{{ directoryName }}",
        "directoryOU":"{{ directoryOU }}",
        "dnsIpAddresses":"{{ dnsIpAddresses }}"
      }
    }
  ]
}
```

Schema 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 }}"
         }
      }
   }
}
```

Eigenschaften

directoryId

Die ID des Verzeichnisses.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel: "directoryId": "d-1234567890"

directoryName

Der Name der Domain.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel: "directoryName": "example.com"

directoryOU

Die Organisationseinheit (OU).

Typ: Zeichenfolge

Erforderlich: Nein

Beispiel: "directoryOU": "OU=test,DC=example,DC=com"

dns IpAddresses

Die IP-Adressen des DNS-Servers.

Typ: StringList

Erforderlich: Ja

Beispiel: „dns IpAddresses „: [" 198.51.100.1", "198.51.100.2"]

Beispiele

Beispiele finden Sie unter Verbinden einer Amazon-EC2-Instance mit Ihrem AWS Managed Microsoft AD im AWS Directory Service -Administratorhandbuch.

`aws:downloadContent`

(Schemaversion 2.0 oder höher) Laden Sie SSM-Dokumente und -Skripts von entfernten Standorten herunter. GitHub EnterpriseRepositorys werden nicht unterstützt. Dieses Plug-In wird unter Linux- und Windows Server-Betriebssystemen unterstützt.

Syntax

Schema 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 }}"
      }
    }
  ]
}
```

Eingaben

sourceType

Die Downloadquelle. Systems Manager unterstützt derzeit die folgenden Quellarten für das Herunterladen von Skripts und SSM documents:-Dokumenten: `GitHub`, `Git`, `HTTP`, `S3` und `SSMDocument`.

Typ: Zeichenfolge

Erforderlich: Ja

sourceInfo

Die erforderlichen Informationen zum Abrufen der Inhalte aus der erforderlichen Quelle.

Typ: StringMap

Erforderlich: Ja

Für sourceType **`GitHub,`** geben Sie Folgendes an:

- owner: Die Eigentümer des Repositorys.
- repository: Der Name des Repositorys.
- path: Der Pfad zu der Datei oder dem Verzeichnis, die bzw. das Sie herunterladen möchten.
- getOptions: Zusätzliche Optionen zum Abrufen von Inhalten aus einem anderen Branch als dem Master-Branch oder aus einem bestimmten Commit im Repository. getOptions kann weggelassen werden, wenn Sie den letzten Commit in der Master-Branch verwenden. Wenn Ihr Repository nach dem 1. Oktober 2020 erstellt wurde, wird der Standardzweig möglicherweise „main“ statt „master“ genannt. In diesem Fall müssen Sie Werte für den getOptions-Parameter angeben.

  Dieser Parameter verwendet das folgende Format:

- branch:refs/heads/*branch_name*

  Der Standardwert ist `master`.

  Verwenden Sie das folgende Format, um einen nicht standardmäßigen Zweig anzugeben:

  branch:refs/heads/*branch_name*

- commitID:*commitID*

  Der Standardwert ist `head`.

  Um die Version Ihres SSM-Dokuments in einem anderen als dem letzten Commit zu verwenden, geben Sie die vollständige Commit-ID an. Beispielsweise:

  ```
  "getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
  ```

- tokenInfo: Der Systems Manager Manager-Parameter (ein SecureString Parameter), in dem Sie Ihre GitHub Zugriffstoken-Informationen speichern, im Format. `{{ssm-secure:`*secure-string-token-name*`}}`

  > **Note**
  >
  > Dieses `tokenInfo` Feld ist das einzige SSM-Dokument-Plugin-Feld, das einen Parameter unterstützt. SecureString SecureString Parameter werden weder für andere Felder noch für andere SSM-Dokument-Plugins unterstützt.

```
{
    "owner":"TestUser",
    "repository":"GitHubTest",
    "path":"scripts/python/test-script",
    "getOptions":"branch:master",
    "tokenInfo":"{{ssm-secure:secure-string-token}}"
}
```

Für sourceType **`Git`** müssen Sie Folgendes angeben:

- Repository

  Die URL des Git-Repositorys zu der Datei oder dem Verzeichnis, die bzw. das Sie herunterladen möchten.

Typ: Zeichenfolge

Sie können zusätzlich einen der folgenden optionalen Parameter angeben:

- getOptions

  Zusätzliche Optionen zum Abrufen von Inhalten aus einem anderen Branch als dem Master-Branch oder aus einem bestimmten Commit im Repository. getOptions kann weggelassen werden, wenn Sie den letzten Commit in der Master-Branch verwenden.

  Typ: Zeichenfolge

  Dieser Parameter verwendet das folgende Format:

  - branch:refs/heads/*branch_name*

    Der Standardwert ist `master`.

    `"branch"` ist nur erforderlich, wenn Ihr SSM-Dokument in einer anderen Verzweigung als `master` gespeichert ist. Zum Beispiel:

    ```
    "getOptions": "branch:refs/head/main"
    ```

  - commitID:*commitID*

    Der Standardwert ist `head`.

    Um die Version Ihres SSM-Dokuments in einem anderen als dem letzten Commit zu verwenden, geben Sie die vollständige Commit-ID an. Zum Beispiel:

    ```
    "getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
    ```

- privateSSHKey

  Der SSH-Schlüssel, der beim Herstellen einer Verbindung zur `repository` verwendet werden soll. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres SSH-Schlüssels zu verweisen: `{{ssm-secure:`*your-secure-string-parameter*`}}`.

  Typ: Zeichenfolge

- Überprüfung überspringen HostKey

Bestimmt den Wert der StrictHostKeyChecking Option, wenn eine Verbindung zu dem von `repository` Ihnen angegebenen hergestellt wird. Der Standardwert ist `false`.

Typ: Boolesch

- username

  Der Benutzername, der bei der Verbindung mit der `repository` verwendet werden soll, die Sie mit HTTP angeben. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres Benutzernamens zu verweisen: `{{ssm-secure:`*`your-secure-string-parameter`*`}}`.

  Typ: Zeichenfolge

- password

  Das Passwort, das bei der Verbindung mit der `repository` verwendet werden soll, die Sie mit HTTP angeben. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres Passworts zu verweisen: `{{ssm-secure:`*`your-secure-string-parameter`*`}}`.

  Typ: Zeichenfolge

Für sourceType **`HTTP`** müssen Sie Folgendes angeben:

- URL

  Die URL zu der Datei oder dem Verzeichnis, die bzw. das Sie herunterladen möchten.

  Typ: Zeichenfolge

Sie können zusätzlich einen der folgenden optionalen Parameter angeben:

- zulassen InsecureDownload

  Bestimmt, ob ein Download über eine Verbindung durchgeführt werden kann, die nicht mit Secure Socket Layer (SSL) oder Transport Layer Security (TLS) verschlüsselt ist. Der Standardwert ist `false`. Wir raten davon ab, Downloads ohne Verschlüsselung durchzuführen. Wenn Sie sich dafür entscheiden, übernehmen Sie alle damit verbundenen Risiken. Sicherheit ist eine gemeinsame Verantwortung zwischen Ihnen AWS und Ihnen. Dies wird als Modell der geteilten Verantwortung beschrieben. Weitere Informationen hierzu finden Sie in Modell der geteilten Verantwortung.

Typ: Boolesch

- authMethod

  Bestimmt, ob ein Benutzername und ein Passwort für die Authentifizierung verwendet werden, wenn eine Verbindung mit der `url` hergestellt wird, die Sie angeben. Wenn Sie `Basic` oder `Digest` angeben, müssen Sie Werte für die `username`- und `password`-Parameter bereitstellen. Um die `Digest`-Methode zu verwenden, muss SSM Agent-Version 3.0.1181.0 oder höher muss auf Ihrer Instance installiert sein. Die `Digest`-Methode unterstützt MD5- und SHA256-Verschlüsselung.

  Typ: Zeichenfolge

  Zulässige Werte: `None` | `Basic` | `Digest`

- username

  Der Benutzername, der bei der Verbindung mit der `url` verwendet werden soll, die Sie mit `Basic`-Authentifizierung angeben. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres Benutzernamens zu verweisen: `{{ssm-secure:`*`your-secure-string-parameter`*`}}`.

  Typ: Zeichenfolge

- password

  Das Passwort, das bei der Verbindung mit der `url` verwendet werden soll, die Sie mit `Basic`-Authentifizierung angeben. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres Passworts zu verweisen: `{{ssm-secure:`*`your-secure-string-parameter`*`}}`.

  Typ: Zeichenfolge

Für sourceType **`S3`** geben Sie Folgendes an:

- Die URL zu der Datei oder dem Verzeichnis, die bzw. das Sie von Amazon S3 herunterladen möchten.

```
{
    "path": "https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/powershell/
helloPowershell.ps1"
}
```

Geben Sie für sourceType **`SSMDocument`** eine der folgenden Optionen an:

- name: Der Name und die Version des Dokuments in folgendem Format: `name:version`. Version ist optional.

```
{
    "name": "Example-RunPowerShellScript:3"
}
```

- name: Der ARN für das Dokument im folgenden Format: `arn:aws:ssm:region:account_id:document/document_name`

```
{
   "name":"arn:aws:ssm:us-east-2:3344556677:document/MySharedDoc"
}
```

destinationPath

Ein optionaler lokaler Pfad auf der Instance, in den die Datei heruntergeladen werden soll. Wenn Sie keinen Pfad angeben, wird der Inhalt in einen Pfad relativ zu Ihrer Befehls-ID heruntergeladen.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:psModule`

Installieren Sie PowerShell Module auf einer Amazon EC2 EC2-Instance. Dieses Plug-In läuft nur unter Windows Server-Betriebssystemen.

Syntax

Schema 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 }}"
      }
    }
  ]
}
```

Schema 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 }}"
            }
         ]
      }
   }
}
```

Eigenschaften

runCommand

Der PowerShell Befehl, der nach der Installation des Moduls ausgeführt werden soll.

Typ: StringList

Erforderlich: Nein

Quelle

Die URL bzw. der lokale Pfad auf der Instance zur `.zip`-Datei der Anwendung.

Typ: Zeichenfolge

Erforderlich: Ja

sourceHash

Der SHA256-Hashwert der `.zip`-Datei.

Typ: Zeichenfolge

Erforderlich: Nein

timeoutSeconds

Die Zeit in Sekunden, die ein Befehl in Anspruch nehmen darf, bevor er als fehlgeschlagen betrachtet wird.

Typ: Zeichenfolge

Erforderlich: Nein

workingDirectory

Der Pfad zum Arbeitsverzeichnis auf der Instance.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:refreshAssociation`

(Schemaversion 2.0 oder höher) Aktualisieren (Erzwingen) Sie bei Bedarf eine Zuweisung. Diese Aktion ändert den Systemstatus basierend auf was in der ausgewählten Verknüpfungen bzw. in allen zielgebundenen Verknüpfungen definiert ist. Dieses Plug-In läuft unter Linux- und Microsoft Windows Server-Betriebssystemen.

Syntax

Schema 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 }}"
        ]
      }
    }
  ]
}
```

Eingaben

associationIds

Liste der Verknüpfungs-IDs. Wenn das Feld leer ist, werden alle Verknüpfungen mit dem angegebenen Ziel angewendet.

Typ: StringList

Erforderlich: Nein

`aws:runDockerAction`

(Schemaversion 2.0 oder höher) Führen Sie Docker-Aktionen auf Containern aus. Dieses Plug-In läuft unter Linux- und Microsoft Windows Server-Betriebssystemen.

Syntax

Schema 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 }}"
```

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 }}"
         }
      }
   ]
}
```

Eingaben

action

Der Typ der Aktion, die durchgeführt werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

Container

Die Container-ID des Dockers.

Typ: Zeichenfolge

Erforderlich: Nein

Abbild

Der Name des Docker-Image.

Typ: Zeichenfolge

Erforderlich: Nein

cmd

Der Container-Befehl.

Typ: Zeichenfolge

Erforderlich: Nein

memory

Die Grenze des Container-Speichers.

Typ: Zeichenfolge

Erforderlich: Nein

cpuShares

Die Container-CPU-Anteile (relative Gewichtung).

Typ: Zeichenfolge

Erforderlich: Nein

Volume

Die Container-Volume-Mounts.

Typ: StringList

Erforderlich: Nein

env

Die Container-Umgebungsvariablen.

Typ: Zeichenfolge

Erforderlich: Nein

user

Der Container-Benutzername.

Typ: Zeichenfolge

Erforderlich: Nein

publish

Die veröffentlichten Container-Ports.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:runDocument`

(Schema-Version 2.0 oder höher) Führt SSM-Dokumente aus, die in Systems Manager oder einem lokal freigegebenen Verzeichnis gespeichert sind. Sie können dieses Plug-In mit dem Plug-In `aws:downloadContent` verwenden, um ein SSM-Dokument von einem Remote-Standort in ein

lokal freigegebenes Verzeichnis herunterzuladen, und es dann ausführen. Dieses Plug-In wird unter Linux- und Windows Server-Betriebssystemen unterstützt. Dieses Plug-In unterstützt nicht das Ausführen des `AWS-UpdateSSMAgent`-Dokuments oder eines anderen Dokuments, das den `aws:updateSsmAgent`-Plug-In verwendet.

Syntax

Schema 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 }}"
      }
    }
  ]
}
```

Eingaben

documentType

Der auszuführende Dokumenttyp. Sie können lokale Dokumente (`LocalPath`) oder in Systems Manager gespeicherte Dokumente (`SSMDocument`) ausführen.

Typ: Zeichenfolge

Erforderlich: Ja

documentPath

Der Pfad zu dem Dokument. Wenn `documentType` `LocalPath` ist, geben Sie den Pfad des Dokuments im lokal freigegebenen Verzeichnis an. Wenn `documentType` `SSMDocument` ist, geben Sie den Namen des Dokuments an.

Typ: Zeichenfolge

Erforderlich: Nein

documentParameters

Parameter für das Dokument.

Typ: StringMap

Erforderlich: Nein

`aws:runPowerShellScript`

Führen Sie PowerShell Skripts aus oder geben Sie den Pfad zu einem auszuführenden Skript an. Dieses Plug-In läuft unter Microsoft Windows Server- und Linux-Betriebssystemen.

Syntax

Schema 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 }}"
        ]
      }
    }
  ]
}
```

Schema 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 }}"
            }
         ]
      }
   }
}
```

Eigenschaften

runCommand

Geben Sie die auszuführenden Befehle oder den Pfad zu einem vorhandenen Skript auf der Instance an.

Typ: StringList

Erforderlich: Ja

timeoutSeconds

Die Zeit in Sekunden, die ein Befehl in Anspruch nehmen darf, bevor er als fehlgeschlagen betrachtet wird. Wenn der Wert für den Timeout erreicht ist, hält Systems Manager die Ausführung des Befehls an.

Typ: Zeichenfolge

Erforderlich: Nein

workingDirectory

Der Pfad zum Arbeitsverzeichnis auf der Instance.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:runShellScript`

Führen Sie Linux-Shell-Skripts aus oder geben Sie den Pfad zu einem auszuführenden Skript an. Dieses Plug-In läuft nur unter Linux-Betriebssystemen.

Syntax

Schema 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 }}"
        ]
      }
    }
  ]
}
```

Schema 1.2

YAML

```
---
runtimeConfig:
  aws:runShellScript:
    properties:
    - runCommand: "{{ commands }}"
      workingDirectory: "{{ workingDirectory }}"
      timeoutSeconds: "{{ executionTimeout }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:runShellScript":{
         "properties":[
            {
               "runCommand":"{{ commands }}",
               "workingDirectory":"{{ workingDirectory }}",
               "timeoutSeconds":"{{ executionTimeout }}"
            }
         ]
      }
   }
}
```

Eigenschaften

runCommand

Geben Sie die auszuführenden Befehle oder den Pfad zu einem vorhandenen Skript auf der Instance an.

Typ: StringList

Erforderlich: Ja

timeoutSeconds

Die Zeit in Sekunden, die ein Befehl in Anspruch nehmen darf, bevor er als fehlgeschlagen betrachtet wird. Wenn der Wert für den Timeout erreicht ist, hält Systems Manager die Ausführung des Befehls an.

Typ: Zeichenfolge

Erforderlich: Nein

workingDirectory

Der Pfad zum Arbeitsverzeichnis auf der Instance.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:softwareInventory`

(Schema-Version 2.0 oder höher) Erfassen von Metadaten zu Anwendungen, Dateien und Konfigurationen auf Ihren verwalteten Instances. Dieses Plug-In läuft unter Linux- und Microsoft Windows Server-Betriebssystemen. Wenn Sie die Inventarerfassung konfigurieren, erstellen Sie zunächst eine AWS Systems Manager State Manager Zuordnung. Systems Manager erfasst die Bestandsdaten, wenn der Zuordnungsstatus ausgeführt wird. Wenn Sie den Zuordnungsstatus nicht zuerst erstellen und versuchen, das `aws:softwareInventory`-Plug-In aufzurufen, gibt das System den folgenden Fehler aus:

```
The aws:softwareInventory plugin can only be invoked via ssm-associate.
```

Pro Instance kann nur jeweils ein Bestandszuordnungsstatus konfiguriert werden. Wenn Sie eine Instance mit zwei oder mehr Zuordnungen konfigurieren, wird der Bestandszuordnungsstatus nicht ausgeführt und es werden keine Bestandsdaten erfasst. Weitere Informationen über das Erfassen des Bestands finden Sie unter AWS Systems Manager-Bestand.

Syntax

Schema 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 }}"
         }
      }
   ]
}
```

Eingaben

applications

(Optional) Erfassen von Metadaten für installierte Anwendungen.

Typ: Zeichenfolge

Erforderlich: Nein

awsComponents

(Optional) Sammeln Sie Metadaten für AWS Komponenten wie amazon-ssm-agent.

Typ: Zeichenfolge

Erforderlich: Nein

files

(Optional, erfordert SSM Agent-Version 2.2.64.0 oder höher) Erfassen von Metadaten für Dateien, einschließlich Dateinamen, der Erstellungszeit der Dateien, der letzten Änderungs- und Zugriffszeit der Dateien oder Dateigrößen usw. Weitere Informationen zum Erfassen eines Dateibestands finden Sie unter Arbeiten mit Datei- und Windows-Registrierungsbestand.

Typ: Zeichenfolge

Erforderlich: Nein

networkConfig

(Optional) Erfassen von Metadaten für Netzwerkkonfigurationen.

Typ: Zeichenfolge

Erforderlich: Nein

windowsUpdates

(Optional) Erfassen von Metadaten für alle Windows-Updates.

Typ: Zeichenfolge

Erforderlich: Nein

Instanz DetailedInformation

(Optional) Erfassen weiterer Instance-Informationen neben den Informationen des Standardbestands-Plug-ins (`aws:instanceInformation`), einschließlich CPU-Modell, Geschwindigkeit und Anzahl der Kerne usw.

Typ: Zeichenfolge

Erforderlich: Nein

service

(Optional, nur Windows-BS, erfordert SSM Agent-Version 2.2.64.0 oder höher) Erfassen von Daten für Servicekonfigurationen.

Typ: Zeichenfolge

Erforderlich: Nein

windowsRegistry

(Optional, nur Windows-BS, erfordert SSM Agent-Version 2.2.64.0 oder höher) Erfassen von Windows Registry-Schlüsseln und -Werten. Sie können einen Schlüssel-Pfad auswählen und alle Schlüssel und Werte rekursiv erfassen. Sie können auch einen bestimmten Registrierungsschlüssel und seinen Wert für einen bestimmten Pfad erfassen. Inventory erfasst den Schlüsselpfad, den Namen, Typ und Wert. Weitere Informationen zur Erfassung von Windows Registry-Bestand finden Sie unter Arbeiten mit Datei- und Windows-Registrierungsbestand.

Typ: Zeichenfolge

Erforderlich: Nein

windowsRoles

(Optional, nur Windows-BS, erfordert SSM Agent-Version 2.2.64.0 oder höher) Erfassen von Metadaten für Microsoft Windows-Rollenkonfigurationen.

Typ: Zeichenfolge

Erforderlich: Nein

customInventory

(Optional) Erfassen von benutzerdefinierten Bestandsdaten. Weitere Informationen zum benutzerdefinierten Bestand finden Sie unter Arbeiten mit benutzerdefiniertem Bestand

Typ: Zeichenfolge

Erforderlich: Nein

`aws:updateAgent`

Aktualisieren Sie den EC2Config-Service auf die neueste Version oder geben Sie eine ältere Version an. Dieses Plug-In läuft nur unter Microsoft Windows Server-Betriebssystemen. Weitere Informationen zum EC2Config-Service finden Sie unter Configuring a Windows Instance using the EC2Config Service (legacy) im Amazon EC2 EC2-Benutzerhandbuch.

Syntax

Schema 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"
      }
    }
```

```
  ]
}
```

Schema 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 }}"
         }
      }
   }
}
```

Eigenschaften

agentName

EC2Config. Dies ist der Name des Agenten, der den EC2Config-Service ausführt.

Typ: Zeichenfolge

Erforderlich: Ja

allowDowngrade

Erlauben Sie, dass der EC2Config-Service auf eine ältere Version zurückgesetzt wird. Wenn hierfür „false“ festgelegt ist, kann der Service nur auf neuere Versionen aktualisiert werden (Standard). Wenn hierfür „true“ festgelegt wurde, geben Sie die ältere Version an.

Typ: Boolesch

Erforderlich: Nein

Quelle

Der Ort, an den Systems Manager die zu installierende Version von EC2Config kopiert. Sie können diesen Speicherort nicht ändern.

Typ: Zeichenfolge

Erforderlich: Ja

targetVersion

Eine bestimmte zu installierende Version des EC2Config-Service. Ist hierfür nichts angegeben, wird der Dienst auf die neueste Version aktualisiert.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:updateSsmAgent`

Aktualisieren Sie SSM Agent auf die neueste Version oder geben Sie eine ältere Version an. Dieses Plug-In läuft unter Linux- und Windows Server-Betriebssystemen. Weitere Informationen finden Sie unter Arbeiten mit SSM Agent.

Syntax

Schema 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 }}"
      }
    }
  ]
}
```

Schema 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 }}"
            }
         ]
      }
   }
}
```

Eigenschaften

agentName

amazon-ssm-agent. Dies ist der Name des Systems Manager-Agenten, der Anforderungen verarbeitet und auf der Instance Befehle ausführt.

Typ: Zeichenfolge

Erforderlich: Ja

allowDowngrade

Erlauben Sie, dass SSM Agent auf eine ältere Version zurückgesetzt wird. Wenn hierfür „false“ festgelegt ist, kann der Agent nur auf neuere Versionen aktualisiert werden (Standard). Wenn hierfür „true“ festgelegt wurde, geben Sie die ältere Version an.

Typ: Boolesch

Erforderlich: Ja

Quelle

Der Ort, an den Systems Manager die zu installierende Version von SSM Agent kopiert. Sie können diesen Speicherort nicht ändern.

Typ: Zeichenfolge

Erforderlich: Ja

targetVersion

Eine bestimmte zu installierende Version von SSM Agent. Ist hierfür nichts angegeben, wird der Agent auf die neueste Version aktualisiert.

Typ: Zeichenfolge

Erforderlich: Nein

Erstellen von SSM-Dokumentinhalten

Wenn die AWS Systems Manager öffentlichen Dokumente nicht alle Aktionen ausführen, die Sie für Ihre AWS Ressourcen ausführen möchten, können Sie Ihre eigenen SSM-Dokumente erstellen. Sie können SSM-Dokumente auch über die Konsole klonen. Beim Klonen von Dokumenten werden Inhalte aus einem vorhandenen Dokument in ein neues Dokument kopiert, das Sie ändern

können. Beim Erstellen oder Klonen eines Dokuments darf der Inhalt des Dokuments 64 KB nicht überschreiten. Dieses Kontingent beinhaltet auch den zur Laufzeit für Eingabeparameter angegebenen Inhalt. Wenn Sie ein neues `Command`- oder `Policy`-Dokument erstellen, wird empfohlen, Schemaversion 2.2 oder höher zu verwenden, damit Sie die neuesten Features wie Dokumentbearbeitung, automatisches Versioning, Sequenzierung usw. nutzen können.

Schreiben von SSM-Dokumentinhalt

Um eigene SSM-Dokumentinhalte zu erstellen, müssen Sie die verschiedenen Schemas, Features, Plugins und die Syntax für SSM-Dokumente verstehen. Wir empfehlen Ihnen, sich mit den folgenden Ressourcen vertraut zu machen.

- Schreiben Sie Ihre eigenen Dokumente AWS Systems Manager
- Datenelemente und Parameter
- Schemata, Features und Beispiele
- Referenz für Befehlsdokument-Plug-ins
- Systems Manager Automation Aktionen-Referenz
- Systemvariablen für Automation
- Weitere Runbook-Beispiele
- Arbeiten mit Systems Manager Automation-Runbooks mithilfe des AWS Toolkit for Visual Studio Code
- Verwenden von Document Builder zur Erstellung von Runbooks
- Verwenden von Skripten in Runbooks

AWS Vordefinierte SSM-Dokumente können einige der von Ihnen benötigten Aktionen ausführen. Sie können diese Dokumente je nach Dokumenttyp mithilfe der Plugins `aws:runDocument`, `aws:runCommand` oder `aws:executeAutomation` in Ihrem benutzerdefinierten SSM-Dokument aufrufen. Sie können Teile dieser Dokumente auch in ein benutzerdefiniertes SSM-Dokument kopieren und den Inhalt entsprechend Ihren Anforderungen bearbeiten.

> **Tip**
>
> Beim Erstellen von SSM-Dokumentinhalten können Sie den Inhalt ändern und das SSM-Dokument während des Tests mehrmals aktualisieren. Mit den folgenden Befehlen wird das SSM-Dokument mit dem neuesten Inhalt aktualisiert und die Standardversion des Dokuments wird auf die neueste Dokumentversion aktualisiert.

> **Note**
>
> Die Linux- und Windows-Befehle nutzen das `jq`-Befehlszeilen-Tool, um die JSON-Antwortdaten zu filtern.

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
```

Klonen eines SSM-Dokuments

Sie können AWS Systems Manager Dokumente mit der Systems Manager Documents Console klonen, um SSM-Dokumente zu erstellen. Durch das Klonen von SSM-Dokumenten werden Inhalte aus einem vorhandenen Dokument in ein neues Dokument kopiert, das Sie ändern können. Sie können kein Dokument klonen, das größer als 64 KB ist.

Klonen eines SSM-Dokuments

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Geben Sie in das Suchfeld den Namen des Dokuments ein, das Sie klonen möchten.
4. Wählen Sie den Namen des Dokuments aus, das Sie klonen möchten. Wählen Sie anschließend die Option Clone document (Dokument klonen) im Dropdownmenü Aktionen.
5. Ändern Sie das Dokument nach Belieben und wählen Sie dann Create document (Dokument erstellen), um das Dokument zu speichern.

Nachdem Sie den SSM-Dokumentinhalt geschrieben haben, können Sie mithilfe eines der folgenden Methoden ein SSM-Dokument erstellen.

Erstellen eines SSM-Dokuments

- Erstellen von zusammengesetzten Dokumenten

Erstellen von zusammengesetzten Dokumenten

Ein zusammengesetztes Dokument AWS Systems Manager (SSM) ist ein benutzerdefiniertes Dokument, das eine Reihe von Aktionen ausführt, indem es ein oder mehrere sekundäre SSM-Dokumente ausführt. Zusammengesetzte Dokumente fördern Infrastruktur als Code, indem sie

Ihnen ermöglichen, einen Standardsatz an SSM-Dokumenten für allgemeine Aufgaben wie das Bootstrapping von Software oder den Domain-Betritt von Instances zu erstellen. Sie können diese Dokumente dann gemeinsam nutzen, um die Wartung von SSM-Dokumenten AWS-Region zu reduzieren und die Konsistenz AWS-Konten zu gewährleisten.

Sie können beispielsweise ein zusammengesetztes Dokument erstellen, das die folgenden Aktionen ausführt:

1. Installiert alle Patches in der Zulassungsliste.
2. Installieren von Antivirensoftware
3. Lädt Skripte von ihnen herunter GitHub und führt sie aus.

In diesem Beispiel umfasst das benutzerdefinierte SSM-Dokument die folgenden Plug-Ins für die Ausführung dieser Aktionen:

1. Das `aws:runDocument`-Plugin zum Ausführen des `AWS-RunPatchBaseline`-Dokuments, das alle aufgeführten Patches installiert.
2. Das Plug-In `aws:runDocument` zum Ausführen des Dokuments `AWS-InstallApplication`, das die Antivirensoftware installiert
3. Das `aws:downloadContent` Plugin zum Herunterladen GitHub und Ausführen von Skripten.

Zusammengesetzte und sekundäre Dokumente können in Systems Manager GitHub (öffentliche und private Repositorys) oder Amazon S3 gespeichert werden. Zusammengesetzte und sekundäre Dokumente lassen sich im JSON- oder YAML-Format erstellen.

> **Note**
>
> Zusammengesetzte Dokumente können maximal drei Dokumente tief ausgeführt werden. Dies bedeutet, dass ein zusammengesetztes Dokument ein untergeordnetes Dokument aufrufen kann, das wiederum ein letztes Dokument aufrufen kann.

Zum Erstellen eines zusammengesetzten Dokuments fügen Sie das Plug-In `aws:runDocument` einem benutzerdefinierten SSM-Dokument hinzu und geben die erforderlichen Eingaben an. Folgendes ist ein Beispiel eines zusammengesetzten Dokuments, das die folgenden Aktionen ausführt:

1. Führt das aws:downloadContent Plugin aus, um ein SSM-Dokument aus einem GitHub öffentlichen Repository in ein lokales Verzeichnis namens Bootstrap herunterzuladen. Das SSM-Dokument heißt StateManagerBootstrap .yml (ein YAML-Dokument).
2. Führt das aws:runDocument Plugin aus, um das .yml-Dokument auszuführen. StateManagerBootstrap Es wurden keine Parameter angegeben.
3. Führt das Plugin aws:runDocument aus, um das AWS-ConfigureDocker pre-defined SSM-Dokument auszuführen. Die angegebenen Parameter installieren Docker in der Instance.

```
{
  "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\"}"
```

```
      }
    }
  ]
}
```

Weitere Informationen

- Informationen zum Neustarten von Servern und Instances bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.
- Weitere Informationen zu den Plug-Ins, die Sie einem benutzerdefinierten SSM-Dokument hinzufügen können, finden Sie unter Referenz für Befehlsdokument-Plug-ins.
- Informationen zum Ausführen von Dokumenten von einem Remote-Standort (ohne Erstellen eines zusammengesetzten Dokuments) finden Sie unter Ausführen von -Dokumenten von Remote-Standorten.

Arbeiten mit Dokumenten

Dieser Abschnitt enthält Informationen darüber, wie Sie SSM-Dokumente verwenden und mit ihnen arbeiten können.

Inhalt

- Verwenden von SSM-Dokumenten in State Manager-Zuordnungen
- Vergleichen von SSM-Dokumentversionen
- Erstellen eines SSM-Dokuments (Konsole)
- Erstellen eines SSM-Dokuments (Befehlszeile)
- Erstellen eines SSM-Dokuments (API)
- Löschen benutzerdefinierter SSM-Dokumente
- Ausführen von -Dokumenten von Remote-Standorten
- Freigeben von SSM-Dokumenten
- Suchen nach SSM-Dokumenten

Verwenden von SSM-Dokumenten in State Manager-Zuordnungen

Wenn Sie ein SSM-Dokument fürState Manager, eine Fähigkeit von, erstellen AWS Systems Manager, müssen Sie das Dokument Ihren verwalteten Instanzen zuordnen, nachdem Sie das

Dokument dem System hinzugefügt haben. Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

Beachten Sie die folgenden Details, wenn Sie SSM-Dokumente in State Manager-Zuordnungen verwenden.

- Sie können einem Ziel mehrere Dokumente zuweisen, indem Sie verschiedene State Manager-Zuweisungen erstellen, die verschiedene Dokumente verwenden.
- Wenn Sie ein Dokument mit Plug-in-Verweisen erstellen, die miteinander in Konflikt stehen (z. B. Plug-in für einen Domain-Beitritt und Plug-in zum Entfernen aus einer Domain), befindet sich das System nach Abschluss der Ausführung in dem Zustand, den das letzte Plug-in hergestellt hat. State Manager überprüft weder die logische Abfolge noch die Semantik der Befehle oder Plug-ins in Ihrem Dokument.
- Bei der Verarbeitung von Dokumenten werden zuerst Instance-Verknüpfungen und dann die Verknüpfungen von getaggten Gruppen angewendet. Wenn eine Instance Teil mehrerer getaggter Gruppen ist, dann werden die Dokumente, die Teil der getaggten Gruppe sind, in keiner bestimmten Reihenfolge ausgeführt. Wenn für eine Instance über ihre Instance-ID mehrere Dokumente direkt als Ziel vorgegeben sind, werden die Dokumente in keine bestimmten Reihenfolge ausgeführt.
- Wenn Sie die Standardversion eines SSM-Richtliniendokuments für State Manager ändern, verwenden alle Zuordnungen, die das Dokument verwenden, ab dem nächsten Mal, wenn Systems Manager die Zuordnung auf die Instance anwendet, die neue Standardversion.
- Wenn Sie eine Zuordnung mit einem SSM-Dokument erstellen, das für Sie freigegeben wurde, und der Besitzer dann die Freigabe des Dokuments für Sie beendet, können Ihre Zuordnungen nicht mehr auf dieses Dokument zugreifen. Wenn jedoch der Besitzer zu einem späteren Zeitpunkt dasselbe SSM-Dokument für Sie erneut freigibt, werden Ihre Zuordnungen diesem automatisch erneut zugewiesen.

Vergleichen von SSM-Dokumentversionen

Sie können die inhaltlichen Unterschiede zwischen den Versionen von AWS Systems Manager (SSM-) Dokumenten in der Systems Manager Manager-Dokumentenkonsole vergleichen. Beim Vergleich von Versionen eines SSM-Dokuments werden Unterschiede zwischen dem Inhalt der Versionen hervorgehoben.

Vergleichen von SSM-Dokumentinhalten (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie in der Dokumentliste das freizugebende Dokument, dessen Inhalt Sie teilen möchten.
4. Wählen Sie auf der Registerkarte Content (Inhalt) die Option Compare versions (Versionen vergleichen) und wählen Sie die Version des Dokuments aus, mit der Sie den Inhalt vergleichen möchten.

Erstellen eines SSM-Dokuments (Konsole)

Nachdem Sie den Inhalt wie unter Schreiben von SSM-Dokumentinhalt beschrieben für das benutzerdefinierte SSM-Dokument erstellt haben, können Sie mithilfe der Systems Manager-Konsole ein SSM-Dokument mit Ihrem Inhalt erstellen.

Erstellen eines SSM-Dokuments (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie Create command or session (Befehl oder Sitzung erstellen) aus.
4. Geben Sie einen aussagekräftigen Namen für das Dokument ein.
5. (Optional) Geben Sie in Target type (Zieltyp) den Typ der Ressourcen an, auf denen das Dokument ausgeführt werden kann.
6. Wählen Sie in der Liste Document type den Typ des zu erstellenden Dokuments aus.
7. Löschen Sie die Klammern im Feld Content (Inhalt) und fügen Sie dann den zuvor erstellten Dokumentinhalt ein.
8. (Optional) Wenden Sie im Abschnitt Document tags (Dokument-Tags) ein oder mehrere Tag-Schlüssel-Name/Wert-Paare auf das Dokument an.

   Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Möglicherweise möchten Sie ein Dokument markieren, um den Typ der von ihm ausgeführten Aufgaben, den Typ der Betriebssysteme, auf die es ausgerichtet ist, und die

Umgebung, in der es ausgeführt wird, zu identifizieren. In diesem Fall könnten Sie z.B. die folgenden Schlüsselname-Wert-Paare angeben:

- `Key=TaskType,Value=MyConfigurationUpdate`
- `Key=OS,Value=AMAZON_LINUX_2`
- `Key=Environment,Value=Production`

Weitere Informationen über das Taggen von System Manager-Ressourcen finden Sie unter Markieren von Systems Manager-Ressourcen.

9. Wählen Sie Create document aus, um das Dokument zu speichern.

Erstellen eines SSM-Dokuments (Befehlszeile)

Nachdem Sie den Inhalt für Ihr benutzerdefiniertes Dokument AWS Systems Manager (SSM) erstellt haben, wie unter beschriebenSchreiben von SSM-Dokumentinhalt, können Sie das AWS Command Line Interface (AWS CLI) oder verwenden, AWS Tools for PowerShell um ein SSM-Dokument mit Ihren Inhalten zu erstellen. Das wird im folgenden Befehl veranschaulicht.

Bevor Sie beginnen

Installieren und konfigurieren Sie das AWS CLI oder das AWS Tools for PowerShell, falls Sie das noch nicht getan haben. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS Tools for PowerShell.

Führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

Linux & macOS

```
aws ssm create-document \
--content file://path/to/file/documentContent.json \
--name "document-name" \
--document-type "Command" \
--tags "Key=tag-key,Value=tag-value"
```

Windows

```
aws ssm create-document ^
--content file://C:\path\to\file\documentContent.json ^
```

```
--name "document-name" ^
--document-type "Command" ^
--tags "Key=tag-key,Value=tag-value"
```

PowerShell

```
$json = Get-Content -Path "C:\path\to\file\documentContent.json" | Out-String
New-SSMDocument `
-Content $json `
-Name "document-name" `
-DocumentType "Command" `
-Tags "Key=tag-key,Value=tag-value"
```

Bei erfolgreicher Ausführung gibt der Befehl eine Antwort zurück, die in etwa wie folgt aussieht:

```
{
"DocumentDescription":{
  "CreatedDate":1.585061751738E9,
  "DefaultVersion":"1",
  "Description":"MyCustomDocument",
  "DocumentFormat":"JSON",
  "DocumentType":"Command",
  "DocumentVersion":"1",
  "Hash":"0d3d879b3ca072e03c12638d0255ebd004d2c65bd318f8354fcde820dEXAMPLE",
  "HashType":"Sha256",
  "LatestVersion":"1",
  "Name":"Example",
  "Owner":"111122223333",
  "Parameters":[
     --truncated--
  ],
  "PlatformTypes":[
     "Windows",
     "Linux"
  ],
  "SchemaVersion":"0.3",
  "Status":"Creating",
  "Tags": [
        {
            "Key": "Purpose",
            "Value": "Test"
        }
```

```
    ]
}
}
```

Erstellen eines SSM-Dokuments (API)

Nachdem Sie den Inhalt für Ihr benutzerdefiniertes Dokument AWS Systems Manager (SSM) erstellt haben, können Sie, wie unter beschriebenSchreiben von SSM-Dokumentinhalt, Ihr bevorzugtes SDK verwenden, um den AWS Systems Manager CreateDocumentAPI-Vorgang zum Erstellen eines SSM-Dokuments mit Ihrem Inhalt aufzurufen. Die JSON- oder YAML-Zeichenfolge für den `Content`-Anforderungsparameter wird in der Regel aus einer Datei gelesen. Mit den folgenden Beispielfunktionen wird ein SSM-Dokument mit den SDKs für Python, Go und Java erstellt.

Python

```
import boto3

ssm = boto3.client('ssm')
filepath = '/path/to/file/documentContent.yaml'

def createDocumentApiExample():
with open(filepath) as openFile:
    documentContent = openFile.read()
    createDocRequest = ssm.create_document(
        Content = documentContent,
        Name = 'createDocumentApiExample',
        DocumentType = 'Automation',
        DocumentFormat = 'YAML'
    )
    print(createDocRequest)

createDocumentApiExample()
```

Go

```
package main

import (
"github.com/aws/aws-sdk-go/aws"
"github.com/aws/aws-sdk-go/aws/session"
```

```
"github.com/aws/aws-sdk-go/service/ssm"

"fmt"
"io/ioutil"
"log"
)


func main() {
openFile, err := ioutil.ReadFile("/path/to/file/documentContent.yaml")
if err != nil {
    log.Fatal(err)
}
documentContent := string(openFile)
sesh := session.Must(session.NewSessionWithOptions(session.Options{
    SharedConfigState: session.SharedConfigEnable}))

ssmClient := ssm.New(sesh)
createDocRequest, err := ssmClient.CreateDocument(&ssm.CreateDocumentInput{
    Content: &documentContent,
    Name:  aws.String("createDocumentApiExample"),
    DocumentType: aws.String("Automation"),
    DocumentFormat: aws.String("YAML"),
})
result := *createDocRequest
fmt.Println(result)
}
```

Java

```


import java.io.IOException;
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;

import com.amazonaws.AmazonClientException;
import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.simplesystemsmanagement.AWSSimpleSystemsManagement;
```

```
import
 com.amazonaws.services.simplesystemsmanagement.AWSSimpleSystemsManagementClientBuilder;
import com.amazonaws.services.simplesystemsmanagement.model.*;

public class createDocumentApiExample {
public static void main(String[] args) {
try {
  createDocumentMethod(getDocumentContent());
}
catch (IOException e) {
  e.printStackTrace();
}
}
public static String getDocumentContent() throws IOException {
  String filepath = new String("/path/to/file/documentContent.yaml");
    byte[] encoded = Files.readAllBytes(Paths.get(filepath));
    String documentContent = new String(encoded, StandardCharsets.UTF_8);
    return documentContent;
}

public static void createDocumentMethod (final String documentContent) {
  AWSSimpleSystemsManagement ssm =
 AWSSimpleSystemsManagementClientBuilder.defaultClient();
  final CreateDocumentRequest createDocRequest = new CreateDocumentRequest()
    .withContent(documentContent)
    .withName("createDocumentApiExample")
    .withDocumentType("Automation")
    .withDocumentFormat("YAML");
  final CreateDocumentResult result = ssm.createDocument(createDocRequest);
}
}
```

Weitere Informationen zum Erstellen von benutzerdefinierten Dokumentinhalt finden Sie unter Datenelemente und Parameter.

Löschen benutzerdefinierter SSM-Dokumente

Wenn Sie ein benutzerdefiniertes SSM-Dokument nicht mehr verwenden möchten, können Sie es entweder mit der AWS Command Line Interface (AWS CLI) oder der AWS Systems Manager Konsole löschen.

Löschen eines SSM-Dokuments (AWS CLI)

1. Bevor Sie das Dokument löschen, sollten Sie die Zuordnung aller Instances aufheben, die dem Dokument zugeordnet sind.

   Führen Sie den folgenden Befehl aus, um die Zuordnung einer Instance zu einem Dokument aufzuheben.

   ```
   aws ssm delete-association --instance-id "123456789012" --name "documentName"
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux

   ```
   aws ssm delete-document \
       --name "document-name" \
       --document-version "document-version" \
       --version-name "version-name"
   ```

   Windows

   ```
   aws ssm delete-document ^
       --name "document-name" ^
       --document-version "document-version" ^
       --version-name "version-name"
   ```

   PowerShell

   ```
   Delete-SSMDocument `
       -Name "document-name" `
       -DocumentVersion 'document-version' `
       -VersionName 'version-name'
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

> **Important**
>
> Wenn das Symbol `document-version` oder der `version-name` nicht zur bereitgestellt werden, werden alle Versionen des Dokuments gelöscht

Löschen eines SSM-Dokuments (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie das Dokument aus, die Sie löschen möchten.
4. Wählen Sie Löschen. Wenn Sie zum Löschen des Dokuments aufgefordert werden, wählen Sie Löschen.

Ausführen von -Dokumenten von Remote-Standorten

Sie können AWS Systems Manager (SSM-) Dokumente von entfernten Standorten aus ausführen, indem Sie das `AWS-RunDocument` vordefinierte SSM-Dokument verwenden. Dieses Dokument unterstützt die Ausführung von SSM-Dokumenten, die an den folgenden Speicherorten gespeichert sind:

- Öffentliche und private GitHub Repositorien (wird nicht unterstützt) GitHub Enterprise
- Amazon-S3-Buckets
- Systems Manager

Sie können zwar auch Remote-Dokumente mithilfe von State Manager oder Automation, Funktionen von ausführen, aber im folgenden Verfahren wird nur beschrieben AWS Systems Manager, wie Sie Remote-SSM-Dokumente mithilfe AWS Systems Manager Run Command der Systems Manager Manager-Konsole ausführen.

> **Note**
>
> `AWS-RunDocument` kann verwendet werden, um nur SSM-Dokumente vom Befehlstyp auszuführen, nicht andere Typen wie Automation-Runbooks. Das `AWS-RunDocument`

verwendet das `aws:downloadContent`-Plugin. Weitere Informationen zum `aws:downloadContent`-Plugin finden Sie unter `aws:downloadContent`.

Bevor Sie beginnen

Bevor Sie ein Remote-Dokument ausführen, müssen Sie die folgenden Aufgaben erledigen.

- Erstellen Sie ein SSM-Befehlsdokument und speichern Sie es an einem Remote-Standort. Weitere Informationen finden Sie unter Erstellen von SSM-Dokumentinhalten
- Wenn Sie ein Remote-Dokument ausführen möchten, das in einem privaten GitHub Repository gespeichert ist, müssen Sie einen Systems Manager `SecureString` Manager-Parameter für Ihr GitHub Sicherheitszugriffstoken erstellen. Sie können nicht auf ein Remote-Dokument in einem privaten GitHub Repository zugreifen, indem Sie Ihr Token manuell über SSH übergeben. Das Zugriffstoken muss als `SecureString`-Systems Manager-Parameter übertragen werden. Weitere Informationen zum Erstellen eines `SecureString`-Parameters finden Sie unter Erstellen von Systems Manager-Parametern.

Ausführen eines Remote-Dokuments (Konsole)

So führen Sie ein Remote-Dokument aus

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Dokument die Option **`AWS-RunDocument`**.
5. Wählen Sie unter Command parameters (Befehlsparameter) für Source Type (Quelltyp) eine Option aus.

   - Wenn Sie möchten GitHub, geben Sie die Quellinformationen im folgenden Format an:

```
{
    "owner": "owner_name",
    "repository": "repository_name",
    "path": "path_to_document",
    "getOptions":"branch:branch_name",
    "tokenInfo": "{{ssm-secure:secure-string-token}}"
```

```
}
```

Beispielsweise:

```
{
    "owner":"TestUser",
    "repository":"GitHubTestExamples",
    "path":"scripts/python/test-script",
    "getOptions":"branch:exampleBranch",
    "tokenInfo":"{{ssm-secure:my-secure-string-token}}"
}
```

> **Note**
>
> `getOptions` sind zusätzliche Optionen zum Abrufen von Inhalten aus einem anderen Branch als dem Master-Branch oder aus einem bestimmten Commit im Repository. `getOptions` kann weggelassen werden, wenn Sie den letzten Commit in der Master-Branch verwenden. Der `branch`-Parameter ist nur erforderlich, wenn Ihr SSM-Dokument in einer anderen Verzweigung als `master` gespeichert ist.
> Um die Version Ihres SSM-Dokuments in einem bestimmten Commit in Ihrem Repository zu verwenden, verwenden Sie `commitID` mit `getOptions`statt `branch`. Zum Beispiel:
>
> ```
> "getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
> ```

- Wenn Sie S3 auswählen, geben Sie für Source Info Informationen in folgendem Format an:

```
{"path":"URL_to_document_in_S3"}
```

Zum Beispiel:

```
{"path":"https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/scripts/ruby/mySSMdoc.json"}
```

- Wenn Sie SSMDocument auswählen, geben Sie für Source Info Informationen in folgendem Format an:

```
{"name": "document_name"}
```

Zum Beispiel:

```
{"name": "mySSMdoc"}
```

6. Geben Sie im Feld Document Parameters Parameter für das Remote-SSM-Dokument ein. Wenn Sie beispielsweise das Dokument `AWS-RunPowerShell` ausführen, könnten Sie Folgendes angeben:

   ```
   {"commands": ["date", "echo \"Hello World\""]}
   ```

   Wenn Sie das Dokument `AWS-ConfigureAWSPack` ausführen, könnten Sie Folgendes angeben:

   ```
   {
      "action":"Install",
      "name":"AWSPVDriver"
   }
   ```

7. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

8. Für Other parameters (Weitere Parameter):

   - Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

9. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

10. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

    > **Note**
    >
    > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

11. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

12. Wählen Sie Ausführen aus.

> **Note**
>
> Informationen zum Neustarten von Servern und Instances bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Freigeben von SSM-Dokumenten

Sie können AWS Systems Manager (SSM) Dokumente privat oder öffentlich mit Konten in demselben AWS-Region teilen. Um ein Dokument privat freizugeben, ändern Sie die Dokumentberechtigungen und erlauben bestimmten Personen entsprechend ihrer AWS-Konto -ID den Zugriff darauf. Wenn Sie ein SSM-Dokument öffentlich freigeben möchten, ändern Sie die Zugriffsberechtigungen des Dokuments und geben `All` an. Dokumente können nicht gleichzeitig öffentlich und privat freigegeben werden.

> **Warning**
>
> Verwenden Sie freigegebene SSM-Dokumente nur, wenn sie aus vertrauenswürdigen Quellen stammen. Wenn Sie ein freigegebenes Dokument verwenden, überprüfen Sie den Inhalt des Dokuments sorgfältig, bevor Sie es verwenden, damit Sie verstehen, wie es die Konfiguration der Instance ändert. Weitere Informationen zu bewährten Methoden für freigegebene Dokumente finden Sie unter Bewährte Methoden für freigegebene SSM-Dokumente.

Einschränkungen

Beachten Sie die folgenden Einschränkungen, wenn Sie zum ersten Mal mit SSM-Dokumenten arbeiten.

- Nur der Eigentümer eines Dokuments kann ein Dokument freigeben.

- Sie müssen die Freigabe eines Dokuments aufheben, bevor Sie ein Dokument löschen können. Weitere Informationen finden Sie unter Modifizieren von Berechtigungen für ein freigegebenes SSM-Dokument.
- Sie können ein Dokument mit maximal 1000 AWS-Konten Personen teilen. Sie können über das AWS Support Center eine Erhöhung dieses Limits anfordern. Wählen Sie unter Limit type (Limittyp) die Option EC2 Systems Manager aus und beschreiben Sie die Gründe für die Anforderung.
- Sie können maximal fünf SSM-Dokumente öffentlich freigeben. Sie können über das AWS Support Center eine Erhöhung dieses Limits anfordern. Wählen Sie unter Limit type (Limittyp) die Option EC2 Systems Manager aus und beschreiben Sie die Gründe für die Anforderung.
- Dokumente können AWS-Region nur mit anderen Konten in demselben Konto geteilt werden. Die Freigabe über Regionsgrenzen hinweg wird nicht unterstützt.

Weitere Informationen zu Service Quotas für Systems Manager finden Sie unter AWS Systems Manager Service Quotas.

Inhalt

- Bewährte Methoden für freigegebene SSM-Dokumente
- Öffentliche Freigabe für SSM-Dokumente blockieren
- Freigeben eines SSM-Dokuments
- Modifizieren von Berechtigungen für ein freigegebenes SSM-Dokument
- Verwenden von freigegebenen SSM-Dokumenten

Bewährte Methoden für freigegebene SSM-Dokumente

Überprüfen Sie die folgenden Richtlinien, bevor Sie ein Dokument freigeben oder ein gemeinsam genutztes Dokument verwenden.

Entfernen sensibler Daten

> Überprüfen Sie Ihr AWS Systems Manager (SSM-) Dokument sorgfältig und entfernen Sie alle vertraulichen Informationen. Stellen Sie beispielsweise sicher, dass das Dokument Ihre AWS Anmeldeinformationen nicht enthält. Wenn Sie ein Dokument für bestimmten Personen freigeben, können die Informationen in dem Dokument anzeigen. Wenn Sie ein Dokument öffentlich freigeben, können beliebige Personen die Informationen in dem Dokument anzeigen.

Öffentliche Freigabe für Dokumente blockieren

Sofern für Ihren Anwendungsfall keine öffentliche Freigabe erforderlich ist, empfehlen wir Ihnen, die Einstellung zum Blockieren der öffentlichen Freigabe für Ihre Systems Manager-Dokumente im Abschnitt Preferences (Einstellungen) der Systems Manager-Dokumentenkonsole zu aktivieren.

Beschränken von Run Command-Aktionen mithilfe einer IAM-Vertrauensrichtlinie

Erstellen Sie eine restriktive Richtlinie AWS Identity and Access Management (IAM) für Benutzer, die Zugriff auf das Dokument haben werden. Die IAM-Richtlinie bestimmt, welche SSM-Dokumente ein Benutzer entweder in der Amazon Elastic Compute Cloud (Amazon EC2) - Konsole oder durch Aufrufen `ListDocuments` mit AWS Command Line Interface (AWS CLI) oder sehen kann. AWS Tools for Windows PowerShell Die Richtlinie schränkt auch die Aktionen ein, die der Benutzer mit SSM-Dokumenten durchführen kann. Sie können eine restriktive Richtlinie erstellen, damit Benutzer nur bestimmte Dokumente verwenden können. Weitere Informationen finden Sie unter Beispiele für vom Kunden verwaltete Richtlinien.

Vorsicht bei der Verwendung freigegebener SSM-Dokumente

Überprüfen Sie den Inhalt jedes Dokuments, das für Sie freigegeben ist, insbesondere öffentliche Dokumente, um die Befehle zu verstehen, die über Ihre Instances ausgeführt werden. Ein Dokument kann absichtlich oder unbeabsichtigterweise negative Auswirkungen haben, wenn es ausgeführt wird. Wenn das Dokument auf ein externes Netzwerk verweist, überprüfen Sie die externe Quelle, bevor Sie das Dokument verwenden.

Versenden von Befehlen mit dem Dokument-Hash

Wenn Sie ein Dokument freigeben, erstellt das System einen SHA-256-Hash und weist diesen dem Dokument zu. Das System speichert außerdem einen Snapshot des Dokumentinhalts. Wenn Sie mit einem freigegebenen Dokument einen Befehl senden, können Sie für den Befehl diesen Hash angeben, um sicherzustellen, dass die folgenden Bedingungen erfüllt sind:

- Sie führen den Befehl über das richtige Systems Manager-Dokument aus.
- Der Inhalt des Dokuments wurde nicht geändert, seit es für Sie freigegeben wurde.

Wenn der Hash nicht mit dem angegebenen Dokument übereinstimmt oder der Inhalt des freigegebenen Dokuments geändert wurde, löst der Befehl eine `InvalidDocument`-Ausnahmebedingung aus. Mit dem Hash können keine Dokumentinhalte von externen Standorten überprüft werden.

Öffentliche Freigabe für SSM-Dokumente blockieren

Sofern Ihr Anwendungsfall nicht erfordert, dass das öffentliche Teilen aktiviert ist, empfehlen wir, die Einstellung „Öffentliches Teilen blockieren“ für Ihre AWS Systems Manager (SSM-) Dokumente zu aktivieren. Wenn Sie diese Einstellung aktivieren, wird unerwünschter Zugriff auf Ihre SSM-Dokumente verhindert. Bei der Einstellung „Öffentliches Teilen blockieren“ handelt es sich um eine Einstellung auf Kontoebene, die für jedes AWS-Region Konto unterschiedlich sein kann. Führen Sie die folgenden Schritte aus, um die öffentliche Freigabe für Ihre SSM-Dokumente zu blockieren.

Öffentliche Freigabe blockieren (Konsole)

Blockieren der öffentlichen Freigabe Ihrer SSM-Dokumente

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie Preferences (Präferenzen) und dann Edit (Bearbeiten) im Abschnitt Block public sharing (Öffentliche Freigabe blockieren) .
4. Wählen Sie das Kontrollkästchen Block public sharing (Öffentliche Freigabe blockieren) und wählen Sie aus Save (speichern).

Öffentliche Freigabe blockieren (Befehlszeile)

Öffnen Sie AWS Command Line Interface (AWS CLI) oder AWS Tools for Windows PowerShell auf Ihrem lokalen Computer und führen Sie den folgenden Befehl aus, um das öffentliche Teilen Ihrer SSM-Dokumente zu blockieren.

Linux & macOS

```
aws ssm update-service-setting  \
    --setting-id /ssm/documents/console/public-sharing-permission \
    --setting-value Disable \
    --region 'The AWS-Region 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 AWS-Region you want to block public sharing in"
```

PowerShell

```
Update-SSMServiceSetting `
    -SettingId /ssm/documents/console/public-sharing-permission `
    -SettingValue Disable `
    -Region The AWS-Region you want to block public sharing in
```

Überprüfen Sie, ob der Einstellungswert aktualisiert wurde, indem Sie den folgenden Befehl verwenden.

Linux & macOS

```
aws ssm get-service-setting   \
    --setting-id /ssm/documents/console/public-sharing-permission \
    --region The AWS-Region you blocked public sharing in
```

Windows

```
aws ssm get-service-setting  ^
    --setting-id /ssm/documents/console/public-sharing-permission ^
    --region "The AWS-Region you blocked public sharing in"
```

PowerShell

```
Get-SSMServiceSetting `
    -SettingId /ssm/documents/console/public-sharing-permission `
    -Region The AWS-Region you blocked public sharing in
```

Beschränken des Zugriffs zum Blockieren der öffentlichen Freigabe mit IAM

Sie können AWS Identity and Access Management (IAM) -Richtlinien erstellen, die Benutzer daran hindern, die Einstellung „Öffentliches Teilen blockieren“ zu ändern. Dadurch wird verhindert, dass Benutzer unerwünschten Zugriff auf Ihre SSM-Dokumente zulassen.

Nachfolgend finden Sie ein Beispiel für eine IAM-Richtlinie, die verhindert, dass Benutzer die Einstellung zum Blockieren der öffentlichen Freigabe zu aktualisieren. Um dieses Beispiel zu

verwenden, müssen Sie die Beispiel-Konto-ID für Amazon Web Services durch Ihre eigene Konto-ID ersetzen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:UpdateServiceSetting",
            "Resource": "arn:aws:ssm:*:987654321098:servicesetting/ssm/documents/
console/public-sharing-permission"
        }
    ]
}
```

Freigeben eines SSM-Dokuments

Sie können AWS Systems Manager (SSM) Dokumente mithilfe der Systems Manager Manager-Konsole teilen. Beim Teilen von Dokumenten über die Konsole kann nur die Standardversion des Dokuments geteilt werden. Sie können SSM-Dokumente auch programmgesteuert teilen, indem Sie die `ModifyDocumentPermission` API-Operation mit dem AWS Command Line Interface (AWS CLI) AWS Tools for Windows PowerShell, oder dem SDK aufrufen. AWS Rufen Sie, bevor Sie ein Dokument freigeben, die AWS-Konto -IDs der Personen ab, für die Sie das Dokument freigeben möchten. Sie müssen diese Konto-IDs angeben, wenn Sie das Dokument freigeben.

Freigeben eines Dokuments (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie in der Dokumentenliste das Dokument aus, das Sie freigeben möchten, und klicken Sie dann auf View details (Details anzeigen). Überprüfen Sie dann auf der Registerkarte Permissions, ob Sie der Besitzer des Dokuments sind. Nur der Eigentümer eines Dokuments kann ein Dokument freigeben.
4. Wählen Sie Bearbeiten aus.
5. Um den Befehl öffentlich freizugeben, wählen Sie Public und dann die Option Save. Wählen Sie zur privaten Freigabe des Befehls die Option Private aus, geben Sie die AWS-Konto -ID ein und wählen Sie Add permission sowie anschließend die Option Save aus.

Freigeben eines Dokuments (Befehlszeile)

Das folgende Verfahren erfordert, dass Sie eine AWS-Region für Ihre Befehlszeilensitzung angeben.

1. Öffnen Sie AWS CLI oder AWS Tools for Windows PowerShell auf Ihrem lokalen Computer und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen anzugeben.

   Ersetzen Sie im folgenden Befehl *region* mit Ihren eigenen Informationen. Eine Liste der unterstützten *Region*-Werte finden Sie in der Spalte Region unter Service-Endpunkte von Systems Manager in der Allgemeine Amazon Web Services-Referenz.

   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. Verwenden Sie den folgenden Befehl, um alle SSM-Dokumente aufzulisten, die für Sie verfügbar sind. Die Liste enthält Dokumente, die Sie erstellt haben, und Dokumente, die für Sie freigegeben wurden.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

Windows

```
aws ssm list-documents
```

PowerShell

```
Get-SSMDocumentList
```

3. Verwenden Sie den folgenden Befehl, um ein bestimmtes Dokument abzurufen.

Linux & macOS

```
aws ssm get-document \
    --name document name
```

Windows

```
aws ssm get-document ^
    --name document name
```

PowerShell

```
Get-SSMDocument `
    -Name document name
```

4. Verwenden Sie den folgenden Befehl, um eine Beschreibung des Dokuments abzurufen.

Linux & macOS

```
aws ssm describe-document \
    --name document name
```

Windows

```
aws ssm describe-document ^
    --name document name
```

PowerShell

```
Get-SSMDocumentDescription `
    -Name document name
```

5. Verwenden Sie den folgenden Befehl, um die Zugriffsberechtigungen für das Dokument anzuzeigen.

   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. Verwenden Sie den folgenden Befehl, um die Zugriffsberechtigungen für das Dokument zu ändern und das Dokument freizugeben. Sie müssen der Eigentümer des Dokuments sein, um die Berechtigungen bearbeiten zu können. Optional können Sie mithilfe des `--shared-document-version`-Parameters eine Version des Dokuments angeben, die Sie teilen möchten. Wenn Sie keine Version angeben, gibt das System die `Default`-Version des Dokuments frei. Mit diesem Beispielbefehl wird das Dokument privat für eine bestimmte Person freigegeben, auf der Grundlage der AWS-Konto -ID der Person.

   Linux & macOS

   ```
   aws ssm modify-document-permission \
       --name document name \
       --permission-type Share \
   ```

```
    --account-ids-to-add AWS-Konto ID
```

Windows

```
aws ssm modify-document-permission ^
    --name document name ^
    --permission-type Share ^
    --account-ids-to-add AWS-Konto ID
```

PowerShell

```
Edit-SSMDocumentPermission `
    -Name document name `
    -PermissionType Share `
    -AccountIdsToAdd AWS-Konto ID
```

7. Verwenden Sie den folgenden Befehl, um ein Dokument öffentlich freizugeben.

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')
```

Modifizieren von Berechtigungen für ein freigegebenes SSM-Dokument

Wenn Sie einen Befehl gemeinsam nutzen, können Benutzer diesen Befehl anzeigen und verwenden, bis Sie entweder den Zugriff auf das AWS Systems Manager (SSM-) Dokument aufheben oder das SSM-Dokument löschen. Sie können ein Dokument jedoch erst löschen, wenn es nicht mehr freigegeben ist. Sie müssen also zuerst die Freigabe beenden und können erst anschließend die Datei löschen.

Beenden der Freigabe eines Dokuments (Konsole)

Beenden der Freigabe eines Dokuments

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie in der Dokumentenliste das Dokument aus, das Sie nicht mehr teilen möchten, und klicken Sie dann auf Details. Vergewissern Sie sich im Abschnitt Berechtigungen, dass Sie der Eigentümer des Dokuments sind. Nur der Eigentümer eines Dokuments kann die Freigabe eines Dokuments beenden.
4. Wählen Sie Bearbeiten aus.
5. Wählen Sie X aus, um die AWS-Konto ID zu löschen, die keinen Zugriff mehr auf den Befehl haben sollte, und wählen Sie dann Speichern.

Beenden der Freigabe eines Dokuments (Befehlszeile)

Öffnen Sie AWS CLI oder AWS Tools for Windows PowerShell auf Ihrem lokalen Computer und führen Sie den folgenden Befehl aus, um die gemeinsame Nutzung eines Befehls zu beenden.

Linux & macOS

```
aws ssm modify-document-permission \
    --name document name \
    --permission-type Share \
    --account-ids-to-remove 'AWS-Konto ID'
```

Windows

```
aws ssm modify-document-permission ^
    --name document name ^
```

```
    --permission-type Share ^
    --account-ids-to-remove "AWS-Konto ID"
```

PowerShell

```
Edit-SSMDocumentPermission `
    -Name document name `
    -PermissionType Share `
    -AccountIdsToRemove AWS-Konto ID
```

Verwenden von freigegebenen SSM-Dokumenten

Wenn Sie ein AWS Systems Manager (SSM) -Dokument teilen, generiert das System einen Amazon-Ressourcennamen (ARN) und weist ihn dem Befehl zu. Wenn Sie ein freigegebenes Dokument über die Systems-Manager-Konsole auswählen und ausführen, wird der ARN nicht angezeigt. Wenn Sie jedoch ein freigegebenes SSM-Dokument mit einer anderen Methode als der Systems-Manager-Konsole ausführen möchten, müssen Sie den vollständigen ARN des Dokuments für den `DocumentName`-Anforderungsparameter angeben. Wenn Sie den Befehl zum Auflisten der Dokumente ausführen, wird jeweils der vollständige ARN für SSM-Dokumente angezeigt.

> **Note**
>
> Sie müssen keine ARNs für AWS öffentliche Dokumente (Dokumente, die mit beginnen`AWS-*`) oder Dokumente, die Ihnen gehören, angeben.

Verwenden eines freigegebenen SSM-Dokuments (Befehlszeile)

So listen Sie öffentliche SSM-Dokumente auf

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)
```

So listen Sie private SSM-Dokumente auf, die für Sie freigegeben wurden

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)
```

So listen Sie alle SSM-Dokumente auf, die für Sie verfügbar sind

Linux & macOS

```
aws ssm list-documents
```

Windows

```
aws ssm list-documents
```

PowerShell

```
Get-SSMDocumentList
```

So rufen Sie Informationen zu einem SSM-Dokument ab, das für Sie freigegeben wurde

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
```

So führen Sie ein freigegebenes SSM-Dokument aus

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
```

Suchen nach SSM-Dokumenten

Sie können den AWS Systems Manager (SSM-) Dokumentenspeicher nach SSM-Dokumenten durchsuchen, indem Sie entweder die Freitextsuche oder eine filterbasierte Suche verwenden. Sie können auch Dokumente als Favoriten markieren, um häufig verwendete SSM-Dokumente zu finden. In den folgenden Abschnitten wird beschrieben, wie Sie diese Funktionen nutzen können.

Verwenden der Freitextsuche

Das Suchfeld auf der Seite Systems Manager-Dokumente unterstützt die Freitextsuche. Die Freitextsuche vergleicht den bzw. die eingegebenen Suchbegriffe mit dem Dokumentnamen in jedem SSM-Dokument. Wenn Sie einen einzelnen Suchbegriff eingeben, z. B. **`ansible`**, gibt Systems Manager alle SSM-Dokumente zurück, in denen dieser Begriff erkannt wurde. Wenn Sie mehrere Suchbegriffe eingeben, sucht Systems Manager mithilfe einer `OR`-Anweisung. Wenn Sie z. B. **`ansible`** und **`linux`** angeben, gibt die Suche alle Dokumente zurück, die eines der beiden Schlüsselwörter im Namen tragen.

Wenn Sie einen Freitext-Suchbegriff eingeben und eine Suchoption wählen, z. B. Plattformtyp, dann verwendet die Suche eine `AND`-Anweisung und gibt alle Dokumente zurück, die das Schlüsselwort im Namen und den angegebenen Plattformtyp enthalten.

> **Note**
>
> Beachten Sie die folgenden Details zur Freitextsuche.
>
> - Bei der Freitextsuche wird nicht zwischen Groß- und Kleinschreibung unterschieden.
> - Die Suchbegriffe müssen mindestens drei und dürfen höchstens 20 Zeichen lang sein.
> - Die Freitextsuche akzeptiert bis zu fünf Suchbegriffe.
> - Wenn Sie ein Leerzeichen zwischen den Suchbegriffen eingeben, schließt das System das Leerzeichen bei der Suche ein.
> - Sie können die Freitextsuche mit anderen Suchoptionen wie Dokumenttyp oder Plattformtyp kombinieren.
> - Der Filter Dokumentname-Präfix und die Freitextsuche können nicht zusammen verwendet werden, da sie sich gegenseitig ausschließen.

Suchen nach SSM-Dokumenten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Geben Sie Ihre Suchbegriffe in das Suchfeld ein und drücken Sie die Eingabetaste.

Durchführen einer Freitextdokumentsuche mit dem AWS CLI

Durchführen der Freitextdokumentsuche mithilfe der CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um eine Freitextdokumentsuche mit einem einzelnen Begriff durchzuführen. Ersetzen Sie in diesem Befehl *search_term* mit Ihren eigenen Informationen.

   ```
   aws ssm list-documents --filters Key="SearchKeyword",Values="search_term"
   ```

   Ein Beispiel:

   ```
   aws ssm list-documents --filters Key="SearchKeyword",Values="aws-asg" --region us-east-2
   ```

   Um mit mehreren Begriffen zu suchen, die eine AND-Anweisung erstellen, führen Sie den folgenden Befehl aus. Ersetzen Sie in diesem Befehl *search_term_1* und *search_term_2* mit Ihren eigenen Informationen.

   ```
   aws ssm list-documents --filters
    Key="SearchKeyword",Values="search_term_1","search_term_2","search_term_3" --region us-east-2
   ```

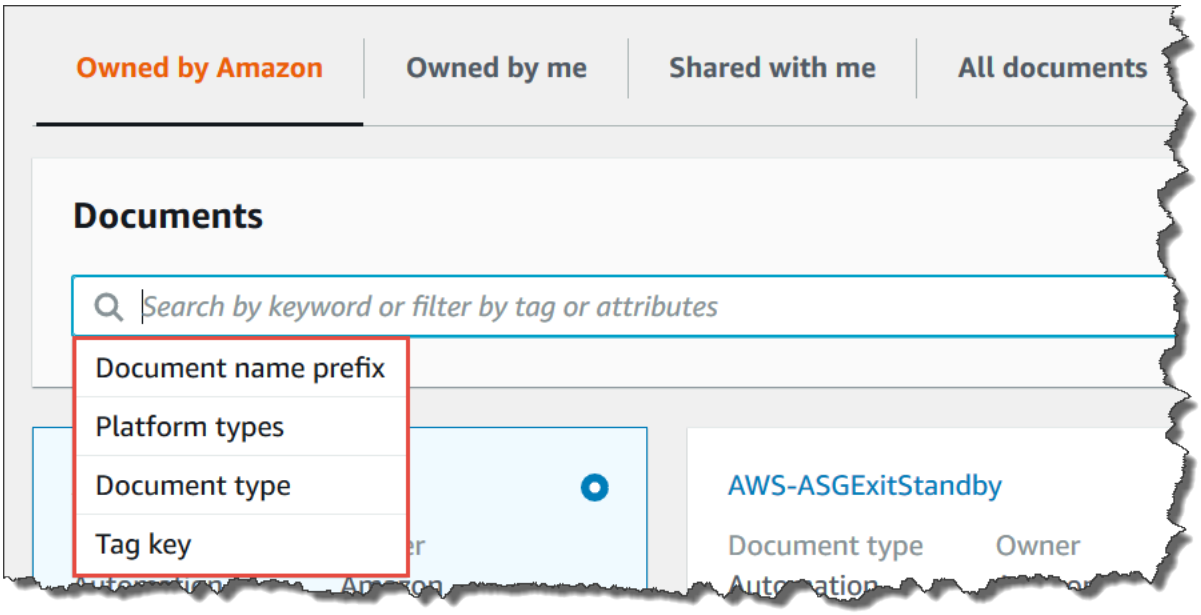
   Ein Beispiel:

```
aws ssm list-documents --filters Key="SearchKeyword",Values="aws-asg","aws-
ec2","restart" --region us-east-2
```

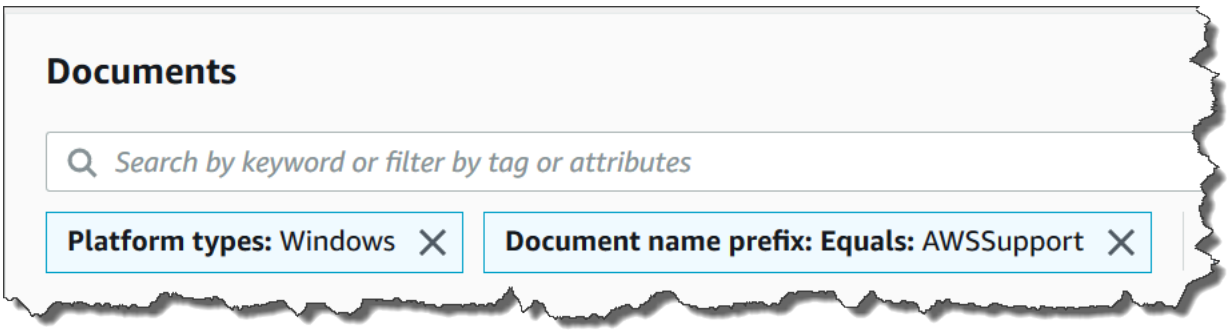
Verwenden von Filtern

Der Systems Manager–Seite Dokumente zeigt automatisch die folgenden Filter an, wenn Sie das Suchfeld auswählen.

- Dokumentnamenpräfix
- Plattformentypen
- Dokumenttyp
- Tag-Schlüssel



Sie können mit einem einzigen Filter nach SSM-Dokumenten suchen. Wenn Sie einen spezifischeren Satz von SSM-Dokumenten zurückgeben möchten, können Sie mehrere Filter anwenden. Hier ein Beispiel für eine Suche, bei der die Filter Plattformtypen und Dokumentnamenpräfix verwendet werden.



Wenn Sie mehrere Filter anwenden, erstellt Systems Manager verschiedene Suchanweisungen basierend auf den ausgewählten Filtern:

- Wenn Sie denselben Filter mehrfach anwenden, z. B. das Präfix für den Dokumentennamenpräfix, sucht Systems Manager mit Hilfe einer OR-Anweisung. Wenn Sie z. B. einen Filter Dokumentname Präfix=**`AWS`** und einen zweiten Filter Dokumentnamenpräfix=**`Lambda`** angeben, liefert die Suche alle Dokumente mit dem Präfix „AWS“ und alle Dokumente mit dem Präfix „Lambda“.
- Wenn Sie verschiedene Filter anwenden, z. B. Document name prefix (Präfix Dokumentname) und Platform types (Plattformtypen), sucht Systems Manager mithilfe einer AND-Anweisung. Wenn Sie z. B. den Filter Document name prefix (Präfix Dokumentname) = **`AWS`** und den Filter Platform types (Plattformtypen) = **`Linux`** angeben, gibt die Suche alle Dokumente mit dem Präfix „AWS“ zurück, die spezifisch für die Linux-Plattform sind.

> **Note**
>
> Dabei wird Groß- und Kleinschreibung beachtet.

Hinzufügen von Dokumenten zu Ihren Favoriten

Fügen Sie Dokumente zu Ihren Favoriten hinzu, um häufig verwendete SSM-Dokumente leichter zu finden. Sie können bis zu 20 Dokumente pro Dokumenttyp, pro AWS-Konto und als Favorit markieren AWS-Region. Sie können Ihre Favoriten in der Dokumenten- AWS Management Console auswählen, ändern und anzeigen. Die folgenden Verfahren beschreiben, wie Sie Ihre Favoriten auswählen, ändern und anzeigen.

So markieren Sie ein SSM-Dokument als Favorit

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie das Sternsymbol neben dem Namen des Dokuments aus, das Sie als Favorit markieren möchten.

So entfernen Sie ein SSM-Dokument aus Ihren Favoriten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie das Sternsymbol neben dem Namen des Dokuments ab, das Sie nicht mehr als Favorit markieren möchten.

Um Ihre Favoriten aus den Dokumenten anzusehen AWS Management Console

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie die Registerkarte Favoriten aus.

Sicherheit in AWS Systems Manager

Cloud-Sicherheit genießt bei Amazon Web Services höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die eingerichtet wurde, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud selbst – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Informationen zu den Compliance-Programmen, die für AWS Systems Manager gelten, finden Sie unter AWS-Services im Rahmen nach Compliance-Programm.
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, die Anforderungen Ihres Unternehmens und die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der geteilten Verantwortung bei der Verwendung von AWS Systems Manager einsetzen können. Die folgenden Themen veranschaulichen, wie Sie Systems Manager zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren außerdem, wie Sie andere AWS-Services verwenden, um Ihre Systems Manager-Ressourcen zu überwachen und zu schützen.

Themen

- Datenschutz in AWS Systems Manager
- Identity and Access Management für AWS Systems Manager
- Verwenden von serviceverknüpften Rollen für Systems Manager
- Protokollieren und Überwachen in AWS Systems Manager
- Compliance-Validierung für AWS Systems Manager
- Ausfallsicherheit in AWS Systems Manager
- Sicherheit der Infrastruktur in AWS Systems Manager
- Konfigurations- und Schwachstellenanalyse in AWS Systems Manager
- Bewährte Methoden für die Sicherheit für Systems Manager

Datenschutz in AWS Systems Manager

Datenschutz bezieht sich auf den Schutz von Daten während der Übertragung (beim Hin- und Systems Manager Rücktransport) und im Ruhezustand (während sie in AWS Rechenzentren gespeichert werden).

Das AWS Modell der der , gilt für den Datenschutz inAWS Systems Manager. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der API Systems Manager oder den SDKs arbeiten oder diese anderweitig AWS-

Services verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Datenverschlüsselung

Verschlüsselung im Ruhezustand

Parameter Store parameters

Die Arten von Parametern, die Sie in Parameter Store, eine Funktion von AWS Systems Manager, erstellen können, beinhalten `String` ,`StringList`, und `SecureString`.

Parameter StoreVerwendet ein AWS KMS key in AWS Key Management Service ()AWS KMS, um `SecureString` Parameterwerte zu verschlüsseln. AWS KMS verwendet entweder einen vom Kunden verwalteten Schlüssel oder einen Von AWS verwalteter Schlüssel , um den Parameterwert in einer AWS verwalteten Datenbank zu verschlüsseln.

> **Important**
>
> Speichern Sie keine vertraulichen Daten in einem `String`- oder `StringList`-Parameter. Verwenden Sie für alle vertraulichen Daten, die verschlüsselt bleiben müssen, nur den `SecureString`-Parametertyp.
> Weitere Informationen finden Sie unter Was ist ein Parameter? und Einschränken des Zugriffs auf Systems Manager-Parameter mithilfe von IAM-Richtlinien.

Inhalt in S3-Buckets

Als Teil Ihrer Systems Manager-Vorgänge können Sie Daten in einen oder mehrere Amazon Simple Storage Service (Amazon S3)-Buckets hochladen oder speichern.

Informationen zur Verschlüsselung von S3-Buckets finden Sie unter Daten durch Verschlüsselung schützen und Datenschutz in Amazon S3 im Benutzerhandbuch zu Amazon Simple Storage Service.

Die folgenden Datentypen können Sie als Teil Ihrer Systems Manager-Aktivitäten hochladen oder in S3 Buckets speichern lassen:

- Die Ausgabe von Befehlen inRun Command, eine Fähigkeit von AWS Systems Manager
- Pakete inDistributor, eine Fähigkeit von AWS Systems Manager
- Der Patchvorgang meldet sich anPatch Manager, eine Fähigkeit von AWS Systems Manager
- Patch Manager Patch-Überschreibungslisten
- Skripts oder Ansible Playbooks zur Ausführung in einem Runbook-Workflow in Automation, eine Fähigkeit von AWS Systems Manager
- Chef InSpecProfile zur Verwendung mit Scans in Compliance, eine Funktion von AWS Systems Manager
- AWS CloudTrail Logs
- Der Sitzungsverlauf meldet sich anSession Manager, eine Fähigkeit von AWS Systems Manager
- Berichte vonExplorer, eine Fähigkeit von AWS Systems Manager
- OpsData vonOpsCenter, eine Fähigkeit von AWS Systems Manager
- AWS CloudFormation Vorlagen zur Verwendung mit Automatisierungs-Workflows
- Compliance-Daten aus einem Resource Data Sync-Scan
- Ausgabe von Anfragen zum Erstellen oder Bearbeiten von Verknüpfungen in State Manager verwalteten Knoten AWS Systems Manager, mit einer Fähigkeit von
- Benutzerdefinierte Systems Manager-Dokumente (SSM-Dokumente), die Sie mit dem AWS -verwalteten SSM-Dokument `AWS-RunDocument` ausführen können

CloudWatch Protokolliert Protokollgruppen

Im Rahmen Ihres Systems Manager Betriebs können Sie sich dafür entscheiden, Daten in eine oder mehrere Amazon CloudWatch Logs-Protokollgruppen zu streamen.

Informationen zur Verschlüsselung von CloudWatch Logs-Protokollgruppen finden Sie unter Verschlüsseln von Protokolldaten in CloudWatch Logs using AWS Key Management Service im Amazon CloudWatch Logs-Benutzerhandbuch.

Die folgenden Datentypen haben Sie möglicherweise im Rahmen Ihrer Systems Manager Aktivitäten in eine CloudWatch Logs-Protokollgruppe gestreamt:

- Die Ausgabe der Run Command-Befehle
- Ausgabe von Skripten, die mit der `aws:executeScript`-Aktion in einem Automation-Runbooks ausgeführt werden
- Session Manager-Sitzungsverlaufsprotokolle

- Protokolle vom SSM Agent auf Ihren verwalteten Nodes

Verschlüsselung während der Übertragung

Wir empfehlen, dass Sie ein Verschlüsselungsprotokoll wie Transport Layer Security (TLS) verwenden, um sensible Daten bei der Übertragung zwischen den Clients und Ihren Knoten zu verschlüsseln.

Systems Manager bietet die folgende Unterstützung für die Verschlüsselung Ihrer Daten während der Übertragung.

Verbindungen zu Systems Manager API-Endpunkten

Systems Manager-API-Endpunkte unterstützen ausschließlich sichere Verbindungen über HTTPS. Wenn Sie Systems Manager Ressourcen mit dem AWS Management Console AWS SDK oder der Systems Manager API verwalten, wird die gesamte Kommunikation mit Transport Layer Security (TLS) verschlüsselt. Eine vollständige Liste der API-Endpunkte finden Sie unter AWS-Service -Endpunkte im Allgemeine Amazon Web Services-Referenz.

Verwaltete Instances

AWS bietet sichere und private Konnektivität zwischen Amazon Elastic Compute Cloud (Amazon EC2) -Instances. Darüber hinaus wird Datenverkehr zwischen unterstützen Instances in einer Virtual Private Cloud (VPC) oder in per Peering verbundenen VPCs automatisch mithilfe von AEAD-Algorithmen mit 256-Bit-Verschlüsselung verschlüsselt. Das Verschlüsselungsfeature verwendet die Offload-Möglichkeiten der zugrunde liegenden Hardware ohne Auswirkungen auf die Netzwerkleistung. Unterstützte Instances: C5n, G4, I3en, M5dn, M5n, P3dn, R5dn und R5n.

Session Manager-Sitzungen

Standardmäßig verwendet Session Manager TLS 1.2 zum Verschlüsseln von Sitzungsdaten, die zwischen lokalen Computern von Benutzern in Ihrem Konto und Ihren EC2-Instances übertragen werden. Sie können sich auch dafür entscheiden, die Daten während der Übertragung mithilfe eines AWS KMS key , das in erstellt wurde, weiter zu verschlüsseln. AWS KMS AWS KMS Verschlüsselung ist für die `NonInteractiveCommands` Sitzungstypen `Standard_StreamInteractiveCommands`, und verfügbar.

Run Command-Zugriff

Standardmäßig wird der Remote-Zugriff auf Ihre Knoten über Run Command mit TLS 1.2 verschlüsselt und Anfragen zum Verbindungsaufbau werden mit SigV4 signiert.

Richtlinie für den Datenverkehr zwischen Netzwerken

Sie können Amazon Virtual Private Cloud (Amazon VPC) verwenden, um Grenzen zwischen Ressourcen in Ihren verwalteten Knoten zu erstellen und den Datenverkehr zwischen ihnen, Ihrem On-Premises-Netzwerk und dem Internet zu steuern. Einzelheiten finden Sie unter Verbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

Weitere Informationen zur Sicherheit der Amazon Virtual Private Cloud finden Sie unter Datenschutz des Internet-Datenverkehrs in Amazon VPC im Benutzerhandbuch Amazon VPC.

Identity and Access Management für AWS Systems Manager

AWS Identity and Access Management (IAM) ist ein AWS-Service, mit dem Administratoren den Zugriff auf AWS-Ressourcen sicher steuern können. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (Berechtigungen besitzt) ist, um Systems Manager Ressourcen zu nutzen. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- Funktionsweise von AWS Systems Manager mit IAM
- AWS Systems ManagerBeispiele für identitätsbasierte -Richtlinien
- AWS verwaltete Richtlinien für AWS Systems Manager
- Fehlerbehebung für AWS Systems Manager-Identität und -Zugriff

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, unterscheidet sich je nach Ihrer Arbeit in Systems Manager.

Service-Benutzer: Wenn Sie den Systems Manager-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie für Ihre Arbeit weitere Systems Manager-Funktionen ausführen, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle

nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Unter Fehlerbehebung für AWS Systems Manager-Identität und -Zugriff finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Funktion in Systems Manager haben.

Service-Administrator: Wenn Sie in Ihrem Unternehmen für Systems Manager-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollständigen Zugriff auf Systems Manager. Es ist Ihre Aufgabe, zu bestimmen, auf welche Systems Manager-Funktionen und Ressourcen Ihre Service-Benutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit Systems Manager verwenden kann, finden Sie unter Funktionsweise von AWS Systems Manager mit IAM.

IAM-Administrator: Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Systems Manager verfassen können. Beispiele für identitätsbasierte Systems Manager-Richtlinien, die Sie in IAM verwenden können, finden Sie unter AWS Systems ManagerBeispiele für identitätsbasierte -Richtlinien.

Authentifizierung mit Identitäten

Authentifizierung ist die Art, wie Sie sich mit Ihren Anmeldeinformationen bei AWS anmelden. Die Authentifizierung (Anmeldung bei AWS) muss als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle erfolgen.

Sie können sich bei AWS als Verbundidentität mit Anmeldeinformationen anmelden, die über eine Identitätsquelle bereitgestellt werden. Benutzer von AWS IAM Identity Center (IAM Identity Center), die Single-Sign-on-Authentifizierung Ihres Unternehmens und Anmeldeinformationen für Google oder Facebook sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie auf AWS mithilfe des Verbunds zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der AWS Management Console oder beim AWS-Zugriffsportal anmelden. Weitere Informationen zum Anmelden bei AWS finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch von AWS-Anmeldung.

Bei programmgesteuertem Zugriff auf AWS bietet AWS ein Software Development Kit (SDK) und eine Command Line Interface (CLI, Befehlszeilenschnittstelle) zum kryptographischen Signieren Ihrer Anfragen mit Ihren Anmeldeinformationen. Wenn Sie keine AWS-Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen

Methode zum eigenen Signieren von Anforderungen finden Sie unter Signieren von AWS-API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise die Verwendung von Multi-Faktor Authentifizierung (MFA), um die Sicherheit Ihres Kontos zu verbessern. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center-Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto-Root-Benutzer

Wenn Sie ein AWS-Konto neu erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen des Kontos hat. Diese Identität wird als AWS-Konto-Root-Benutzer bezeichnet. Für den Zugriff auf den Root-Benutzer müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die zur Erstellung des Kontos verwendet wurden. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder eine einzelne Anwendung. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität in Ihrem AWS-Konto mit spezifischen Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der AWS Management Console übernehmen, indem Sie Rollen wechseln. Sie können eine Rolle annehmen, indem Sie eine AWS CLI oder AWS-API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff: Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center-Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen: Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. In einigen AWS-Services können Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

- Serviceübergreifender Zugriff: Einige AWS-Services verwenden Features in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward access sessions (FAS) – Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in AWS verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anfragen werden nur dann gestellt, wenn ein Service eine Anfrage erhält, die eine Interaktion mit anderen AWS-Services oder -Ressourcen erfordert. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle: Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
  - Serviceverknüpfte Rolle: Eine serviceverknüpfte Rolle ist ein Typ von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene Rollen anzeigen, aber nicht bearbeiten.
- Anwendungen in Amazon EC2: Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und AWS CLI- oder AWS-API-Anforderungen durchführen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle einer EC2-Instance zuzuweisen und die Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Für die Zugriffssteuerung in AWS erstellen Sie Richtlinien und weisen diese den AWS-Identitäten oder -Ressourcen zu. Eine Richtlinie ist ein Objekt in AWS, das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Benutzerinformationen über die AWS Management Console, die AWS CLI oder die AWS -API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto anfügen können.

Verwaltete Richtlinien umfassen von AWS verwaltete und von Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Weitere Informationen über von AWS verwaltete Richtlinien für Systems Manager finden Sie unter Von AWS Systems Manager-verwaltete Richtlinien.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können verwaltete AWS-Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3, AWS WAF und Amazon VPC sind Beispiele für Dienste, die ACLs unterstützen. Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen:Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-

Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

- Service-Kontrollrichtlinien (SCPs) – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OE) in AWS Organizations angeben. AWS Organizations ist ein Dienst für die Gruppierung und zentrale Verwaltung mehrerer AWS-Konten Ihres Unternehmens. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. SCPs schränken Berechtigungen für Entitäten in Mitgliedskonten einschließlich des jeweiligen Root-Benutzer des AWS-Kontos ein. Weitere Informationen zu Organizations und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations-Benutzerhandbuch.
- Sitzungsrichtlinien:Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen dazu, wie AWS die Zulässigkeit einer Anforderung ermittelt, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter Logik für die Richtlinienauswertung im IAM-Benutzerhandbuch.

Funktionsweise von AWS Systems Manager mit IAM

Bevor Sie AWS Identity and Access Management (IAM) zur Verwaltung des Zugriffs auf verwenden AWS Systems Manager, sollten Sie wissen, mit welchen IAM-Funktionen Sie verwenden können. Systems Manager Einen allgemeinen Überblick über die Funktionsweise von IAM Systems Manager und andere AWS-Services Funktionen finden Sie im AWS-Services IAM-Benutzerhandbuch unter Funktionen mit IAM.

Themen

- Identitätsbasierte Systems Manager-Richtlinien
- Ressourcenbasierte Systems Manager-Richtlinien
- Autorisierung auf der Basis von Systems Manager-Tags
- Systems ManagerIAM-Rollen

Identitätsbasierte Systems Manager-Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie festlegen, welche Aktionen und Ressourcen gewährt oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Systems Manager unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Aktionen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Richtlinienaktionen in Systems Manager verwenden das folgende Präfix vor der Aktion: `ssm:`. Um jemandem beispielsweise die Berechtigung zu erteilen, einen Systems Manager-Parameter (SSM-Parameter) mit der Systems Manager `PutParameter` API-Operation zu erstellen, nehmen Sie die Aktion `ssm:PutParameter` in seine Richtlinie auf. Richtlinienanweisungen müssen ein `Action`- oder `NotAction`-Element enthalten. Systems Manager definiert seinen eigenen Satz an Aktionen, die Aufgaben beschreiben, die Sie mit diesem Service durchführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

```
"Action": [
      "ssm:action1",
      "ssm:action2"
```

> **Note**
>
> Die folgenden Funktionen ermöglichen die AWS Systems Manager Verwendung verschiedener Präfixe vor Aktionen.
>
> - AWS AppConfig verwendet das Präfix `appconfig:` vor Aktionen.
> - Incident Manager verwendet das Präfix `ssm-incidents:` oder `ssm-contacts:` vor Aktionen.
> - Systems Manager GUI Connect verwendet das Präfix `ssm-guiconnect` vor Aktionen.

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "ssm:Describe*"
```

Eine Liste der Systems Manager-Aktionen finden Sie unter Von AWS Systems Manager definierte Aktionen in der Service-Autorisierungs-Referenz.

Ressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Die Ressource des Systems Manager-Wartungsfensters hat beispielsweise das folgende ARN-Format.

```
arn:aws:ssm:region:account-id:maintenancewindow/window-id
```

Um die mw-0c50858d01EXAMPLE-Wartungsfenster in Ihrer Anweisung in der Region USA Ost (Ohio) anzugeben, verwenden Sie einen ARN ähnlich dem folgenden.

```
"Resource": "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0c50858d01EXAMPLE"
```

Um alle Wartungsfenster anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*).

```
"Resource": "arn:aws:ssm:region:123456789012:maintenancewindow/*"
```

Bei `Parameter Store` API-Vorgängen können Sie den Zugriff auf alle Parameter auf einer Hierarchieebene bereitstellen oder einschränken, indem Sie hierarchische Namen und AWS Identity and Access Management (IAM-) Richtlinien wie folgt verwenden.

```
"Resource": "arn:aws:ssm:region:123456789012:parameter/Dev/ERP/Oracle/*"
```

Einige Systems Manager-Aktionen, z. B. zum Erstellen von Ressourcen, können auf bestimmten Ressourcen nicht ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

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

Manche Systems Manager-API-Operationen akzeptieren mehrere Ressourcen. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie ihre ARNs mit Kommas wie folgt.

```
"Resource": [
      "resource1",
      "resource2"
```

> **Note**
>
> Die meisten AWS-Services behandeln einen Doppelpunkt (:) oder einen Schrägstrich (/) als dasselbe Zeichen in ARNs. Allerdings erfordert Systems Manager eine exakte Übereinstimmung in den Ressourcenmustern und -regeln. Verwenden Sie also die richtigen ARN-Zeichen zum Erstellen von Ereignismustern, sodass sie mit dem ARN der Ressource übereinstimmen.

In der folgenden Tabelle werden die ARN-Formate für die Ressourcentypen beschrieben, die von unterstützt werdenSystems Manager.

> **Note**
>
> Beachten Sie die folgenden Ausnahmen für ARN-Formate.
>
> - Die folgenden Funktionen ermöglichen die AWS Systems Manager Verwendung verschiedener Präfixe vor Aktionen.
>   - AWS AppConfig verwendet das Präfix `appconfig:` vor Aktionen.
>   - Incident Manager verwendet das Präfix `ssm-incidents:` oder `ssm-contacts:` vor Aktionen.
>   - Systems Manager GUI Connect verwendet das Präfix `ssm-guiconnect` vor Aktionen.
> - Dokumente und Automatisierungsdefinitionsressourcen, die Amazon gehören, sowie öffentliche Parameter, die sowohl von Amazon als auch von Drittanbietern bereitgestellt werden, enthalten keine Konto-IDs in ihren ARN-Formaten. Beispielsweise:
>   - Das SSM-Dokument`AWS-RunPatchBaseline`:
>
>   `arn:aws:ssm:us-east-2:::document/AWS-RunPatchBaseline`
>   - Das `AWS-ConfigureMaintenanceWindows` Runbook für die Automatisierung:
>
>   `arn:aws:ssm:us-east-2:::automation-definition/AWS-ConfigureMaintenanceWindows`
>   - Der öffentliche Parameter`/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`

Weitere Informationen zu diesen drei Ressourcentypen finden Sie in den folgenden Themen:

- Arbeiten mit Dokumenten
- Ausführen von Automatisierungen
- Arbeiten mit öffentlichen Parametern

Ressourcentyp	ARN-Format
Anwendung (AWS AppConfig)	arn:aws:appconfig:*region*:*account-id* :application/*application-id*
Zuordnung	arn:aws:ssm:*region*:*account-id* :association/*association-id*
Automatisierungsausführung	arn:aws:ssm:*region*:*account-id* :automation-execution/*automation-execution-id*
Automatisierungsdefinition (mit Versions-Subressource)	arn:aws:ssm:*region*:*account-id* :automation-definition/*automation-definition-id* :*version-id* [1]
Konfigurationsprofil (AWS AppConfig)	arn:aws:appconfig:*region*:*account-id* :application/*application-id* /configurationprofile/*configurationprofile-id*
Kontakt (Incident Manager)	arn:aws:ssm-contacts:*region*:*account-id* :contact/*contact-alias*
Bereitstellungsstrategie (AWS AppConfig)	arn:aws:appconfig:*region*:*account-id* :deploymentstrategy/*deploymentstrategy-id*
Dokument	arn:aws:ssm:*region*:*account-id* :document/*document-name*
Umgebung (AWS AppConfig)	arn:aws:appconfig:*region*:*account-id* :application/*application-id* /environment/*environment-id*
Vorfall	arn:aws:ssm-incidents:*region*:*account-id* :incident-record/*response-plan-name* /*incident-id*

Ressourcentyp	ARN-Format
Wartungsfenster	arn:aws:ssm:*region*:*account-id* :maintenancewindow/*window-id*
Verwalteter Knoten	arn:aws:ssm:*region*:*account-id* :managed-instance/*managed-node-id*
Bestand an verwalteten Knoten	arn:aws:ssm:*region*:*account-id* :managed-instance-inventory /*managed-node-id*
OpsItem	*arn:aws:ssm: region: account-id:opsitem/ -id OpsItem*
Parameter	Ein Parameter mit einer Ebene: • arn:aws:ssm:*region*:*account-id* :parameter/*parameter-name*/ Ein Parameter, der mit einer hierarchischen Struktur benannt ist: • arn:aws:ssm:*region*:*account-id* :parameter/*parameter-name-root* /*level-2*/*level-3*/*level-4*/*level-5* [2]
Patch-Baseline	arn:aws:ssm:*region*:*account-id* :patchbaseline/*patch-baseline-id*
Response-Plan	arn:aws:ssm-incidents:*region*:*account-id* :response-plan/*response-plan-name*
Sitzung	arn:aws:ssm:*region*:*account-id* :session/*session-id* [3]
Alle Systems Manager-Ressourcen	arn:aws:ssm:*

Ressourcentyp	ARN-Format
Alle Ressourcen gehören Systems Manager den in der angegebenen Liste angegebenen AWS-Konto AWS-Region	arn:aws:ssm:*region*:*account-id* :*

1

Für Automatisierungsdefinitionen unterstützt Systems Manager eine Second-Level-Ressource (version-ID). In AWS werden diese Ressourcen der zweiten Ebene als Unterressourcen bezeichnet. Wenn Sie eine Versions-Subressource für eine Automatisierungsdefinition-Ressource angeben, können Sie Zugriff auf bestimmte Versionen einer Automatisierungsdefinition erteilen. So können Sie beispielsweise sicherstellen, dass nur die neueste Version einer Automatisierungsdefinition in der Verwaltung Ihrer Knoten verwendet wird.

2

Um Parameter zu organisieren und zu verwalten, können Sie Namen für Parameter mit hierarchischem Aufbau erstellen. Bei dem hierarchischen Aufbau kann ein Parametername einen Pfad enthalten, den Sie mit Schrägstrichen definieren. Der Name einer Parameterressource darf maximal fünfzehn Ebenen umfassen. Wir empfehlen, dass Sie Hierarchien erstellen, die eine vorhandene hierarchische Struktur in Ihrer Umgebung abbilden. Weitere Informationen finden Sie unter Erstellen von Systems Manager-Parametern.

3

In den meisten Fällen wird die Sitzungs-ID aus der ID des Kontobenutzers, der die Sitzung gestartet hat, und einem alphanumerischen Suffix aufgebaut. Zum Beispiel:

```
arn:aws:us-east-2:111122223333:session/JohnDoe-1a2b3c4sEXAMPLE
```

Wenn die Benutzer-ID jedoch nicht verfügbar ist, wird der ARN stattdessen auf diese Weise erstellt:

```
arn:aws:us-east-2:111122223333:session/session-1a2b3c4sEXAMPLE
```

Weitere Informationen zum Format von ARNs finden Sie unter Amazon-Ressourcennamen (ARNs) im Allgemeine Amazon Web Services-Referenz.

Eine Liste der Systems Manager-Ressourcentypen und ihrer ARNs finden Sie unter Von AWS Systems Managerdefinierte Ressourcen in der Service-Autorisierungs-Referenz. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von AWS Systems Manager definierte Aktionen.

Bedingungsschlüssel für Systems Manager

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste von Systems Manager-Bedingungsschlüsseln finden Sie unter Bedingungsschlüssel für AWS Systems Manager in der Service-Autorisierungs-Referenz. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von AWS Systems Manager definierte Aktionen.

Weitere Informationen zum Verwenden des `ssm:resourceTag/*`-Bedingungsschlüssels finden Sie in den folgenden Themen:

- Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent
- Den Zugriff von Run Command anhand von Tags beschränken
- Beschränkung des Sitzungszugriffs auf Instance-Tags

Weitere Informationen zum Verwenden der Bedingungsschlüssels `ssm:Recursive` und `ssm:Overwrite` finden Sie unter Arbeiten mit Parameterhierarchien.

Beispiele

Beispiele für identitätsbasierte Systems Manager-Richtlinien finden Sie unter AWS Systems ManagerBeispiele für identitätsbasierte -Richtlinien.

Ressourcenbasierte Systems Manager-Richtlinien

Andere AWS-Services, wie Amazon Simple Storage Service (Amazon S3), unterstützen ressourcenbasierte Berechtigungsrichtlinien. Beispielsweise können Sie einem S3-Bucket eine Berechtigungsrichtlinie zuweisen, um die Zugriffsberechtigungen für diesen Bucket zu verwalten.

Systems Manager unterstützt keine ressourcenbasierten Richtlinien.

Autorisierung auf der Basis von Systems Manager-Tags

Sie können Tags an Systems Manager-Ressourcen anfügen oder Tags in einer Anforderung an Systems Manager übergeben. Um den Zugriff basierend auf Tags zu steuern, stellen Sie Tag-Informationen im Bedingungselement einer Richtlinie unter Verwendung der Bedingungsschlüssel `ssm:resourceTag/`*`key-name`*, `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* oder `aws:TagKeys` zur Verfügung. Sie können den folgenden Ressourcentypen beim Erstellen oder Aktualisieren Tags hinzufügen:

- Dokument
- Verwalteter Knoten
- Wartungsfenster
- Parameter
- Patch-Baseline
- OpsItem

Weitere Informationen zur Markierung von Systems Manager-Ressourcen finden Sie unter Markieren von Systems Manager-Ressourcen.

Ein Beispiel für eine identitätsbasierte Richtlinie zur Einschränkung des Zugriffs auf eine Ressource auf der Grundlage der Markierungen dieser Ressource finden Sie unter Anzeigen von Systems Manager-Dokumenten basierend auf Tags.

Systems ManagerIAM-Rollen

Eine IAM-Rolle ist eine Entität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt.

Verwenden temporärer Anmeldeinformationen mit Systems Manager

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie API-Operationen AWS Security Token Service (AWS STS) wie AssumeRoleoder GetFederationToken aufrufen.

Systems Manager unterstützt die Verwendung temporärer Anmeldeinformationen.

Service-verknüpfte Rollen

Mit Diensten verknüpfte Rollen ermöglichen AWS-Services den Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-Konto aufgelistet und gehören zum Service. Ein -Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Systems Manager unterstützt serviceverknüpfte Rollen. Details zum Erstellen oder Verwalten von serviceverknüpften Systems Manager-Rollen finden Sie unter Verwenden von serviceverknüpften Rollen für Systems Manager.

Servicerollen

Dieses Feature ermöglicht einem Service das Annehmen einer Servicerolle in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Servicerollen werden in Ihrem IAM-Konto angezeigt und gehören zum Konto. Dies bedeutet, dass ein -Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktion des Services beeinträchtigen.

Systems Manager unterstützt Servicerollen.

Auswählen einer IAM-Rolle in Systems Manager

Damit Systems Manager mit Ihren verwalteten Knoten interagieren kann, müssen Sie eine Rolle wählen, die Systems Manager den Zugriff auf Knoten in Ihrem Namen erlaubt. Wenn Sie zuvor eine Servicerolle oder serviceverknüpfte Rolle erstellt haben, stellt Ihnen Systems Manager eine Liste mit Rollen bereit, aus denen Sie wählen können. Es ist wichtig, eine Rolle zu wählen, die den Zugriff auf das Starten und Stoppen von verwalteten Knoten erlaubt.

Um auf EC2-Instances zuzugreifen, müssen Sie Instance-Berechtigungen konfigurieren. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

Für den Zugriff auf Nicht-EC2-Knoten in einer Hybrid- und Multi-Cloud benötigt Ihr AWS-Konto eine IAM-Servicerolle. Weitere Informationen finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Servicerolle in Hybrid- und Multicloud-Umgebungen.

Ein Automation-Workflow kann im Kontext einer Service-Rolle initiiert werden (oder eine Rolle übernehmen). Auf diese Weise kann der Service Aktionen in Ihrem Namen ausführen. Wenn Sie keine Übernahmerolle angeben, verwendet Automation den Kontext des Benutzers, der die Ausführung aufgerufen hat. In bestimmten Situationen ist es jedoch erforderlich, dass Sie eine Service-Rolle für Automation angeben. Weitere Informationen finden Sie unter Konfigurieren eines Service-Rollenzugriffs (Rolle übernehmen) für Automatisierungen.

Von AWS Systems Manager-verwaltete Richtlinien

AWS adressiert viele gängige Anwendungsfälle durch die Bereitstellung eigenständiger IAM-Richtlinien, die von erstellt und verwaltet AWS werden. Diese von AWS verwalteten Richtlinien erteilen die erforderlichen Berechtigungen für häufige Anwendungsfälle, sodass Sie nicht mühsam ermitteln müssen, welche Berechtigungen erforderlich sind. (Sie können auch Ihre eigenen, benutzerdefinierten IAM-Richtlinien erstellen, um Berechtigungen für Systems Manager-Aktionen und -Ressourcen zu gewähren.)

Weitere Informationen zu verwalteten Richtlinien für Systems Manager finden Sie unter AWS verwaltete Richtlinien für AWS Systems Manager

Allgemeine Informationen zu verwalteten Richtlinien finden Sie im IAM-Benutzerhandbuch unter AWS Verwaltete Richtlinien.

AWS Systems ManagerBeispiele für identitätsbasierte -Richtlinien

Standardmässig verfügen AWS Identity and Access Management (IAM)-Entitäten über keine Berechtigungen zum Erstellen oder Ändern von AWS Systems Manager-Ressourcen. Sie können auch keine Aufgaben unter Verwendung der Systems Manager-Konsole, AWS Command Line Interface, (AWS CLI) oder AWS-API ausführen. Ein Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den -Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Das folgende Beispiel zeigt eine Berechtigungsrichtlinie, die es einem Benutzer erlaubt, Dokumente in USA Ost (Ohio) (us-east-2) AWS-Region, deren Name mit **`MyDocument-`** beginnt, zu löschen.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "ssm:DeleteDocument"
      ],
      "Resource" : [
        "arn:aws:ssm:us-east-2:111122223333:document/MyDocument-*"
      ]
    }
  ]
}
```

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der Systems Manager-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer
- Vermeidung des Problems des verwirrten Stellvertreters (dienstübergreifend)
- Beispiele für vom Kunden verwaltete Richtlinien

- Anzeigen von Systems Manager-Dokumenten basierend auf Tags

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Systems Manager-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie AWS-kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS-verwaltete Richtlinien oder AWS-verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS-Service, wie beispielsweise AWS CloudFormation, verwendet werden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als

100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.

- Bedarf einer Multi-Faktor-Authentifizierung (MFA) – Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Systems Manager-Konsole

Um auf die Systems Manager-Konsole zuzugreifen, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details über die Systems Manager-Ressourcen und andere Ressourcen in Ihrem AWS-Konto aufzulisten und anzuzeigen.

Um Systems Manager in der Systems Manager-Konsole vollständig nutzen zu können, müssen Sie über die Berechtigungen der folgenden Services verfügen:

- AWS Systems Manager
- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Identity and Access Management (IAM)

Sie können die erforderlichen Berechtigungen mit der folgenden Richtlinienanweisung erteilen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:*",
                "ec2:describeInstances",
                "iam:ListRoles"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                }
            }
        }
    ]
}
```

Wenn Sie eine identitätsbasierte Richtlinie erstellen, die restriktiver ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole für IAM-Entitäten (Benutzer oder Rollen) mit dieser Richtlinie nicht wie vorgesehen.

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die den API-Operation entsprechen, die Sie ausführen möchten.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

```
{
    "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": "*"
        }
    ]
}
```

Vermeidung des Problems des verwirrten Stellvertreters (dienstübergreifend)

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine Entität, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine Entität mit größeren Rechten zwingen kann, die Aktion auszuführen. In AWS kann der dienstübergreifende Identitätswechsel zu Confused-Deputy-Problem führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen die Verwendung der globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen, die AWS Systems Manager einem anderen Service erteilt, auf eine bestimmte Ressource zu beschränken. Wenn der `aws:SourceArn`-Wert nicht die Konto-ID enthält, z. B. den Amazon-Ressourcenname (ARN) eines S3-Buckets, müssen Sie beide globale Bedingungskontext-Schlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der

gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

In den folgenden Abschnitten finden Sie Beispielrichtlinien für AWS Systems Manager-Funktionen.

Beispiel für hybride Aktivierungsrichtlinien

Bei Servicerollen, die bei einer Hybrid-Aktivierung verwendet werden, muss der Wert von `aws:SourceArn` der ARN des AWS-Konto sein. Geben Sie die AWS-Region im ARN an, mit dem Sie Ihre Hybrid-Aktivierung erstellt haben. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Bedingungskontext-Schlüssel `aws:SourceArn` mit Platzhaltern (*) für die unbekannten Teile des ARN. Zum Beispiel `arn:aws:ssm:*:region:123456789012:*`.

Das folgende Beispiel veranschaulicht die Verwendung der globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` für Automatisierung, um das Confused-Deputy-Problem in der Region USA Ost (Ohio) (us-east-2) zu verhindern.

```
{
   "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:*"
            }
         }
      }
   ]
}
```

Beispiel-Richtlinie für Ressourcen-Datensynchronisierung

Systems Manager Bestand, Explorer und Compliance ermöglichen es Ihnen, eine Ressourcen-Datensynchronisierung zu erstellen, um die Speicherung Ihrer Betriebsdaten (OpsData) in einem zentralen Amazon-Simple-Storage-Service-Bucket zu zentralisieren. Wenn Sie eine Ressourcen-Datensynchronisierung mit AWS Key Management Service (AWS KMS) verschlüsseln möchten, müssen Sie entweder einen neuen Schlüssel erstellen, in dem die folgende Richtlinie enthalten ist, oder Sie aktualisieren einen vorhandenen Schlüssel und fügen ihm diese Richtlinie hinzu. Die `aws:SourceArn` und `aws:SourceAccount`-Bedingungsschlüssel in dieser Richtlinie verhindern das Confused-Deputy-Problem. Hier ist eine Beispielrichtlinie.

```
{
    "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**
>
> Der ARN in der Beispielrichtlinie ermöglicht es dem System, OpsData aus allen Quellen außer AWS Security Hub zu verschlüsseln. Wenn Sie Security-Hub-Daten verschlüsseln müssen, zum Beispiel wenn SieExplorer verwenden, um Security-Hub-Daten zu sammeln, müssen Sie eine zusätzliche Richtlinie anfügen, die den folgenden ARN angibt: `"aws:SourceArn": "arn:aws:ssm:*:account-id:role/aws-service-role/opsdatasync.ssm.amazonaws.com/AWSServiceRoleForSystemsManagerOpsDataSync"`

Beispiele für vom Kunden verwaltete Richtlinien

Sie können eigenständige Richtlinien erstellen, die Sie in Ihrem eigenen AWS-Konto verwalten. Wir bezeichnen diese als vom Kunden verwaltete Richtlinien. Sie können diese Richtlinien an mehrere Prinzipal-Entitäten in Ihrem AWS-Konto anfügen. Wenn Sie eine Richtlinie an eine Auftraggeber-Entität anfügen, gewähren Sie ihr die in der Richtlinie festgelegten Berechtigungen. Weitere Informationen finden Sie unter Beispiele für kundenverwaltete Richtlinien im IAM-Benutzerhandbuch.

Die folgenden Beispiele für Benutzerrichtlinien gewähren Berechtigungen für verschiedene Aktionen von Systems Manager. Verwenden Sie diese, um den Systems Manager-Zugriff für Ihre IAM-Entitäten (Benutzer und Rollen) einzuschränken. Diese Richtlinien funktionieren bei der Ausführung von Aktionen in der Systems Manager-API, den AWS-SDKs oder der AWS CLI. Für Benutzer, die die Konsole verwenden, müssen Sie zusätzliche konsolenspezifische Berechtigungen erteilen. Weitere Informationen finden Sie unter Verwenden der Systems Manager-Konsole.

> **Note**
>
> In allen Beispielen werden die Region USA West (Oregon) (us-west-2) und fiktive Konto-IDs verwendet. Die Konto-ID sollte nicht im Amazon-Ressourcennamen (ARN) für öffentliche AWS-Dokumente (Dokumente, die mit `AWS-*` anfangen) enthalten.

Beispiele

- Beispiel 1: Zulassen, dass ein Benutzer Systems Manager-Operationen in einer einzelnen Region ausführt
- Beispiel 2: Zulassen, dass ein Benutzer Dokumente für eine einzelne Region auflistet

Beispiel 1: Zulassen, dass ein Benutzer Systems Manager-Operationen in einer einzelnen Region ausführt

Im folgenden Beispiel werden die Berechtigungen zur Ausführung von Systems Manager-Operationen nur in der Region USA Ost (Ohio) (us-east-2) gewährt.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "ssm:*"
      ],
      "Resource" : [
        "arn:aws:ssm:us-east-2:aws-account-ID:*"
      ]
    }
  ]
}
```

Beispiel 2: Zulassen, dass ein Benutzer Dokumente für eine einzelne Region auflistet

Das folgende Beispiel erteilt Berechtigungen zum Auflisten aller Dokumentennamen, die mit **`Update`** in der Region USA Ost (Ohio) (us-east-2) beginnen.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "ssm:ListDocuments"
      ],
      "Resource" : [
        "arn:aws:ssm:us-east-2:aws-account-ID:document/Update*"
      ]
    }
  ]
}
```

Beispiel 3: Erlauben, dass ein Benutzer ein spezifisches SSM-Dokument zum Ausführen von Befehlen auf bestimmten Knoten verwendet

Die folgende IAM-Beispielrichtlinie ermöglicht es einem Benutzer, in der Region USA Ost (Ohio) (us-east-2) die folgenden Aktionen auszuführen:

- Listen Sie Systems Manager-Dokumente (SSM-Dokumente) und Dokumentversionen auf.
- Zeigen Sie Details zu Dokumenten an.
- Senden Sie einen Befehl mit dem in der Richtlinie angegebenen Dokument. Der Name des Dokuments wird durch den folgenden Eintrag bestimmt.

```
arn:aws:ssm:us-east-2:aws-account-ID:document/Systems-Manager-document-name
```

- Senden Sie einen Befehl an drei Knoten. Die Knoten werden anhand der folgenden Einträge im zweiten `Resource`-Abschnitt bestimmt.

```
"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"
```

- Zeigen Sie Details zu einem Befehl an, nachdem er gesendet wurde.
- Starten und Beenden von Workflows in Automation, eine Funktion von AWS Systems Manager.
- Informationen zum Automation-Workflows

Wenn Sie einem Benutzer die Berechtigung gewähren möchten, dieses Dokument zu verwenden, um Befehle an jeden Knoten zu senden, auf den der Benutzer Zugriff hat, können Sie einen Eintrag ähnlich dem folgenden im `Resource`-Abschnitt angeben und die anderen Knoteneinträge entfernen. Im folgenden Beispiel wird die Region USA Ost (Ohio) (us-east-2) verwendet.

```
"arn:aws:ec2:us-east-2:*:instance/*"
```

```
{
    "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: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",

                "arn:aws:ssm:us-east-2:aws-account-ID: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:aws-account-ID:automation-definition/*"
            ]
        },
        {
            "Action": "ssm:DescribeAutomationExecutions",
            "Effect": "Allow",
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "ssm:StopAutomationExecution",
                "ssm:GetAutomationExecution"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Anzeigen von Systems Manager-Dokumenten basierend auf Tags

Sie können in Ihrer identitätsbasierten Richtlinie Bedingungen für die Steuerung des Zugriffs auf Systems Manager-Ressourcen auf der Basis von Tags verwenden. Dieses Beispiel zeigt, wie Sie eine Richtlinie erstellen können, die die Anzeige eines SSM-Dokuments ermöglicht. Die Berechtigung wird jedoch nur gewährt, wenn das Dokument-Tag `Owner` den Wert des Benutzernamens dieses Benutzers hat. Diese Richtlinie gewährt auch die Berechtigungen, die für die Ausführung dieser Aktion auf der Konsole erforderlich sind.

```
{
    "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}"}
```

```
            }
        }
    ]
}
```

Sie können diese Richtlinie den -Benutzern in Ihrem Konto zuweisen. Wenn ein Benutzer mit dem Namen `richard-roe` versucht, ein Dokument mit dem Namen Systems Manager anzuzeigen, muss das Dokument mit dem Tag `0wner=richard-roe` oder `owner=richard-roe` gekennzeichnet werden. Andernfalls wird diesen der Zugriff verweigert. Der Tag-Schlüssel `0wner` der Bedingung stimmt sowohl mit `0wner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinien für AWS Systems Manager

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie für alle AWS Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: ServiceRole AmazonSSM-Richtlinie

Sie können nichts an Ihre AWS Identity and Access Management (IAM-) Entitäten anhängen`AmazonSSMServiceRolePolicy`. Diese Richtlinie ist mit einer dienstbezogenen Rolle

verknüpft, die es AWS Systems Manager ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Verwenden von Rollen zum Sammeln und Anzeigen von Inventar OpsData.

`AmazonSSMServiceRolePolicy` ermöglicht es Systems Manager, die folgenden Aktionen auf allen zugehörigen Ressourcen (`"Resource": "*"`) durchzuführen, außer wenn es anders angegeben ist:

- `ssm:CancelCommand`
- `ssm:GetCommandInvocation`
- `ssm:ListCommandInvocations`
- `ssm:ListCommands`
- `ssm:SendCommand`
- `ssm:GetAutomationExecution`
- `ssm:GetParameters`
- `ssm:StartAutomationExecution`
- `ssm:StopAutomationExecution`
- `ssm:ListTagsForResource`
- `ssm:GetCalendarState`
- `ssm:UpdateServiceSetting` [1]
- `ssm:GetServiceSetting` [1]
- `ec2:DescribeInstanceAttribute`
- `ec2:DescribeInstanceStatus`
- `ec2:DescribeInstances`
- `lambda:InvokeFunction` [2]
- `states:DescribeExecution` [3]
- `states:StartExecution` [3]
- `resource-groups:ListGroups`
- `resource-groups:ListGroupResources`
- `resource-groups:GetGroupQuery`
- `tag:GetResources`
- `config:SelectResourceConfig`

- config:DescribeComplianceByConfigRule
- config:DescribeComplianceByResource
- config:DescribeRemediationConfigurations
- config:DescribeConfigurationRecorders
- cloudwatch:DescribeAlarms
- compute-optimizer:GetEC2InstanceRecommendations
- compute-optimizer:GetEnrollmentStatus
- support:DescribeTrustedAdvisorChecks
- support:DescribeTrustedAdvisorCheckSummaries
- support:DescribeTrustedAdvisorCheckResult
- support:DescribeCases
- iam:PassRole [4]
- cloudformation:DescribeStacks
- cloudformation:ListStackResources
- cloudformation:ListStackInstances [5]
- cloudformation:DescribeStackSetOperation [5]
- cloudformation:DeleteStackSet [5]
- cloudformation:DeleteStackInstances [6]
- events:PutRule [7]
- events:PutTargets [7]
- events:RemoveTargets [8]
- events:DeleteRule [8]
- events:DescribeRule
- securityhub:DescribeHub

[1] Die Aktionen ssm:UpdateServiceSetting und ssm:GetServiceSetting sind nur für die folgenden Ressourcen zulässig.

```
arn:aws:ssm:*:*:servicesetting/ssm/opsitem/*
arn:aws:ssm:*:*:servicesetting/ssm/opsdata/*
```

[2] Die Aktion lambda:InvokeFunction ist nur für die folgenden Ressourcen zulässig.

```
arn:aws:lambda:*:*:function:SSM*
arn:aws:lambda:*:*:function:*:SSM*
```

[3] Die Aktionen `states:` sind nur für die folgenden Ressourcen zulässig.

```
arn:aws:states:*:*:stateMachine:SSM*
arn:aws:states:*:*:execution:SSM*
```

[4] Die Aktion `iam:PassRole` ist nur mit der folgenden Bedingung für den Systems Manager-Service zulässig.

```
"Condition": {
   "StringEquals": {
      "iam:PassedToService": [
         "ssm.amazonaws.com"
      ]
   }
}
```

[5] Die Aktionen `cloudformation:ListStackInstances`, `cloudformation:DescribeStackSetOperation`, und `cloudformation:DeleteStackSet` sind nur für die folgenden Ressourcen zulässig.

```
arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-SSM*:*
```

[6] Die Aktion `cloudformation:DeleteStackInstances` ist nur für die folgenden Ressourcen zulässig.

```
arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-SSM*:*
arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-SSM*:*
arn:aws:cloudformation:*:*:type/resource/*
```

[7] Die Aktionen `events:PutRule` und `events:PutTargets` sind nur mit der folgenden Bedingung für den Systems Manager-Service zulässig.

```
"Condition": {
    "StringEquals": {
        "events:ManagedBy": "ssm.amazonaws.com"
    }
```

```
}
```

[8] Die Aktionen `events:RemoveTargets` und `events:DeleteRule` sind nur für die folgenden Ressourcen zulässig.

```
arn:aws:events:*:*:rule/SSMExplorerManagedRule
```

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter ServiceRoleAmazonSSM-Richtlinie im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AmazonSSM Access ReadOnly

Sie können die `AmazonSSMReadOnlyAccess`-Richtlinie an Ihre IAM-Identitäten anfügen. Diese Richtlinie gewährt schreibgeschützten Zugriff auf AWS Systems Manager API-Operationen, einschließlich`Describe*`, und. `Get*` `List*`

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AmazonSSM ReadOnly Access im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSSystemsManagerOpsDataSyncServiceRolePolicy

Sie können `AWSSystemsManagerOpsDataSyncServiceRolePolicy` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist an eine servicegebundene Rolle angehängt, die Systems Manager die Durchführung von Aktionen in Ihrem Namen ermöglicht. Weitere Informationen finden Sie unter Verwenden von Rollen zum Erstellen OpsData und OpsItems für Explorer.

`AWSSystemsManagerOpsDataSyncServiceRolePolicy`ermöglicht der `AWSServiceRoleForSystemsManagerOpsDataSync` serviceverknüpften Rolle das Erstellen und Aktualisieren von Ergebnissen OpsItems sowie das Verwenden OpsData von AWS Security Hub Ergebnissen.

Die Richtlinie erlaubt es Systems Manager, die folgenden Aktionen für alle damit verbundenen Ressourcen (`"Resource": "*"`) auszuführen, außer wenn dies angegeben ist:

- `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] Die Aktionen `ssm:GetOpsItem` und `ssm:UpdateOpsItem` sind nur mit der folgenden Bedingung für den Systems Manager-Service zulässig.

```
"Condition": {
    "StringEquals": {
        "aws:ResourceTag/ExplorerSecurityHubOpsItem": "true"
    }
}
```

[2] Die Aktion `ssm:AddTagsToResource` ist nur für die folgende Ressource zulässig.

```
arn:aws:ssm:*:*:opsitem/*
```

[3] Die Aktionen `ssm:UpdateServiceSetting` und `ssm:GetServiceSetting` sind nur für die folgenden Ressourcen zulässig.

```
arn:aws:ssm:*:*:servicesetting/ssm/opsitem/*
arn:aws:ssm:*:*:servicesetting/ssm/opsdata/*
```

[4] Die `securityhub:BatchUpdateFindings` werden die Berechtigungen durch die folgende Bedingung nur für den Systems Manager-Dienst verweigert.

```
{
   "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
```

```
    }
  }
```

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie AWSSystemsManagerOpsDataSyncServiceRolePolicyim Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: InstanceDefault AmazonSSMManagedEC2-Richtlinie

Sie sollten nur IAM-Rollen für Amazon EC2 EC2-Instances anhängen`AmazonSSMManagedEC2InstanceDefaultPolicy`, für die Sie die Berechtigung zur Nutzung Systems Manager von Funktionen benötigen. Sie sollten diese Rolle nicht anderen IAM-Entitäten wie IAM-Benutzern und IAM-Gruppen oder IAM-Rollen zuordnen, die anderen Zwecken dienen. Weitere Informationen finden Sie unter Verwenden der Standardeinstellung für die Host-Management-Konfiguration.

Diese Richtlinie erteilt Berechtigungen, die es dem SSM Agent auf Ihrer Amazon-EC2-Instance ermöglichen, Documents abzurufen, Befehle mit Run Command auszuführen, Sitzungen mit Session Manager einzurichten, eine Bestandsaufnahme der Instance zu erfassen und mit Patch Manager nach Patches und Patch-Compliance zu suchen.

Systems Manager verwendet für jede Instance ein personalisiertes Autorisierungs-Token, um sicherzustellen, dass SSM Agent die API-Operationen auf der richtigen Instance ausführt. Systems Manager validiert das personalisierte Autorisierungs-Token anhand des Amazon-Ressourcennamens (ARN) der Instance, der in der API-Operation angegeben wurde.

Die Berechtigungsrichtlinie der Rolle `AmazonSSMManagedEC2InstanceDefaultPolicy` ermöglicht Systems Manager die Durchführung der folgenden Aktionen für alle verwandten Ressourcen:

- `ssm:DescribeAssociation`
- `ssm:GetDeployablePatchSnapshotForInstance`
- `ssm:GetDocument`
- `ssm:DescribeDocument`
- `ssm:GetManifest`
- `ssm:ListAssociations`
- `ssm:ListInstanceAssociations`

- ssm:PutInventory
- ssm:PutComplianceItems
- ssm:PutConfigurePackageResult
- ssm:UpdateAssociationStatus
- ssm:UpdateInstanceAssociationStatus
- ssm:UpdateInstanceInformation
- ssmmessages:CreateControlChannel
- ssmmessages:CreateDataChannel
- ssmmessages:OpenControlChannel
- ssmmessages:OpenDataChannel
- ec2messages:AcknowledgeMessage
- ec2messages:DeleteMessage
- ec2messages:FailMessage
- ec2messages:GetEndpoint
- ec2messages:GetMessages
- ec2messages:SendReply

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AmazonSSMManagedEC2-Richtlinie im Referenzhandbuch für verwaltete Richtlinien. InstanceDefaultAWS

Systems ManagerAktualisierungen der verwalteten Richtlinien AWS

In der folgenden Tabelle finden Sie Details zu Aktualisierungen AWS verwalteter Richtlinien Systems Manager seit Beginn der Erfassung dieser Änderungen durch diesen Dienst am 12. März 2021. Informationen zu anderen verwalteten Richtlinien für den Systems Manager Manager-Dienst finden Sie Zusätzliche verwaltete Richtlinien für Systems Manager weiter unten in diesem Thema. Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf der Systems Manager Dokumentverlauf-Seite.

Änderung	Beschreibung	Datum
AWSSystemsManagerOpsDataSyncServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie.	OpsCenterDie Richtlinie wurde aktualisiert, um die Sicherheit des Dienstcodes innerhalb der OpsData dienstbezogenen Rolle für die Verwaltung verwandter Vorgänge Explorer zu verbessern.	28. Juni 2023
AmazonSSMManagedEC2InstanceDefaultPolicy - Neue Richtlinie.	Systems Manager hat eine neue Richtlinie hinzugefügt, um die Systems Manager-Funktionalität auf Amazon-EC2-Instances ohne die Verwendung eines IAM-Instance-Profils zuzulassen.	18. August 2022
ServiceRoleAmazonSSM-Richtlinie — Aktualisierung einer bestehenden Richtlinie.	Systems Manager hat neue Berechtigungen hinzugefügt, damit Explorer eine verwaltete Regel erstellen kann, wenn Sie Security Hub von Explorer oder OpsCenter aktivieren. Neue Berechtigungen wurden hinzugefügt, um zu überprüfen, ob die Konfiguration und der Compute-Optimizer die erforderlichen Anforderungen erfüllen, bevor sie zugelassen werden. OpsData	27. April 2021
AWSSystemsManagerOpsDataSyncServiceRolePolicy - Neue Richtlinie.	Systems Managerhat eine neue Richtlinie zum Erstellen und Aktualisieren OpsData von OpsItems und aus Security Hub Hub-Erken	27. April 2021

Änderung	Beschreibung	Datum
	ntnissen in Explorer und hinzugefügtOpsCenter.	
AmazonSSMServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie.	Systems Managerneue Berechtigungen hinzugefügt, um die Anzeige von Aggregaten OpsData und OpsItems Details aus mehreren Konten und AWS-Regionen in zu ermöglichenExplorer.	24. März 2021
Systems Manager hat die Änderungsverfolgung gestartet	Systems Managerhat begonnen, Änderungen an den AWS verwalteten Richtlinien zu verfolgen.	12. März 2021

Zusätzliche verwaltete Richtlinien für Systems Manager

Zusätzlich zu den weiter oben in diesem Thema beschriebenen verwalteten Richtlinien werden die folgenden Richtlinien auch von Systems Manager unterstützt.

- AmazonSSMAutomationApproverAccess— AWS verwaltete Richtlinie, die den Zugriff ermöglicht, um Automatisierungsausführungen einzusehen und Genehmigungsentscheidungen an die Automatisierung zu senden, die auf die Genehmigung wartet.
- AmazonSSMAutomationRole— AWS verwaltete Richtlinie, die dem Systems Manager Automatisierungsdienst Berechtigungen zur Ausführung von Aktivitäten gewährt, die in Automatisierungs-Runbooks definiert sind. Weisen Sie diese Richtlinie Administratoren und vertrauenswürdigen Hauptbenutzern zu.
- AmazonSSMDirectoryServiceAccess— AWS verwaltete Richtlinie, die SSM Agent den Zugriff im Namen des Benutzers AWS Directory Service auf Anfragen zum Beitritt zur Domäne durch den verwalteten Knoten ermöglicht.
- AmazonSSMFullAccess— AWS verwaltete Richtlinie, die vollen Zugriff auf die Systems Manager API und Dokumente gewährt.

- AmazonSSMMaintenanceWindowRole— AWS verwaltete Richtlinie, die Wartungsfenstern Berechtigungen für die Systems Manager Manager-API gewährt.
- AmazonSSMManagedInstanceCore – Von AWS verwaltete Richtlinie, die einem Knoten die Nutzung von Systems Manager-Servicekern-Funktionalität erlaubt.
- AmazonSSMPatchAssociation— AWS verwaltete Richtlinie, die den Zugriff auf untergeordnete Instanzen für Patch-Zuordnungsvorgänge ermöglicht.
- AmazonSSMReadOnlyAccess— AWS verwaltete Richtlinie, die Zugriff auf Systems Manager schreibgeschützte API-Operationen wie und gewährt. `Get*` `List*`
- AWSSSMOpsInsightsServiceRolePolicy— AWS verwaltete Richtlinie, die Berechtigungen für die Erstellung und Aktualisierung betrieblicher Einblicke in gewährt OpsItems. Systems Manager Wird verwendet, um Berechtigungen über die serviceverknüpfte Rolle AWSServiceRoleForAmazonSSM_OpsInsightsbereitzustellen.
- AWSSystemsManagerAccountDiscoveryServicePolicy— AWS verwaltete Richtlinie, die Systems Manager die Erlaubnis erteilt, AWS-Konto Informationen zu ermitteln.
- AWSSystemsManagerChangeManagementServicePolicy— AWS verwaltete Richtlinie, die Zugriff auf AWS Ressourcen gewährt, die vom Systems Manager Change-Management-Framework verwaltet oder genutzt und von der dienstbezogenen Rolle `AWSServiceRoleForSystemsManagerChangeManagement` genutzt werden.
- AmazonEC2RoleforSSM— Diese Richtlinie wird nicht mehr unterstützt und sollte nicht verwendet werden. Verwenden Sie stattdessen die `AmazonSSMManagedInstanceCore` Richtlinie, um die Kernfunktionen des Systems Manager Dienstes auf EC2-Instances zuzulassen. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

Fehlerbehebung für AWS Systems Manager-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die beim Arbeiten mit AWS Systems Manager undAWS Identity and Access Management (IAM) auftreten könnten.

Themen

- Ich bin nicht autorisiert, eine Aktion in Systems Manager auszuführen.
- Ich bin nicht zur Ausführung von iam:PassRole autorisiert
- Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine Systems Manager-Ressourcen gewähren

Ich bin nicht autorisiert, eine Aktion in Systems Manager auszuführen.

Wenn die AWS Management Console Ihnen mitteilt, dass Sie nicht zur Ausführung einer Aktion autorisiert sind, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson`-Benutzer versucht, die Konsole zum Anzeigen von Details zu einem Dokument zu verwenden, jedoch nicht über `ssm:GetDocument`-Berechtigungen verfügt.

```
User: arn:aws:ssm::123456789012:user/mateojackson isn't authorized to perform:
 ssm:GetDocument on resource: MyExampleDocument
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion `MyExampleDocument` auf die Ressource `ssm:GetDocument` zugreifen zu können.

Ich bin nicht zur Ausführung von iam:PassRole autorisiert

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Ausführung der Aktion `iam:PassRole` autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an Systems Manager übergeben zu können.

Einige AWS-Services erlauben die Übergabe einer vorhandenen Rolle an diesen Service, sodass keine neue Servicerolle oder serviceverknüpfte Rolle erstellt werden muss. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Service.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Systems Manager auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine Systems Manager-Ressourcen gewähren

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob Systems Manager diese Funktionen unterstützt, finden Sie unter Funktionsweise von AWS Systems Manager mit IAM.
- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konto im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie AWS-Konten-Drittanbieter Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter Gewähren des Zugriffs auf AWS-Konten von externen Benutzern im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

Verwenden von serviceverknüpften Rollen für Systems Manager

AWS Systems Managerverwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Systems Manager verknüpft ist. Serviceverknüpfte Rollen werden von Systems Manager vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

> **Note**
>
> Eine Servicerolle Rolle unterscheidet sich von einer servicegebundenen Rolle. Eine Servicerolle ist eine Art von AWS Identity and Access Management (IAM-) Rolle, die einer Person Berechtigungen erteilt, AWS-Service sodass der Dienst auf Ressourcen zugreifen kann. AWS Nur einige Systems Manager-Szenarien erfordern eine Servicerolle. Wenn Sie eine Servicerolle für Systems Manager erstellen, wählen Sie die dafür zu erteilenden Berechtigungen aus, damit ein Zugriff auf oder eine Interaktion mit anderen AWS - Ressourcen möglich ist.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von Systems Manager, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Systems Manager definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur Systems Manager die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Systems Manager-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

> **Note**
>
> Für Nicht-EC2-Knoten in einer Hybrid- und Multi-Cloud-Umgebung benötigen Sie eine zusätzliche IAM-Rolle, die es diesen Maschinen ermöglicht, mit dem Systems Manager-Service zu kommunizieren. Dies ist die IAM-Servicerolle für Systems Manager. Diese Rolle gewährt AWS Security Token Service (AWS STS) AssumeRoleVertrauen in den Systems Manager Dienst. Die `AssumeRole`-Aktion gibt temporäre Sicherheitsanmeldeinformationen zurück (bestehend aus einer Zugriffsschlüssel-ID, einem geheimen Zugriffsschlüssel und einem Sicherheits-Token). Sie verwenden diese temporären Anmeldeinformationen, um auf AWS Ressourcen zuzugreifen, auf die Sie normalerweise keinen Zugriff haben. Weitere Informationen finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Servicerolle in Hybrid- und Multicloud-Umgebungen und AssumeRolein der AWS Security Token Service API-Referenz.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS-Services , die mit IAM arbeiten. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Themen

- Verwenden von Rollen zum Sammeln und Anzeigen von Inventar OpsData
- Verwenden von Rollen zum Sammeln von AWS-Konto Informationen für OpsCenter und Explorer
- Verwenden von Rollen zum Erstellen OpsData und OpsItems für Explorer
- Verwenden von Rollen zur Erstellung von OpsItems für betriebliche Einblicke in Systems Manager OpsCenter
- Rollen zum Exportieren verwenden Explorer OpsData

Verwenden von Rollen zum Sammeln und Anzeigen von Inventar OpsData

Systems Managerverwendet die angegebene dienstbezogene Rolle. **`AWSServiceRoleForAmazonSSM`** AWS Systems Manager verwendet diese IAM-Servicerolle, um AWS Ressourcen in Ihrem Namen zu verwalten.

Dienstbezogene Rollenberechtigungen für Inventar, und OpsData OpsItems

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM` vertraut nur darauf, dass `ssm.amazonaws.com` die Rolle annimmt.

Sie können die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM` im Systems Manager für Folgendes verwenden:

- Die Inventory-Funktion in Systems Manager verwendet die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM` zum Erfassen von Bestand-Metadaten von Tags und Ressourcengruppen.
- Die Explorer Funktion verwendet die dienstbezogene Rolle, `AWSServiceRoleForAmazonSSM` um das Anzeigen OpsItems von OpsData und von mehreren Konten aus zu ermöglichen. Außerdem ermöglicht diese serviceverknüpfte Rolle Explorer, eine verwaltete Regel zu erstellen, wenn Sie Security Hub als Datenquelle von Explorer oder OpsCenter aktivieren.

> **Important**
>
> Bisher bot Ihnen die Systems Manager Manager-Konsole die Möglichkeit, die AWS verwaltete, mit dem IAM-Dienst verknüpfte Rolle `AWSServiceRoleForAmazonSSM` als Wartungsrolle für Ihre Aufgaben auszuwählen. Die Verwendung dieser Rolle und der zugehörigen Richtlinie, `AmazonSSMServiceRolePolicy`, für Wartungsfenster-Aufgaben wird nicht mehr empfohlen. Wenn Sie diese Rolle jetzt für Wartungsfenster-Aufgaben verwenden, empfehlen wir Ihnen, sie nicht mehr zu verwenden. Erstellen Sie stattdessen Ihre eigene IAM-Rolle, die die Kommunikation zwischen Systems Manager und anderen AWS-Services ermöglicht, wenn Ihre Wartungsfenster-Aufgaben ausgeführt werden. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

Die verwaltete Richtlinie, die zum Bereitstellen von Berechtigungen für die `AWSServiceRoleForAmazonSSM`-Rolle verwendet wird, ist `AmazonSSMServiceRolePolicy`. Einzelheiten zu den Berechtigungen, gewährt werden, finden Sie unter AWS verwaltete Richtlinie: ServiceRole AmazonSSM-Richtlinie.

Erstellen einer **`AWSServiceRoleForAmazonSSM`**-serviceverknüpften Rolle für Systems Manager

Sie können die IAM-Konsole verwenden, um eine serviceverknüpfte Rolle mit dem Anwendungsfall EC2 zu erstellen. Erstellen Sie mithilfe von Befehlen für IAM in AWS Command Line Interface (AWS CLI) oder mithilfe der IAM-API eine dienstverknüpfte Rolle mit dem `ssm.amazonaws.com`-Servicenamen. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen.

Bearbeiten einer **`AWSServiceRoleForAmazonSSM`**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AWSServiceRoleForAmazonSSM` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AWSServiceRoleForAmazonSSM**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie können die IAM-Konsole, die oder die IAM-API verwenden AWS CLI, um die dienstverknüpfte Rolle manuell zu löschen. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zuerst manuell bereinigen, bevor Sie diese manuell löschen können.

Da die serviceverknüpfte Rolle für `AWSServiceRoleForAmazonSSM` mehrere Funktionen verwendet werden kann, müssen Sie sicherstellen, dass keine der Funktionen die Rolle verwendet, bevor Sie versuchen, diese zu löschen.

- Inventory: Wenn Sie die von der Bestands-Funktion verwendete serviceverknüpfte Rolle löschen, werden die Bestands-Daten für Tags und Ressourcengruppen nicht mehr synchronisiert. Sie müssen die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.
- Explorer: Wenn Sie die serviceverknüpfte Rolle löschen, die von der Explorer Funktion verwendet wird, sind die konto- und regionsübergreifenden Rollen nicht mehr OpsData sichtbar. OpsItems

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, Tags oder Ressourcengruppen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Löschen Sie die von **AWSServiceRoleForAmazonSSM** verwendeten Systems Manager-Ressourcen wie folgt:

1. Weitere Informationen zum Löschen von Tags finden Sie unter Hinzufügen und Löschen von Tags für einzelne Ressourcen.
2. Informationen zum Löschen von Ressourcengruppen finden Sie unter Gruppen löschen von. AWS Resource Groups

To manually delete the **AWSServiceRoleForAmazonSSM** service-linked role using IAM (So löschen Sie die servicegebundene Rolle mit IAM)

Verwenden Sie die IAM-Konsole, die oder die IAM-API AWS CLI, um die `AWSServiceRoleForAmazonSSM` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **AWSServiceRoleForAmazonSSM**-serviceverknüpfte Rolle

Systems Managerunterstützt die Verwendung der `AWSServiceRoleForAmazonSSM` serviceverknüpften Rolle überall dort, AWS-Regionen wo der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Systems Manager Endpunkte und -Kontingente.

Verwenden von Rollen zum Sammeln von AWS-Konto Informationen für OpsCenter und Explorer

Systems Manager verwendet die serviceverknüpfte Rolle namens **AWSServiceRoleForAmazonSSM_AccountDiscovery**. AWS Systems Manager verwendet diese IAM-Servicerolle, um andere aufzurufen, um AWS-Konto Informationen AWS-Services zu ermitteln.

Berechtigungen von serviceverknüpften Rollen für Systems Manager-Kontoerkennung

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM_AccountDiscovery` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `accountdiscovery.ssm.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- `organizations:DescribeAccount`
- `organizations:DescribeOrganizationalUnit`
- `organizations:DescribeOrganization`
- `organizations:ListAccounts`
- `organizations:ListAWSServiceAccessForOrganization`
- `organizations:ListChildren`
- `organizations:ListParents`

- `organizations:ListDelegatedServicesForAccount`
- `organizations:ListDelegatedAdministrators`
- `organizations:ListRoots`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-serviceverknüpften Rolle für Systems Manager

Sie müssen eine serviceverknüpfte Rolle erstellen, wenn Sie Explorer- und OpsCenter-Funktionen von Systems Manager über mehrere AWS-Konten hinweg verwenden möchten. Für OpsCenter müssen Sie die serviceverknüpfte Rolle manuell erstellen. Weitere Informationen finden Sie unter (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

Wenn Sie für Explorer eine Ressourcendatensynchronisierung erstellen, indem Sie Systems Manager in der AWS Management Console verwenden, können Sie die serviceverknüpfte Rolle erstellen, indem Sie die Schaltfläche Create role (Rolle erstellen) auswählen. Wenn Sie eine Resource Data Sync programmgesteuert erstellen möchten, müssen Sie die Rolle erstellen, bevor Sie die Resource Data Sync erstellen. Sie können die Rolle mithilfe der CreateServiceLinkedRole-API-Operation erstellen.

Bearbeiten einer **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AWSServiceRoleForAmazonSSM_AccountDiscovery` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität

vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen der **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-serviceverknüpften Rolle

Bevor Sie mit IAM eine `AWSServiceRoleForAmazonSSM_AccountDiscovery`-serviceverknüpfte Rolle löschen können, müssen Sie zunächst alle Explorer Resource Data Syncs löschen. Weitere Informationen finden Sie unter Löschen einer Systems-Manager-Explorer-Ressourcendatensynchronisierung.

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Manuelles Löschen der **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die oder die - AWS API AWS CLI, um die `AWSServiceRoleForAmazonSSM_AccountDiscovery` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-serviceverknüpfte Rolle

Systems Manager unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Systems Manager Endpunkte und -Kontingente.

Aktualisierungen der serviceverknüpfte AWSServiceRoleForAmazonSSM_AccountDiscovery-Rolle

Anzeigen von Details zu Aktualisierungen der AWSServiceRoleForAmazonSSM_AccountDiscovery serviceverknüpften Rolle, seit dieser Service mit der Verfolgung dieser Änderungen begonnen hat.

Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf der Systems Manager Dokumentverlauf-Seite.

Änderung	Beschreibung	Datum
Neue Berechtigungen hinzugefügt	Diese serviceverknüpfte Rolle enthält jetzt `organizations:DescribeOrganizationalUnit` - und `organizations:ListRoots` -Berechtigungen. Diese Berechtigungen ermöglichen es einem - AWS Organizations Verwaltungskonto oder einem delegierten Administratorkonto von Systems Manager, OpsItemskontenübergreifend mit zu arbeiten. Weitere Informationen finden Sie unter (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.	17. Oktober 2022

Verwenden von Rollen zum Erstellen OpsData und OpsItems für Explorer

Systems Managerverwendet die benannte dienstverknüpfte Rolle. **`AWSServiceRoleForSystemsManagerOpsDataSync`** AWS Systems Manager verwendet diese IAM-Dienstrolle Explorer zum Erstellen OpsData von und. OpsItems

Mit dem Dienst verknüpfte Rollenberechtigungen für die Synchronisierung Systems Manager OpsData

Die serviceverknüpfte Rolle `AWSServiceRoleForSystemsManagerOpsDataSync` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `opsdatasync.ssm.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- Systems Manager Explorer erfordert, dass eine serviceverknüpfte Rolle die Berechtigung zum Aktualisieren eines Sicherheitsergebnisses erteilt, wenn ein OpsItem aktualisiert wird. Erstellen und aktualisieren Sie ein OpsItem und deaktivieren Sie die Security-Hub-Datenquelle, wenn eine SSM-verwaltete Regel von Kunden gelöscht wird.

Die verwaltete Richtlinie, die zum Bereitstellen von Berechtigungen für die `AWSServiceRoleForSystemsManagerOpsDataSync`-Rolle verwendet wird, ist `AWSSystemsManagerOpsDataSyncServiceRolePolicy`. Einzelheiten zu den Berechtigungen, gewährt werden, finden Sie unter AWS verwaltete Richtlinie: AWSSystemsManagerOpsDataSyncServiceRolePolicy.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **`AWSServiceRoleForSystemsManagerOpsDataSync`**-serviceverknüpften Rolle für Systems Manager

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie die Option aktivieren Explorer AWS Management Console, Systems Manager wird die dienstverknüpfte Rolle für Sie erstellt.

> **Important**
>
> Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Features verwendet. Wenn Sie den Systems Manager-Service vor dem 1. Januar 2017 verwendet haben, als dieser begann, serviceverknüpfte Rollen zu unterstützen, dann hat Systems Manager die Rolle `AWSServiceRoleForSystemsManagerOpsDataSync` in Ihrem Konto erstellt. Weitere Informationen finden Sie unter In meinem IAM-Konto wird eine neue Rolle angezeigt.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie Explorer in aktivieren AWS Management Console, Systems Manager wird die dienstverknüpfte Rolle erneut für Sie erstellt.

Sie können auch die IAM-Konsole verwenden, um eine serviceverknüpfte Rolle mit der AWS Servicerolle zu erstellen, mit der Sie einen Fall erstellen OpsData und OpsItems verwenden können Explorer. Erstellen Sie in der AWS CLI oder der AWS API eine dienstbezogene Rolle mit dem `opsdatasync.ssm.amazonaws.com` Dienstnamen. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Bearbeiten einer **`AWSServiceRoleForSystemsManagerOpsDataSync`**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AWSServiceRoleForSystemsManagerOpsDataSync` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **`AWSServiceRoleForSystemsManagerOpsDataSync`**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Das Verfahren zum Löschen Systems Manager-Ressourcen, das von der `AWSServiceRoleForSystemsManagerOpsDataSync`-Rolle verwendet wird, hängt davon ab, ob Sie Explorer oder OpsCenter für die Integration in den Security Hub konfiguriert haben.

Löschen Sie die von der **`AWSServiceRoleForSystemsManagerOpsDataSync`**-Rolle verwendeten Systems Manager-Ressourcen wie folgt:

- Um zu verhindern, dass Explorer neue OpsItems für Security Hub-Ergebnisse erstellt, lesen Sie Empfangen von Ergebnisse stoppen.
- Informationen dazu, wie OpsCenter daran gehindert wird, neue OpsItems für Security-Hub-Ergebnisse zu erstellen, finden Sie unter

To manually delete the **`AWSServiceRoleForSystemsManagerOpsDataSync`** service-linked role using IAM (So löschen Sie die servicegebundene Rolle mit IAM)

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die `AWSServiceRoleForSystemsManagerOpsDataSync` dienstverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **`AWSServiceRoleForSystemsManagerOpsDataSync`**-serviceverknüpfte Rolle

Systems Manager unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Systems Manager Endpunkte und -Kontingente.

Systems Manager unterstützt die Verwendung von serviceverknüpften Rollen nicht in allen Regionen, in denen der Service verfügbar ist. Sie können die Rolle `AWSServiceRoleForSystemsManagerOpsDataSync` in den folgenden Regionen verwenden.

AWS-Region Name	Regions-ID	Unterstützung in Systems Manager
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja

AWS-Region Name	Regions-ID	Unterstützung in Systems Manager
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Osaka)	ap-northeast-3	Ja
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Kanada (Zentral)	ca-central-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Ireland)	eu-west-1	Ja
Europa (London)	eu-west-2	Ja
Europa (Paris)	eu-west-3	Ja
Europa (Stockholm)	eu-north-1	Ja
Südamerika (São Paulo)	sa-east-1	Ja
AWS GovCloud (US)	us-gov-west-1	Nein

Verwenden von Rollen zur Erstellung von OpsItems für betriebliche Einblicke in Systems Manager OpsCenter

Systems Manager verwendet die serviceorientierte Rolle namens **`AWSServiceRoleForAmazonSSM_OpsInsights`**. AWS Systems Manager verwendet diese

IAM-Servicerolle zum Erstellen und Aktualisieren von OpsItems für operative Einblicke in Systems Manager OpsCenter.

Berechtigungen von **AWSServiceRoleForAmazonSSM_OpsInsights** serviceverknüpften Rollen für Systems Manager-OpsItems für betriebliche Einblicke

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM_OpsInsights` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `opsinsights.ssm.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

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

```
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **AWSServiceRoleForAmazonSSM_OpsInsights**-serviceverknüpften Rolle für Systems Manager

Sie müssen eine serviceverknüpfte Rolle erstellen. Wenn Sie Operational Insights mithilfe von Systems Manager in der AWS Management Console aktivieren, können Sie die serviceverknüpfte Rolle erstellen, indem Sie die Schaltfläche Enable (Aktivieren) wählen.

Bearbeiten einer **AWSServiceRoleForAmazonSSM_OpsInsights**-serviceverknüpften Rolle für Systems Manager

Systems Manager verhindert die Bearbeitung der serviceverknüpften Rolle `AWSServiceRoleForAmazonSSM_OpsInsights`. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AWSServiceRoleForAmazonSSM_OpsInsights**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine servicegebundene Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen der **AWSServiceRoleForAmazonSSM_OpsInsights**-serviceverknüpften Rolle

Bevor Sie IAM zum Löschen einer serviceverknüpften `AWSServiceRoleForAmazonSSM_OpsInsights`-Rolle verwenden können, müssen Sie zunächst betriebliche Einblicke in Systems Manager OpsCenter deaktivieren. Weitere Informationen finden Sie unter Analyse betrieblicher Einblicke zur Reduzierung von OpsItems.

Manuelles Löschen der **`AWSServiceRoleForAmazonSSM_OpsInsights`**-serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, AWS CLI- oder AWS-API, um die `AWSServiceRoleForAmazonSSM_OpsInsights` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **`AWSServiceRoleForAmazonSSM_OpsInsights`**-serviceverknüpfte Rolle

Systems Manager unterstützt die Verwendung von serviceverknüpften Rollen nicht in allen Regionen, in denen der Service verfügbar ist. Sie können die Rolle AWSServiceRoleForAmazonSSM_OpsInsights in den folgenden Regionen verwenden.

Name der Region	Regions-ID	Unterstützung in Systems Manager
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Hongkong)	ap-east-1	Ja
Kanada (Zentral)	ca-central-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Ireland)	eu-west-1	Ja

Name der Region	Regions-ID	Unterstützung in Systems Manager
Europa (London)	eu-west-2	Ja
Europa (Paris)	eu-west-3	Ja
Europa (Stockholm)	eu-north-1	Ja
Europa (Mailand)	eu-south-1	Ja
Südamerika (São Paulo)	sa-east-1	Ja
Naher Osten (Bahrain)	me-south-1	Ja
Afrika (Kapstadt)	af-south-1	Ja
AWS GovCloud (US)	us-gov-west-1	Ja
AWS GovCloud (US)	us-gov-east-1	Ja

Rollen zum Exportieren verwenden Explorer OpsData

AWS Systems Manager Explorerverwendet die AmazonSSM ExplorerExport Role Service-Rolle, um Betriebsdaten (OpsData) mithilfe des Automatisierungs-Runbooks zu exportieren. `AWS-ExportOpsDataToS3`

Berechtigungen von serviceverknüpften Rollen für Explorer

Die serviceverknüpfte Rolle `AmazonSSMExplorerExportRole` vertraut nur darauf, dass `ssm.amazonaws.com` die Rolle annimmt.

Sie können die `AmazonSSMExplorerExportRole` serviceverknüpfte Rolle verwenden, um Betriebsdaten (OpsData) mithilfe des Automatisierungs-Runbooks zu exportieren. `AWS-ExportOpsDataToS3` Sie können 5.000 OpsData Artikel aus Explorer einer Datei mit kommagetrennten Werten (.csv) in einen Amazon Simple Storage Service (Amazon S3) -Bucket exportieren.

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- `s3:PutObject`
- `s3:GetBucketAcl`
- `s3:GetBucketLocation`
- `sns:Publish`
- `logs:DescribeLogGroups`
- `logs:DescribeLogStreams`
- `logs:CreateLogGroup`
- `logs:PutLogEvents`
- `logs:CreateLogStream`
- `ssm:GetOpsSummary`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **`AmazonSSMExplorerExportRole`**-serviceverknüpften Rolle für Systems Manager

Systems Manager erstellt die `AmazonSSMExplorerExportRole` serviceverknüpfte Rolle, wenn Sie Explorer in der Systems Manager OpsData Manager-Konsole exportieren. Weitere Informationen finden Sie unter OpsData Aus Systems Manager exportieren Explorer.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen.

Bearbeiten einer **`AmazonSSMExplorerExportRole`**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AmazonSSMExplorerExportRole` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AmazonSSMExplorerExportRole**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie können die IAM-Konsole, die oder die IAM-API verwenden AWS CLI, um die dienstverknüpfte Rolle manuell zu löschen. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zuerst manuell bereinigen, bevor Sie diese manuell löschen können.

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, Tags oder Ressourcengruppen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Löschen Sie die von **AmazonSSMExplorerExportRole** verwendeten Systems Manager-Ressourcen wie folgt:

1. Weitere Informationen zum Löschen von Tags finden Sie unter Hinzufügen und Löschen von Tags für einzelne Ressourcen.
2. Informationen zum Löschen von Ressourcengruppen finden Sie unter Gruppen löschen von. AWS Resource Groups

To manually delete the **AmazonSSMExplorerExportRole** service-linked role using IAM (So löschen Sie die servicegebundene Rolle mit IAM)

Verwenden Sie die IAM-Konsole, die oder die IAM-API AWS CLI, um die `AmazonSSMExplorerExportRole` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **AmazonSSMExplorerExportRole**-serviceverknüpfte Rolle

Systems Managerunterstützt die Verwendung der `AmazonSSMExplorerExportRole` serviceverknüpften Rolle überall dort, AWS-Regionen wo der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Systems Manager Endpunkte und -Kontingente.

Protokollieren und Überwachen in AWS Systems Manager

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit AWS Systems Manager und Leistung Ihrer AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen Fehler an mehreren Stellen besser debuggen können, falls einer auftritt. AWS stellt mehrere Tools bereit, mit denen Sie Ihre Systems Manager und andere Ressourcen überwachen und auf mögliche Vorfälle reagieren können.

AWS CloudTrail Logs

CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS-Service Mitglied ausgeführt wurdenSystems Manager. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurdeSystems Manager, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Informationen. Weitere Informationen finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Amazon-Alarmen beobachten Sie eine einzelne Metrik über einen Zeitraum, den Sie für Ihre Amazon Elastic Compute Cloud (Amazon EC2) -Instances und andere Ressourcen angeben. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Thema oder eine AWS Auto Scaling Richtlinie von Amazon Simple Notification Service (Amazon SNS) gesendet. CloudWatch Alarme lösen keine Aktionen aus, da sie sich in einem bestimmten Status befinden. Der Status muss sich stattdessen geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein. Weitere Informationen finden Sie unter Verwenden von CloudWatch Amazon-Alarmen im CloudWatch Amazon-Benutzerhandbuch.

CloudWatch Amazon-Dashboards

CloudWatch Dashboards sind anpassbare Homepages in der CloudWatch Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können, auch die Ressourcen, die auf verschiedene verteilt sind. AWS-Regionen Mithilfe von CloudWatch Dashboards können Sie benutzerdefinierte Ansichten der Metriken und Alarme für Ihre AWS Ressourcen erstellen. Weitere Informationen finden Sie unter Von Systems Manager gehostete CloudWatch Amazon-Dashboards.

Amazon EventBridge

Mithilfe von Amazon können Sie Regeln konfigurieren EventBridge, die Sie über Änderungen an Systems Manager Ressourcen informieren und Sie veranlassen, auf der Grundlage des Inhalts dieser Ereignisse Maßnahmen EventBridge zu ergreifen. EventBridge bietet Unterstützung für eine Reihe von Ereignissen, die von verschiedenen Systems Manager Funktionen ausgelöst werden. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge.

CloudWatch Amazon-Logs und SSM Agent Logs

SSM Agent schreibt Informationen zu Ausführungen, geplanten Aktionen, Fehlern und dem Zustandsstatus in Protokolldateien auf jedem Knoten. Sie können Protokolldateien anzeigen, indem Sie sich manuell mit einem Knoten verbinden. Wir empfehlen, Agenten-Protokolldaten zur Analyse automatisch an eine Protokollgruppe in CloudWatch Logs zu senden. Weitere Informationen finden Sie unter Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent) und Anzeigen von SSM Agent-Protokollen.

AWS Systems Manager-Compliance

Sie können Compliance, eine Funktion von, verwenden AWS Systems Manager, um Ihre Flotte verwalteter Knoten auf Patch-Konformität und Konfigurationsinkonsistenzen zu überprüfen. Sie können Daten aus mehreren Bereichen sammeln und aggregieren AWS-Konten und dann nach bestimmten Ressourcen suchen AWS-Regionen, die nicht den Vorschriften entsprechen. Standardmäßig zeigt Compliance aktuelle Compliance-Daten zum Patch-InPatch Manager, zu einer Fähigkeit von AWS Systems Manager und zu Verknüpfungen zu einer Fähigkeit von an AWS Systems Manager. State Manager Weitere Informationen finden Sie unter AWS Systems Manager-Compliance.

AWS Systems Manager Explorer

Explorer, eine Funktion von AWS Systems Manager, ist ein anpassbares Operations-Dashboard, das Informationen über Ihre AWS Ressourcen enthält. Explorerzeigt eine aggregierte Ansicht der Betriebsdaten (OpsData) für Sie AWS-Konten und Across AWS-Regionen an. OpsData Enthält Metadaten zu Ihren EC2-Instances, Details zur Patch-Konformität und betriebliche Arbeitselemente ()OpsItems. Explorer Explorerbietet Informationen darüber, wie sie auf Ihre Geschäftsbereiche oder Anwendungen verteilt OpsItems sind, wie sie sich im Laufe der Zeit entwickeln und wie sie sich je nach Kategorie unterscheiden. Sie können Informationen in Explorer gruppieren und filtern, um sich auf die Elemente zu konzentrieren, die für Sie relevant sind und eine Aktion erfordern. Weitere Informationen finden Sie unter AWS Systems Manager Explorer.

AWS Systems Manager OpsCenter

OpsCenter, eine Funktion von AWS Systems Manager, bietet einen zentralen Ort, an dem Betriebsingenieure und IT-Experten betriebliche Arbeitsaufgaben (OpsItems) im Zusammenhang mit AWS Ressourcen einsehen, untersuchen und lösen können. OpsCenteraggregiert und standardisiert OpsItems alle Services und stellt gleichzeitig kontextbezogene Untersuchungsdaten zu den einzelnen OpsItemOpsItems, zugehörigen und verwandten Ressourcen bereit. OpsCenterstellt außerdem Runbooks in Automation bereit, eine Funktion AWS Systems Manager, mit der Sie Probleme schnell lösen können. OpsCenterist in Amazon integriert EventBridge. Das bedeutet, dass Sie EventBridge Regeln erstellen können, die automatisch OpsItems für alle erstellt werden, für AWS-Service die Ereignisse veröffentlicht EventBridge werden. Weitere Informationen finden Sie unter AWS Systems Manager OpsCenter.

Amazon Simple Notification Service

Sie können Amazon Simple Notification Service (Amazon SNS) zum Senden von Benachrichtigungen über den Status der Befehle konfigurieren, die Sie mithilfe von Run Command oder Maintenance Windows, Funktionen von AWS Systems Manager, senden. Amazon SNS koordiniert und verwaltet das Senden und Zustellen von Benachrichtigungen an Clients oder Endpunkte, die Amazon SNS-Themen abonniert haben. Sie können eine Benachrichtigung erhalten, wenn ein Befehl in einen neuen Status oder in einen bestimmten Status wechselt, z. B. `Failed` oder `Timed Out`. In Fällen, in denen Sie einen Befehl an mehrere Knoten senden, können Sie eine Benachrichtigung für jede Kopie des Befehls abrufen, die an einen bestimmten Knoten gesendet wurde. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

AWS Trusted Advisor und AWS Health Dashboard

Trusted Advisor stützt sich auf bewährte Verfahren, die wir bei der Betreuung von Hunderttausenden von AWS Kunden gelernt haben. Trusted Advisor untersucht Ihre AWS Umgebung und gibt dann Empfehlungen, wenn Möglichkeiten bestehen, Geld zu sparen, die Systemverfügbarkeit und -leistung zu verbessern oder Sicherheitslücken zu schließen. Alle AWS Kunden haben Zugriff auf fünf Trusted Advisor Schecks. Kunden mit einem AWS Support Business- oder Enterprise-Tarif können alle Trusted Advisor Schecks einsehen. Weitere Informationen finden Sie unter AWS Trusted Advisor im AWS Support -Benutzerhandbuch und im AWS Health -Benutzerhandbuch.

Weitere Informationen

- Überwachung AWS Systems Manager

Compliance-Validierung für AWS Systems Manager

Dieses Thema befasst sich mit der AWS Systems Manager-Compliance mit Assurance-Programmen von Drittanbietern. Informationen zum Anzeigen von Compliance-Daten für Ihre verwalteten Knoten finden Sie unter AWS Systems Manager-Compliance.

Externe Prüfer bewerten im Rahmen verschiedener AWS-Compliance-Programme die Sicherheit und Compliance von Systems Manager. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Eine Liste der AWS-Services, die in den Anwendungsbereich bestimmter Compliance-Programme fallen, finden Sie unter AWS-Services im Anwendungsbereich nach Compliance-Programm . Allgemeine Informationen finden Sie unter AWS-Compliance-Programme.

Sie können Auditberichte von Drittanbietern unter AWS Artifact herunterladen. Weitere Informationen finden Sie unter Berichte herunterladen in AWS Artifact.

Ihre Compliance-Verantwortung bei der Verwendung von Systems Manager ist von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften abhängig. AWS stellt die folgenden Ressourcen zur Unterstützung der Compliance bereit:

- Kurzanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden finden Sie wichtige Überlegungen zur Architektur sowie die einzelnen Schritte zur Bereitstellung von sicherheits- und Compliance-orientierten Basisumgebungen in AWS.
- Whitepaper zur Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe von AWS HIPAA-konforme Anwendungen erstellen können.
- AWS-Compliance-Ressourcen – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort relevant sein.
- Auswertung von Ressourcen mit Regeln im AWS ConfigEntwicklerhandbuch – Der AWS Config-Service bewertet, wie gut Ihre Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften übereinstimmen.
- AWS Security Hub – Dieser AWS-Service liefert einen umfassenden Überblick über den Sicherheitsstatus in AWS. So können Sie die Compliance mit den Sicherheitsstandards in der Branche und den bewährten Methoden abgleichen.

Ausfallsicherheit in AWS Systems Manager

Die globale AWS-Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit geringer Latenz, hohem Durchsatz und hoher Redundanz verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter Globale AWS-Infrastruktur.

Sicherheit der Infrastruktur in AWS Systems Manager

Als verwalteter Service ist AWS Systems Manager durch die globalen Verfahren zur Gewährleistung der Netzwerksicherheit von AWS geschützt. Informationen zu AWS-Sicherheitsdiensten und wie AWS die Infrastruktur schützt, finden Sie unter AWSCloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastrukturschutz im Security Pillar AWS Well-Architected Framework.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Systems Manager zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Konfigurations- und Schwachstellenanalyse in AWS Systems Manager

AWS übernimmt grundlegende Sicherheitsaufgaben wie Firewall-Konfiguration und Notfallwiederherstellung. Diese Verfahren wurden von qualifizierten Dritten überprüft und zertifiziert. Weitere Informationen finden Sie in den folgenden Ressourcen:

- Compliance-Validierung für AWS Systems Manager
- Modell der übergreifenden Verantwortlichkeit
- Bewährte Methoden für Sicherheit, Identität und Compliance

Bewährte Methoden für die Sicherheit für Systems Manager

AWS Systems Manager bietet eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

Themen

- Bewährte Methoden für vorbeugende Systems Manager-Sicherheitsmaßnahmen
- Bewährte Methoden zur Überwachung und Prüfung von Systems Manager

Bewährte Methoden für vorbeugende Systems Manager-Sicherheitsmaßnahmen

Die folgenden bewährten Methoden für Systems Manager können dabei helfen, Sicherheitsvorfälle zu verhindern.

Implementieren des Zugriffs mit geringsten Berechtigungen

Beim Erteilen von Berechtigungen entscheiden Sie, wer welche Berechtigungen für welche Systems Manager-Ressourcen erhält. Sie aktivieren die spezifischen Aktionen, die daraufhin für die betreffenden Ressourcen erlaubt sein sollen. Aus diesem Grund sollten Sie nur Berechtigungen gewähren, die zum Ausführen einer Aufgabe erforderlich sind. Die Implementierung der geringstmöglichen Zugriffsrechte ist eine grundlegende Voraussetzung

zum Reduzieren des Sicherheitsrisikos und der Auswirkungen, die aufgrund von Fehlern oder böswilligen Absichten entstehen könnten.

Die folgenden Tools stehen zur Implementierung der geringstmöglichen Zugriffsrechte zur Verfügung:

- IAM-Richtlinien und Berechtigungsgrenzen für IAM-Entitäten
- Service-Kontrollrichtlinien

Verwenden Sie die empfohlenen EinstellungenSSM Agent, wenn Sie für die Verwendung eines Proxys konfiguriert sind

Wenn Sie die Verwendung eines Proxys konfigurierenSSM Agent, verwenden Sie die `no_proxy` Variable mit der IP-Adresse des Metadatendienstes der Systems Manager-Instanz, um sicherzustellen, dass Aufrufe von Systems Manager nicht die Identität des Proxydienstes annehmen.

Weitere Informationen finden Sie unter Konfiguration SSM Agent für die Verwendung eines Proxys auf Linux-Knoten und Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances.

Verwenden Sie SecureString Parameter, um geheime Daten zu verschlüsseln und zu schützen

Bei Parameter Store einer Fähigkeit von handelt es AWS Systems Manager sich bei einem `SecureString` Parameter um alle sensiblen Daten, die auf sichere Weise gespeichert und referenziert werden müssen. Wenn Sie über Daten verfügen, die Benutzer nicht ändern oder im Klartext referenzieren sollen, z. B. Kennwörter oder Lizenzschlüssel, erstellen Sie diese Parameter mithilfe des `SecureString` Datentyps. Parameter Storeverwendet ein AWS KMS key in AWS Key Management Service (AWS KMS), um den Parameterwert zu verschlüsseln. AWS KMS verwendet Von AWS verwalteter Schlüssel beim Verschlüsseln des Parameterwerts entweder einen vom Kunden verwalteten Schlüssel oder einen. Für maximale Sicherheit empfehlen wir die Verwendung eines eigenen KMS-Schlüssel. Wenn Sie den verwenden Von AWS verwalteter Schlüssel, kann jeder Benutzer, der berechtigt ist, die GetParameterGetParametersAND-Aktionen in Ihrem Konto auszuführen, den Inhalt aller `SecureString` Parameter anzeigen oder abrufen. Wenn Sie vom Kunden verwaltete Schlüssel zur Verschlüsselung Ihrer sicheren `SecureString`-Werte verwenden, können Sie IAM-Richtlinien und -Schlüsselrichtlinien verwenden, um die Berechtigungen für die Ver- und Entschlüsselung von Parametern zu verwalten. Es ist schwieriger, Richtlinien für die Zugriffssteuerung für diese Vorgänge zu erstellen, wenn Sie die vom Kunden verwalteten Schlüssel verwenden. Wenn Sie beispielsweise die Von AWS verwalteter Schlüssel zum

Verschlüsseln von `SecureString` Parametern verwenden und nicht möchten, dass Benutzer mit `SecureString` Parametern arbeiten, müssen deren IAM-Richtlinien den Zugriff auf den Standardschlüssel ausdrücklich verweigern.

Weitere Informationen finden Sie unter Einschränken des Zugriffs auf Systems Manager-Parameter mithilfe von IAM-Richtlinien und How AWS Systems ManagerParameter Store Uses AWS KMS in the AWS Key Management Service Developer Guide.

Definieren von allowedValues und allowedPattern für Dokumentparameter

Sie können Benutzereingaben für Parameter in Systems Manager-Dokumenten (SSM-Dokumenten) validieren, indem Sie `allowedValues` und `allowedPattern` definieren. Für `allowedValues` definieren Sie ein Array von Werten, die für den Parameter zulässig sind. Wenn ein Benutzer einen Wert eingibt, der nicht zulässig ist, kann die Ausführung nicht gestartet werden. Für `allowedPattern` definieren Sie einen regulären Ausdruck, der überprüft, ob die Benutzereingabe mit dem definierten Muster für den Parameter übereinstimmt. Wenn die Benutzereingabe nicht mit dem zulässigen Muster übereinstimmt, kann die Ausführung nicht gestartet werden.

Weitere Informationen zu `allowedValues` und `allowedPattern` finden Sie unter Datenelemente und Parameter.

Öffentliche Freigabe für Dokumente blockieren

Sofern für Ihren Anwendungsfall keine öffentliche Freigabe erforderlich ist, empfehlen wir Ihnen, die Einstellung zum Blockieren der öffentlichen Freigabe für Ihre SSM-Dokumente im Abschnitt Preferences (Einstellungen) der Systems Manager-Dokumentenkonsole zu aktivieren.

Amazon Virtual Private Cloud (Amazon VPC) und VPC-Endpunkte verwenden

Sie können Amazon VPC verwenden, um AWS Ressourcen in einem von Ihnen definierten virtuellen Netzwerk bereitzustellen. Dieses virtuelle Netzwerk entspricht weitgehend einem herkömmlichen Netzwerk, wie Sie es in Ihrem Rechenzentrum betreiben, kann jedoch die Vorzüge der skalierbaren Infrastruktur von AWS nutzen.

Durch die Implementierung eines VPC-Endpunkts können Sie Ihre VPC privat mit unterstützten AWS-Services und unterstützten VPC-Endpunktdiensten verbinden, AWS PrivateLink ohne dass ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine Verbindung erforderlich ist. AWS Direct Connect Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit Ressourcen im Service zu kommunizieren. Der Datenverkehr zwischen Ihrer VPC und dem anderen Service verlässt das Amazon-Netzwerk nicht.

Weitere Informationen zur Amazon VPC-Sicherheit finden Sie unter Verbessern der Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager und Internetwork Traffic Privacy in Amazon VPC im Amazon VPC-Benutzerhandbuch.

Beschränken Sie Session Manager-Benutzer auf Sitzungen mit interaktiven Befehlen und bestimmten SSM-Sitzungsdokumenten

Session Manager, eine Funktion von AWS Systems Manager, bietet mehrere Methoden zum Starten von Sitzungen für Ihre verwalteten Knoten. Für die sichersten Verbindungen können Sie von den Benutzern verlangen, dass sie sich mit der Methode interaktive Befehle verbinden, um die Benutzerinteraktion auf einen bestimmten Befehl oder eine bestimmte Befehlssequenz zu beschränken. Dies hilft Ihnen bei der Verwaltung der interaktiven Aktionen, die ein Benutzer durchführen kann. Weitere Informationen finden Sie unter Starten einer Sitzung (interaktive und nicht interaktive Befehle).

Für zusätzliche Sicherheit können Sie den Session Manager-Zugriff auf bestimmte Amazon-EC2-Instances und bestimmte Session Manager-Sitzungsdokumente beschränken. Sie gewähren oder widerrufen den Session Manager Zugriff auf diese Weise mithilfe von (IAM-) Richtlinien. AWS Identity and Access Management Weitere Informationen finden Sie unter Schritt 3: Steuern des Sitzungs-Zugriffs auf verwaltete Knoten.

Bereitstellen von temporären Knoten-Berechtigungen für Automatisierungs-Workflows

Während eines Automatisierungs-Workflows, eine Funktion von AWS Systems Manager, benötigen Ihre Knoten möglicherweise Berechtigungen, die nur für diese Ausführung, nicht aber für andere Systems Manager-Operationen benötigt werden. Für einen Automatisierungs-Workflow kann es beispielsweise erforderlich sein, dass ein Knoten während des Workflows eine bestimmte API-Operation aufruft oder auf eine AWS Ressource zugreift. Wenn diese Aufrufe oder Ressourcen solche sind, auf die Sie den Zugriff beschränken möchten, können Sie temporäre, zusätzliche Berechtigungen für Ihre Knoten im Automatisierungs-Runbook selbst bereitstellen, anstatt die Berechtigungen zu Ihrem IAM-Instance-Profil hinzuzufügen. Am Ende des Automation-Workflows werden die temporären Berechtigungen entfernt. Weitere Informationen finden Sie unter Bereitstellung temporärer Instance-Berechtigungen mit AWS Systems Manager Automations im AWS Management- und Governance-Blog.

Halten Sie AWS die Systems Manager Tools auf dem neuesten Stand

AWS veröffentlicht regelmäßig aktualisierte Versionen von Tools und Plugins, die Sie in Ihren AWS Systems Manager Betriebsabläufen verwenden können. Wenn Sie diese Ressourcen auf dem neuesten Stand halten, wird sichergestellt, dass Benutzer und Knoten in Ihrem Konto Zugriff auf die neueste Funktion und Sicherheitsfeatures dieser Tools haben.

- SSM Agent – AWS Systems Manager Agent (SSM Agent) ist eine Amazon-Software, die auf einer Amazon Elastic Compute Cloud (Amazon EC2)-Instance, einem On-Premises-Server oder in einer virtuellen Maschine (VM) installiert und konfiguriert werden kann. SSM Agent ermöglicht es Systems Manager, diese Ressourcen zu aktualisieren, zu verwalten und zu konfigurieren. Es wird empfohlen, mindestens alle zwei Wochen nach neuen Versionen zu suchen oder Aktualisierungen des Agenten zu automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Wir empfehlen außerdem, die Signatur von SSM Agent im Rahmen Ihres Aktualisierungsprozesses zu überprüfen. Weitere Informationen finden Sie unter Verifizieren der Signatur von SSM Agent.
- AWS CLI — The AWS Command Line Interface (AWS CLI) ist ein Open-Source-Tool, mit dem Sie AWS-Services mithilfe von Befehlen in Ihrer Befehlszeilen-Shell interagieren können. Um das zu aktualisieren AWS CLI, führen Sie denselben Befehl aus, mit dem Sie das installiert haben. AWS CLI Wir empfehlen, mindestens alle zwei Wochen eine geplante Aufgabe auf Ihrem lokalen Rechner zu erstellen, um den für Ihr Betriebssystem geeigneten Befehl auszuführen. Informationen zu Installationsbefehlen finden Sie im AWS Command Line Interface Benutzerhandbuch unter Installation der AWS CLI Version 2.
- AWS Tools for Windows PowerShell — Die Tools für Windows PowerShell sind eine Reihe von PowerShell Modulen, die auf der Funktionalität aufbauen, die das AWS SDK for .NET. Sie AWS Tools for Windows PowerShell ermöglichen es Ihnen, Operationen auf Ihren AWS Ressourcen von der PowerShell Befehlszeile aus per Skript auszuführen. Wenn aktualisierte Versionen der Tools für Windows veröffentlicht PowerShell werden, sollten Sie regelmäßig die Version aktualisieren, die Sie lokal ausführen. Weitere Informationen finden Sie unter Aktualisieren von AWS Tools for Windows PowerShell unter Windows oder Aktualisieren von unter Linux oder macOS im IAM Policy Simulator-Benutzerhandbuch. AWS Tools for Windows PowerShell
- Session Manager-Plug-In – Wenn Benutzer in Ihrer Organisation mit den Berechtigungen für die Verwendung von Session Manager eine Verbindung zu einem Knoten mit AWS CLI herstellen möchten, müssen sie zuerst Session Manager auf ihren lokalen Maschinen installieren. Um das Plugin zu aktualisieren, führen Sie denselben Befehl aus, der für die Installation des Plugins verwendet wird. Wir empfehlen, mindestens alle zwei Wochen eine geplante Aufgabe auf Ihrem lokalen Rechner zu erstellen, um den für Ihr Betriebssystem geeigneten Befehl auszuführen. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.
- CloudWatch Agent — Sie können den CloudWatch Agenten konfigurieren und verwenden, um Metriken und Protokolle von Ihren EC2-Instances, lokalen Instanzen und virtuellen Maschinen (VMs) zu sammeln. Diese Protokolle können zur Überwachung und Analyse an Amazon CloudWatch Logs gesendet werden. Es wird empfohlen, mindestens alle zwei Wochen nach

neuen Versionen zu suchen oder Aktualisierungen des Agenten zu automatisieren. Für die einfachsten Aktualisierungen verwenden Sie AWS Systems Manager Quick Setup. Weitere Informationen finden Sie unter AWS Systems Manager Quick Setup.

Bewährte Methoden zur Überwachung und Prüfung von Systems Manager

Mithilfe der folgenden bewährten Methoden für Systems Manager können Sie potenzielle Sicherheitsschwächen und Vorfälle erkennen.

Identifizieren und prüfen all Ihrer Systems Manager-Ressourcen

Die Identifikation Ihrer IT-Assets ist ein wichtiger Aspekt von Governance und Sicherheit. Sie müssen alle Ihre Systems Manager-Ressourcen identifizieren, um ihre Sicherheitssituation zu bewerten und Maßnahmen in potentiellen Schwachstellenbereichen zu ergreifen.

Verwenden Sie den Tag-Editor, um sicherheits- und prüfungsrelevante Ressourcen zu identifizieren. Verwenden Sie dann diese Markierungen zur Suche nach den entsprechenden Ressourcen. Weitere Informationen finden Sie unter Suchen nach zu markierenden Ressourcen im AWS Resource Groups -Benutzerhandbuch.

Erstellen Sie Ressourcengruppen für Ihre Systems Manager-Ressourcen. Weitere Informationen finden Sie unter Was sind Ressourcengruppen?

Implementieren Sie die Überwachung mithilfe der CloudWatch Amazon-Überwachungstools

Die Überwachung ist ein wichtiger Teil der Aufrechterhaltung von Zuverlässigkeit, Sicherheit, Verfügbarkeit und Performance von Systems Manager und Ihren AWS -Lösungen. Amazon CloudWatch bietet verschiedene Tools und Dienste, die Ihnen bei der Überwachung Systems Manager und Ihrer anderen helfen AWS-Services. Weitere Informationen finden Sie unter Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent) und Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge.

Verwenden CloudTrail

AWS CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS-Service Mitglied ausgeführt wurdenSystems Manager. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurdeSystems Manager, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Informationen. Weitere Informationen finden Sie unter AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail.

Einschalten AWS Config

AWS Config ermöglicht es Ihnen, die Konfigurationen Ihrer AWS Ressourcen zu bewerten, zu prüfen und zu bewerten. AWS Config überwacht die Ressourcenkonfigurationen, sodass Sie die aufgezeichneten Konfigurationen mit den erforderlichen sicheren Konfigurationen vergleichen können. Mithilfe AWS Config dieser Funktion können Sie Änderungen an Konfigurationen und Beziehungen zwischen AWS Ressourcen überprüfen, den detaillierten Verlauf der Ressourcenkonfigurationen untersuchen und die allgemeine Konformität mit den in Ihren internen Richtlinien festgelegten Konfigurationen ermitteln. Dadurch können Sie die Compliance-Prüfung, die Sicherheitsanalyse, das Änderungsmanagement und die Fehlerbehebung bei Betriebsabläufen vereinfachen. Weitere Informationen finden Sie unter Einrichten von AWS Config mit der Konsole im AWS Config -Entwicklerhandbuch. Achten Sie bei der Angabe der Ressourcentypen, die aufgezeichnet werden sollen, darauf, dass Systems Manager-Ressourcen enthalten sind.

Überwachen Sie die AWS Sicherheitsempfehlungen

Sie sollten regelmäßig die Trusted Advisor für Sie veröffentlichten Sicherheitshinweise überprüfen. AWS-Konto Sie können dies auch programmgesteuert mit describe-trusted-advisor-checks durchführen.

Überwachen Sie außerdem aktiv die primäre E-Mail-Adresse, die für jeden von Ihnen AWS-Konten registriert ist. AWS wird Sie unter Verwendung dieser E-Mail-Adresse über neu auftretende Sicherheitsprobleme kontaktieren, die Sie betreffen könnten.

AWS Betriebsprobleme mit weitreichenden Auswirkungen werden im AWS Service Health Dashboard veröffentlicht. Operative Probleme werden ebenfalls über das Personal Health Dashboard in den einzelnen Konten gepostet. Weitere Informationen finden Sie in der AWS Health Dokumentation.

Weitere Informationen

- Bewährte Methoden für Sicherheit, Identität und Compliance
- Erste Schritte: Halten Sie sich bei der Konfiguration Ihrer AWS Ressourcen an bewährte Sicherheitsmethoden (AWS Sicherheitsblog)
- Bewährte Methoden für die Sicherheit in IAM
- Bewährte Sicherheitsmethoden in AWS CloudTrail
- Bewährte Methoden für die Sicherheit in Simple Storage Service (Amazon S3)

- Bewährte Sicherheitsmethoden für AWS Key Management Service

Codebeispiele für Systems Manager mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Systems Manager mit einem AWS Software Development Kit (SDK) verwendet wird.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Servicefunktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien und serviceübergreifenden Beispiele sehen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie eine bestimmte Aufgabe ausführen können, indem Sie mehrere Funktionen innerhalb desselben Services aufrufen.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterSystems Manager mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erste Schritte

Hallo Systems Manager

Das folgende Codebeispiel zeigt die ersten Schritte mit Systems Manager.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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);
    }
}
```

- Einzelheiten zur API finden Sie unter ListThings in der AWS SDK for Java 2.x API-Referenz.

Codebeispiele

- Aktionen für Systems Manager mithilfe von AWS SDKs
  - Verwendung AddTagsToResource mit einem AWS SDK oder CLI
  - Verwendung CancelCommand mit einem AWS SDK oder CLI
  - Verwendung CreateActivation mit einem AWS SDK oder CLI
  - Verwendung CreateAssociation mit einem AWS SDK oder CLI
  - Verwendung CreateAssociationBatch mit einem AWS SDK oder CLI
  - Verwendung CreateDocument mit einem AWS SDK oder CLI
  - Verwendung CreateMaintenanceWindow mit einem AWS SDK oder CLI
  - Verwendung CreateOpsItem mit einem AWS SDK oder CLI
  - Verwendung CreatePatchBaseline mit einem AWS SDK oder CLI
  - Verwendung DeleteActivation mit einem AWS SDK oder CLI
  - Verwendung DeleteAssociation mit einem AWS SDK oder CLI
  - Verwendung DeleteDocument mit einem AWS SDK oder CLI
  - Verwendung DeleteMaintenanceWindow mit einem AWS SDK oder CLI
  - Verwendung DeleteParameter mit einem AWS SDK oder CLI
  - Verwendung DeletePatchBaseline mit einem AWS SDK oder CLI
  - Verwendung DeregisterManagedInstance mit einem AWS SDK oder CLI
  - Verwendung DeregisterPatchBaselineForPatchGroup mit einem AWS SDK oder CLI
  - Verwendung DeregisterTargetFromMaintenanceWindow mit einem AWS SDK oder CLI
  - Verwendung DeregisterTaskFromMaintenanceWindow mit einem AWS SDK oder CLI
  - Verwendung DescribeActivations mit einem AWS SDK oder CLI

- Verwendung DescribeAssociation mit einem AWS SDK oder CLI
- Verwendung DescribeAssociationExecutionTargets mit einem AWS SDK oder CLI
- Verwendung DescribeAssociationExecutions mit einem AWS SDK oder CLI
- Verwendung DescribeAutomationExecutions mit einem AWS SDK oder CLI
- Verwendung DescribeAutomationStepExecutions mit einem AWS SDK oder CLI
- Verwendung DescribeAvailablePatches mit einem AWS SDK oder CLI
- Verwendung DescribeDocument mit einem AWS SDK oder CLI
- Verwendung DescribeDocumentPermission mit einem AWS SDK oder CLI
- Verwendung DescribeEffectiveInstanceAssociations mit einem AWS SDK oder CLI
- Verwendung DescribeEffectivePatchesForPatchBaseline mit einem AWS SDK oder CLI
- Verwendung DescribeInstanceAssociationsStatus mit einem AWS SDK oder CLI
- Verwendung DescribeInstanceInformation mit einem AWS SDK oder CLI
- Verwendung DescribeInstancePatchStates mit einem AWS SDK oder CLI
- Verwendung DescribeInstancePatchStatesForPatchGroup mit einem AWS SDK oder CLI
- Verwendung DescribeInstancePatches mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindowExecutionTaskInvocations mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindowExecutionTasks mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindowExecutions mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindowTargets mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindowTasks mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindows mit einem AWS SDK oder CLI
- Verwendung DescribeOpsItems mit einem AWS SDK oder CLI
- Verwendung DescribeParameters mit einem AWS SDK oder CLI
- Verwendung DescribePatchBaselines mit einem AWS SDK oder CLI
- Verwendung DescribePatchGroupState mit einem AWS SDK oder CLI
- Verwendung DescribePatchGroups mit einem AWS SDK oder CLI
- Verwendung GetAutomationExecution mit einem AWS SDK oder CLI
- Verwendung GetCommandInvocation mit einem AWS SDK oder CLI

- Verwendung GetConnectionStatus mit einem AWS SDK oder CLI

- Verwendung GetDefaultPatchBaseline mit einem AWS SDK oder CLI
- Verwendung GetDeployablePatchSnapshotForInstance mit einem AWS SDK oder CLI
- Verwendung GetDocument mit einem AWS SDK oder CLI
- Verwendung GetInventory mit einem AWS SDK oder CLI
- Verwendung GetInventorySchema mit einem AWS SDK oder CLI
- Verwendung GetMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung GetMaintenanceWindowExecution mit einem AWS SDK oder CLI
- Verwendung GetMaintenanceWindowExecutionTask mit einem AWS SDK oder CLI
- Verwendung GetParameterHistory mit einem AWS SDK oder CLI
- Verwendung GetParameters mit einem AWS SDK oder CLI
- Verwendung GetPatchBaseline mit einem AWS SDK oder CLI
- Verwendung GetPatchBaselineForPatchGroup mit einem AWS SDK oder CLI
- Verwendung ListAssociationVersions mit einem AWS SDK oder CLI
- Verwendung ListAssociations mit einem AWS SDK oder CLI
- Verwendung ListCommandInvocations mit einem AWS SDK oder CLI
- Verwendung ListCommands mit einem AWS SDK oder CLI
- Verwendung ListComplianceItems mit einem AWS SDK oder CLI
- Verwendung ListComplianceSummaries mit einem AWS SDK oder CLI
- Verwendung ListDocumentVersions mit einem AWS SDK oder CLI
- Verwendung ListDocuments mit einem AWS SDK oder CLI
- Verwendung ListInventoryEntries mit einem AWS SDK oder CLI
- Verwendung ListResourceComplianceSummaries mit einem AWS SDK oder CLI
- Verwendung ListTagsForResource mit einem AWS SDK oder CLI
- Verwendung ModifyDocumentPermission mit einem AWS SDK oder CLI
- Verwendung PutComplianceItems mit einem AWS SDK oder CLI
- Verwendung PutInventory mit einem AWS SDK oder CLI
- Verwendung PutParameter mit einem AWS SDK oder CLI
- Verwendung RegisterDefaultPatchBaseline mit einem AWS SDK oder CLI
- Verwendung RegisterPatchBaselineForPatchGroup mit einem AWS SDK oder CLI
- Verwendung RegisterTargetWithMaintenanceWindow mit einem AWS SDK oder CLI

- Verwendung RegisterTaskWithMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung RemoveTagsFromResource mit einem AWS SDK oder CLI
- Verwendung SendCommand mit einem AWS SDK oder CLI
- Verwendung StartAutomationExecution mit einem AWS SDK oder CLI
- Verwendung StopAutomationExecution mit einem AWS SDK oder CLI
- Verwendung UpdateAssociation mit einem AWS SDK oder CLI
- Verwendung UpdateAssociationStatus mit einem AWS SDK oder CLI
- Verwendung UpdateDocument mit einem AWS SDK oder CLI
- Verwendung UpdateDocumentDefaultVersion mit einem AWS SDK oder CLI
- Verwendung UpdateMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung UpdateManagedInstanceRole mit einem AWS SDK oder CLI
- Verwendung UpdateOpsItem mit einem AWS SDK oder CLI
- Verwendung UpdatePatchBaseline mit einem AWS SDK oder CLI

- Szenarien für Systems Manager mit AWS SDKs
  - Erste Schritte mit Systems Manager mithilfe eines AWS SDK

Aktionen für Systems Manager mithilfe von AWS SDKs

Die folgenden Codebeispiele zeigen, wie einzelne Systems Manager Manager-Aktionen mit AWS SDKs ausgeführt werden. Diese Auszüge rufen die Systems Manager Manager-API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der AWS Systems Manager -API-Referenz.

Beispiele

- Verwendung AddTagsToResource mit einem AWS SDK oder CLI
- Verwendung CancelCommand mit einem AWS SDK oder CLI
- Verwendung CreateActivation mit einem AWS SDK oder CLI
- Verwendung CreateAssociation mit einem AWS SDK oder CLI
- Verwendung CreateAssociationBatch mit einem AWS SDK oder CLI
- Verwendung CreateDocument mit einem AWS SDK oder CLI

- Verwendung CreateMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung CreateOpsItem mit einem AWS SDK oder CLI
- Verwendung CreatePatchBaseline mit einem AWS SDK oder CLI
- Verwendung DeleteActivation mit einem AWS SDK oder CLI
- Verwendung DeleteAssociation mit einem AWS SDK oder CLI
- Verwendung DeleteDocument mit einem AWS SDK oder CLI
- Verwendung DeleteMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung DeleteParameter mit einem AWS SDK oder CLI
- Verwendung DeletePatchBaseline mit einem AWS SDK oder CLI
- Verwendung DeregisterManagedInstance mit einem AWS SDK oder CLI
- Verwendung DeregisterPatchBaselineForPatchGroup mit einem AWS SDK oder CLI
- Verwendung DeregisterTargetFromMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung DeregisterTaskFromMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung DescribeActivations mit einem AWS SDK oder CLI
- Verwendung DescribeAssociation mit einem AWS SDK oder CLI
- Verwendung DescribeAssociationExecutionTargets mit einem AWS SDK oder CLI
- Verwendung DescribeAssociationExecutions mit einem AWS SDK oder CLI
- Verwendung DescribeAutomationExecutions mit einem AWS SDK oder CLI
- Verwendung DescribeAutomationStepExecutions mit einem AWS SDK oder CLI
- Verwendung DescribeAvailablePatches mit einem AWS SDK oder CLI
- Verwendung DescribeDocument mit einem AWS SDK oder CLI
- Verwendung DescribeDocumentPermission mit einem AWS SDK oder CLI
- Verwendung DescribeEffectiveInstanceAssociations mit einem AWS SDK oder CLI
- Verwendung DescribeEffectivePatchesForPatchBaseline mit einem AWS SDK oder CLI
- Verwendung DescribeInstanceAssociationsStatus mit einem AWS SDK oder CLI
- Verwendung DescribeInstanceInformation mit einem AWS SDK oder CLI
- Verwendung DescribeInstancePatchStates mit einem AWS SDK oder CLI
- Verwendung DescribeInstancePatchStatesForPatchGroup mit einem AWS SDK oder CLI
- Verwendung DescribeInstancePatches mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindowExecutionTaskInvocations mit einem AWS SDK oder CLI

- Verwendung DescribeMaintenanceWindowExecutionTasks mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindowExecutions mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindowTargets mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindowTasks mit einem AWS SDK oder CLI
- Verwendung DescribeMaintenanceWindows mit einem AWS SDK oder CLI
- Verwendung DescribeOpsItems mit einem AWS SDK oder CLI
- Verwendung DescribeParameters mit einem AWS SDK oder CLI
- Verwendung DescribePatchBaselines mit einem AWS SDK oder CLI
- Verwendung DescribePatchGroupState mit einem AWS SDK oder CLI
- Verwendung DescribePatchGroups mit einem AWS SDK oder CLI
- Verwendung GetAutomationExecution mit einem AWS SDK oder CLI
- Verwendung GetCommandInvocation mit einem AWS SDK oder CLI
- Verwendung GetConnectionStatus mit einem AWS SDK oder CLI
- Verwendung GetDefaultPatchBaseline mit einem AWS SDK oder CLI
- Verwendung GetDeployablePatchSnapshotForInstance mit einem AWS SDK oder CLI
- Verwendung GetDocument mit einem AWS SDK oder CLI
- Verwendung GetInventory mit einem AWS SDK oder CLI
- Verwendung GetInventorySchema mit einem AWS SDK oder CLI
- Verwendung GetMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung GetMaintenanceWindowExecution mit einem AWS SDK oder CLI
- Verwendung GetMaintenanceWindowExecutionTask mit einem AWS SDK oder CLI
- Verwendung GetParameterHistory mit einem AWS SDK oder CLI
- Verwendung GetParameters mit einem AWS SDK oder CLI
- Verwendung GetPatchBaseline mit einem AWS SDK oder CLI
- Verwendung GetPatchBaselineForPatchGroup mit einem AWS SDK oder CLI
- Verwendung ListAssociationVersions mit einem AWS SDK oder CLI
- Verwendung ListAssociations mit einem AWS SDK oder CLI
- Verwendung ListCommandInvocations mit einem AWS SDK oder CLI
- Verwendung ListCommands mit einem AWS SDK oder CLI
- Verwendung ListComplianceItems mit einem AWS SDK oder CLI

- Verwendung ListComplianceSummaries mit einem AWS SDK oder CLI
- Verwendung ListDocumentVersions mit einem AWS SDK oder CLI
- Verwendung ListDocuments mit einem AWS SDK oder CLI
- Verwendung ListInventoryEntries mit einem AWS SDK oder CLI
- Verwendung ListResourceComplianceSummaries mit einem AWS SDK oder CLI
- Verwendung ListTagsForResource mit einem AWS SDK oder CLI
- Verwendung ModifyDocumentPermission mit einem AWS SDK oder CLI
- Verwendung PutComplianceItems mit einem AWS SDK oder CLI
- Verwendung PutInventory mit einem AWS SDK oder CLI
- Verwendung PutParameter mit einem AWS SDK oder CLI
- Verwendung RegisterDefaultPatchBaseline mit einem AWS SDK oder CLI
- Verwendung RegisterPatchBaselineForPatchGroup mit einem AWS SDK oder CLI
- Verwendung RegisterTargetWithMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung RegisterTaskWithMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung RemoveTagsFromResource mit einem AWS SDK oder CLI
- Verwendung SendCommand mit einem AWS SDK oder CLI
- Verwendung StartAutomationExecution mit einem AWS SDK oder CLI
- Verwendung StopAutomationExecution mit einem AWS SDK oder CLI
- Verwendung UpdateAssociation mit einem AWS SDK oder CLI
- Verwendung UpdateAssociationStatus mit einem AWS SDK oder CLI
- Verwendung UpdateDocument mit einem AWS SDK oder CLI
- Verwendung UpdateDocumentDefaultVersion mit einem AWS SDK oder CLI
- Verwendung UpdateMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung UpdateManagedInstanceRole mit einem AWS SDK oder CLI
- Verwendung UpdateOpsItem mit einem AWS SDK oder CLI
- Verwendung UpdatePatchBaseline mit einem AWS SDK oder CLI

Verwendung **`AddTagsToResource`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`AddTagsToResource`.

CLI

AWS CLI

Beispiel 1: Um einem Wartungsfenster Tags hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel wird dem angegebenen Wartungsfenster ein Tag hinzugefügt.

```
aws ssm add-tags-to-resource \
    --resource-type "MaintenanceWindow" \
    --resource-id "mw-03eb9db428EXAMPLE" \
    --tags "Key=Stack,Value=Production"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um einem Parameter Tags hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel werden dem angegebenen Parameter zwei Tags hinzugefügt.

```
aws ssm add-tags-to-resource \
    --resource-type "Parameter" \
    --resource-id "My-Parameter" \
    --tags '[{"Key":"Region","Value":"East"},{"Key":"Environment",
 "Value":"Production"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 3: Um einem SSM-Dokument Tags hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel wird dem angegebenen Dokument ein Tag hinzugefügt.

```
aws ssm add-tags-to-resource \
    --resource-type "Document" \
    --resource-id "My-Document" \
    --tags "Key=Quarter,Value=Q322"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Tagging Systems Manager Manager-Ressourcen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter AddTagsToResource AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Wartungsfenster mit neuen Tags aktualisiert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$option1 = @{Key="Stack";Value=@("Production")}
Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $option1
```

Beispiel 2: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jedes Tag zu erstellen. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
$tag1 = New-Object Amazon.SimpleSystemsManagement.Model.Tag
$tag1.Key = "Stack"
$tag1.Value = "Production"

Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $tag1
```

- Einzelheiten zur API finden Sie unter AddTagsToResource AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CancelCommand`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CancelCommand`.

CLI

AWS CLI

Beispiel 1: Um einen Befehl für alle Instanzen abzubrechen

Im folgenden `cancel-command` Beispiel wird versucht, den angegebenen Befehl abzubrechen, der bereits für alle Instanzen ausgeführt wird.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um einen Befehl für bestimmte Instanzen abzubrechen

Im folgenden `cancel-command` Beispiel wird versucht, einen Befehl nur für die angegebene Instanz abzubrechen.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
    --instance-ids "i-02573cafcfEXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Tagging Systems Manager Manager-Parameter im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter CancelCommand AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird versucht, einen Befehl abzubrechen. Wenn der Vorgang erfolgreich ist, erfolgt keine Ausgabe.

```
Stop-SSMCommand -CommandId "9ded293e-e792-4440-8e3e-7b8ec5feaa38"
```

- Einzelheiten zur API finden Sie unter CancelCommand AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateActivation`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreateActivation`.

CLI

AWS CLI

Um eine verwaltete Instanzaktivierung zu erstellen

Im folgenden `create-activation` Beispiel wird eine verwaltete Instanzaktivierung erstellt.

```
aws ssm create-activation \
    --default-instance-name "HybridWebServers" \
    --iam-role "HybridWebServersRole" \
    --registration-limit 5
```

Ausgabe:

```
{
    "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
    "ActivationCode": "dRmgnYaFv567vEXAMPLE"
}
```

Weitere Informationen finden Sie unter Schritt 4: Aktivierung einer verwalteten Instanz für eine Hybridumgebung erstellen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter CreateActivation AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine verwaltete Instanz erstellt.

```
New-SSMActivation -DefaultInstanceName "MyWebServers" -IamRole
 "SSMAutomationRole" -RegistrationLimit 10
```

Ausgabe:

```
ActivationCode       ActivationId
```

```
--------------       ------------
KWChhOxBTiwDcKE9BlKC 08e51e79-1e36-446c-8e63-9458569c1363
```

- Einzelheiten zur API finden Sie unter CreateActivation AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateAssociation`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreateAssociation`.

CLI

AWS CLI

Beispiel 1: Um ein Dokument mithilfe von Instanz-IDs zuzuordnen

In diesem Beispiel wird mithilfe von Instanz-IDs ein Konfigurationsdokument einer Instanz zugeordnet.

```
aws ssm create-association \
    --instance-id "i-0cb2b964d3e14fd9f" \
    --name "AWS-UpdateSSMAgent"
```

Ausgabe:

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

Weitere Informationen finden Sie CreateAssociationin der AWS Systems Manager API-Referenz.

Beispiel 2: So verknüpfen Sie ein Dokument mithilfe von Zielen

In diesem Beispiel wird mithilfe von Zielen ein Konfigurationsdokument einer Instanz zugeordnet.

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"
```

Ausgabe:

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

Weitere Informationen finden Sie CreateAssociationin der AWS Systems Manager API-Referenz.

Beispiel 3: So erstellen Sie eine Assoziation, die nur einmal ausgeführt wird

In diesem Beispiel wird eine neue Assoziation erstellt, die nur einmal am angegebenen Datum und zu der angegebenen Uhrzeit ausgeführt wird. Verknüpfungen, die mit einem Datum in der Vergangenheit oder Gegenwart erstellt wurden (zum Zeitpunkt der Verarbeitung liegt das Datum in der Vergangenheit), werden sofort ausgeführt.

```
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
```

Ausgabe:

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

Weitere Informationen finden Sie CreateAssociationin der AWS Systems Manager API-Referenz oder Referenz: Cron- und Rate-Ausdrücke für Systems Manager im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter CreateAssociation AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Konfigurationsdokument mithilfe von Instanz-IDs einer Instanz zugeordnet.

```
New-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
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 :
```

Beispiel 2: In diesem Beispiel wird mithilfe von Zielen ein Konfigurationsdokument einer Instanz zugeordnet.

```
$target = @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
New-SSMAssociation -Name "AWS-UpdateSSMAgent" -Target $target
```

Ausgabe:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           :
Date                 : 3/1/2017 6:22:21 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

Beispiel 3: In diesem Beispiel wird ein Konfigurationsdokument mithilfe von Zielen und Parametern einer Instanz zugeordnet.

```
$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
```

Ausgabe:

```
Name                 : Configure-CloudWatch
InstanceId           :
Date                 : 5/17/2018 3:17:44 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

Beispiel 4: In diesem Beispiel wird eine Assoziation mit allen Instanzen in der Region erstellt, mit**`AWS-GatherSoftwareInventory`**. Außerdem werden benutzerdefinierte Dateien und Registrierungsverzeichnisse in den zu erfassenden Parametern bereitgestellt

```
$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)"
```

Ausgabe:

```
Name                 : AWS-GatherSoftwareInventory
InstanceId           :
Date                 : 6/9/2019 8:57:56 AM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

- Einzelheiten zur API finden Sie unter CreateAssociation AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateAssociationBatch`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreateAssociationBatch`.

CLI

AWS CLI

Um mehrere Verknüpfungen zu erstellen

In diesem Beispiel wird ein Konfigurationsdokument mehreren Instanzen zugeordnet. Die Ausgabe gibt gegebenenfalls eine Liste mit erfolgreichen und fehlgeschlagenen Vorgängen zurück.

Befehl:

```
aws ssm create-association-batch --entries "Name=AWS-
UpdateSSMAgent,InstanceId=i-1234567890abcdef0" "Name=AWS-
UpdateSSMAgent,InstanceId=i-9876543210abcdef0"
```

Ausgabe:

```
{
  "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": []
}
```

- Einzelheiten zur API finden Sie unter CreateAssociationBatch in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Konfigurationsdokument mehreren Instanzen zugeordnet. Die Ausgabe gibt gegebenenfalls eine Liste mit erfolgreichen und fehlgeschlagenen Vorgängen zurück.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
New-SSMAssociationFromBatch -Entry $option1,$option2
```

Ausgabe:

```
Failed  Successful
------  ----------
{}      {Amazon.SimpleSystemsManagement.Model.FailedCreateAssociation,
 Amazon.SimpleSystemsManagement.Model.FailedCreateAsso...
```

Beispiel 2: In diesem Beispiel werden alle Details eines erfolgreichen Vorgangs angezeigt.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
(New-SSMAssociationFromBatch -Entry $option1,$option2).Successful
```

- Einzelheiten zur API finden Sie unter CreateAssociationBatch in AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateDocument`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreateDocument`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit Systems Manager

CLI

AWS CLI

Um ein Dokument zu erstellen

Im folgenden `create-document` Beispiel wird ein Systems Manager Manager-Dokument erstellt.

```
aws ssm create-document \
    --content file://exampleDocument.yml \
```

```
    --name "Example" \
    --document-type "Automation" \
    --document-format YAML
```

Ausgabe:

```
{
    "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": []
```

```
    }
}
```

Weitere Informationen finden Sie unter Erstellen von Systems Manager Manager-Dokumenten im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie CreateDocumentunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    // Create an AWS SSM document to use in this scenario.
    public static void createSSMDoc(SsmClient ssmClient, String docName) {
        // Create JSON for the content
        String jsonData = """
            {
                "schemaVersion": "2.2",
                "description": "Run a simple shell command",
                "mainSteps": [
                    {
                        "action": "aws:runShellScript",
                        "name": "runEchoCommand",
                        "inputs": {
                          "runCommand": [
                            "echo 'Hello, world!'"
                          ]
                        }
                      }
                    ]
                }
            """;

        try {
            CreateDocumentRequest request = CreateDocumentRequest.builder()
                .content(jsonData)
```

```
                .name(docName)
                .documentType(DocumentType.COMMAND)
                .build();

            // Create the document.
            CreateDocumentResponse response = ssmClient.createDocument(request);
            System.out.println("The status of the document is " +
 response.documentDescription().status());

        } catch (DocumentAlreadyExistsException e) {
            System.err.println("The document already exists. Moving on." );
        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie CreateDocumentin der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Dokument in Ihrem Konto erstellt. Das Dokument muss im JSON-Format sein. Weitere Informationen zum Schreiben eines Konfigurationsdokuments finden Sie unter Konfigurationsdokument in der SSM-API-Referenz.

```
New-SSMDocument -Content (Get-Content -Raw "c:\temp\RunShellScript.json") -Name
 "RunShellScript" -DocumentType "Command"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter CreateDocument AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateMaintenanceWindow`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreateMaintenanceWindow`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit Systems Manager

CLI

AWS CLI

Beispiel 1: Um ein Wartungsfenster zu erstellen

Das folgende `create-maintenance-window` Beispiel erstellt ein neues Wartungsfenster, das alle fünf Minuten für bis zu zwei Stunden (je nach Bedarf) alle fünf Minuten erstellt, verhindert, dass neue Aufgaben innerhalb einer Stunde nach dem Ende der Ausführung des Wartungsfensters gestartet werden, nicht zugeordnete Ziele (Instanzen, die Sie nicht für das Wartungsfenster registriert haben) zulässt und durch die Verwendung von benutzerdefinierten Tags anzeigt, dass der Ersteller beabsichtigt, es in einem Tutorial zu verwenden.

```
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"
```

Ausgabe:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE"
}
```

Beispiel 2: Um ein Wartungsfenster zu erstellen, das nur einmal ausgeführt wird

Im folgenden `create-maintenance-window` Beispiel wird ein neues Wartungsfenster erstellt, das nur einmal am angegebenen Datum und zur angegebenen Uhrzeit ausgeführt wird.

```
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"
```

Ausgabe:

```
{
    "WindowId": "mw-01234567890abcdef"
}
```

Weitere Informationen finden Sie unter Maintenance Windows im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter CreateMaintenanceFenster in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static String createMaintenanceWindow(SsmClient ssmClient, String
winName) {
       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();

       try {
           CreateMaintenanceWindowResponse response =
ssmClient.createMaintenanceWindow(request);
           String maintenanceWindowId = response.windowId();
           System.out.println("The maintenance window id is " +
maintenanceWindowId);
           return maintenanceWindowId;

       } catch (DocumentAlreadyExistsException e) {
           System.err.println("The maintenance window already exists. Moving
on.");
       } catch (SsmException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }

       MaintenanceWindowFilter filter = MaintenanceWindowFilter.builder()
           .key("name")
           .values(winName)
           .build();
```

```
        DescribeMaintenanceWindowsRequest winRequest =
DescribeMaintenanceWindowsRequest.builder()
            .filters(filter)
            .build();

        String windowId = "";
        DescribeMaintenanceWindowsResponse response =
ssmClient.describeMaintenanceWindows(winRequest);
        List<MaintenanceWindowIdentity> windows = response.windowIdentities();
        if (!windows.isEmpty()) {
            windowId = windows.get(0).windowId();
            System.out.println("Window ID: " + windowId);
        } else {
            System.out.println("Window not found.");
        }
       return windowId;
   }
```

- Einzelheiten zur API finden Sie unter CreateMaintenanceFenster in der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein neues Wartungsfenster mit dem angegebenen Namen erstellt, das an jedem Dienstag um 16 Uhr für 4 Stunden läuft, mit einem Grenzwert von 1 Stunde, und das Ziele ohne Zuordnung zulässt.

```
New-SSMMaintenanceWindow -Name "MyMaintenanceWindow" -Duration 4 -Cutoff 1 -
AllowUnassociatedTarget $true -Schedule "cron(0 16 ? * TUE *)"
```

Ausgabe:

```
mw-03eb53e1ea7383998
```

- Einzelheiten zur API finden Sie unter CreateMaintenanceWindow in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateOpsItem`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreateOpsItem`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit Systems Manager

CLI

AWS CLI

Um eine zu erstellen OpsItems

Im folgenden `create-ops-item` Beispiel wird der Schlüssel /aws/resources verwendet, OperationalData um eine OpsItem mit einer Amazon DynamoDB DynamoDB-bezogene Ressource zu erstellen.

```
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"
```

Ausgabe:

```
{
    "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

Weitere Informationen finden Sie unter Creating OpsItems im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter CreateOpsElement in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   // Create an SSM OpsItem
   public static String createSSMOpsItem(SsmClient ssmClient, String title,
String source, String category, String severity) {
       try {
           CreateOpsItemRequest opsItemRequest = CreateOpsItemRequest.builder()
               .description("Created by the Systems Manager Java API")
               .title(title)
               .source(source)
               .category(category)
               .severity(severity)
               .build();

           CreateOpsItemResponse itemResponse =
ssmClient.createOpsItem(opsItemRequest);
           return itemResponse.opsItemId();

       } catch (SsmException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
      return "";
   }
```

- Einzelheiten zur API finden Sie unter CreateOpsArtikel in der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterSystems Manager mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreatePatchBaseline`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreatePatchBaseline`.

CLI

AWS CLI

Beispiel 1: So erstellen Sie eine Patch-Baseline mit automatischer Genehmigung

Im folgenden `create-patch-baseline` Beispiel wird eine Patch-Baseline für Windows Server erstellt, die Patches für eine Produktionsumgebung sieben Tage nach ihrer Veröffentlichung durch Microsoft genehmigt.

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
    --operating-system "WINDOWS" \
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Import
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},App
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

Ausgabe:

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Beispiel 2: So erstellen Sie eine Patch-Baseline mit einem Stichtag für die Genehmigung

Im folgenden `create-patch-baseline` Beispiel wird eine Patch-Baseline für Windows Server erstellt, die alle Patches für eine Produktionsumgebung genehmigt, die am oder vor dem 7. Juli 2020 veröffentlicht wurden.

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
```

```
    --operating-system "WINDOWS" \
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Import
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},App
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

Ausgabe:

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Beispiel 3: So erstellen Sie eine Patch-Baseline mit Genehmigungsregeln, die in einer JSON-Datei gespeichert sind

Im folgenden `create-patch-baseline` Beispiel wird eine Patch-Baseline für Amazon Linux 2017.09 erstellt, die Patches für eine Produktionsumgebung sieben Tage nach ihrer Veröffentlichung genehmigt, Genehmigungsregeln für die Patch-Baseline festlegt und ein benutzerdefiniertes Repository für Patches festlegt.

```
aws ssm create-patch-baseline \
    --cli-input-json file://my-amazon-linux-approval-rules-and-repo.json
```

Inhalt von `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"
        }
    ]
}
```

Beispiel 4: Um eine Patch-Baseline zu erstellen, die genehmigte und abgelehnte Patches angibt

Im folgenden `create-patch-baseline` Beispiel werden Patches, die genehmigt und abgelehnt werden sollen, ausdrücklich als Ausnahme von den Standard-Genehmigungsregeln angegeben.

```
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"
```

Weitere Informationen finden Sie unter Erstellen einer benutzerdefinierten Patch-Baseline im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter CreatePatchBaseline in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline erstellt, die Patches sieben Tage nach ihrer Veröffentlichung durch Microsoft für verwaltete Instanzen genehmigt, auf denen Windows Server 2019 in einer Produktionsumgebung ausgeführt wird.

```
$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
```

Ausgabe:

```
pb-0z4z6221c4296b23z
```

- Einzelheiten zur API finden Sie unter CreatePatchBaseline in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteActivation`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteActivation`.

CLI

AWS CLI

Um eine verwaltete Instanzaktivierung zu löschen

Im folgenden `delete-activation` Beispiel wird eine verwaltete Instanzaktivierung gelöscht.

```
aws ssm delete-activation \
    --activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Setting Up AWS Systems Manager for Hybrid Environments im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeleteActivationunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Aktivierung gelöscht. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Remove-SSMActivation -ActivationId "08e51e79-1e36-446c-8e63-9458569c1363"
```

- Einzelheiten zur API finden Sie unter DeleteActivation AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteAssociation`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteAssociation`.

CLI

AWS CLI

Beispiel 1: Um eine Assoziation mithilfe der Zuordnungs-ID zu löschen

Im folgenden `delete-association` Beispiel wird die Assoziation für die angegebene Zuordnungs-ID gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
aws ssm delete-association \
```

```
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So löschen Sie eine Zuordnung

Im folgenden `delete-association` Beispiel wird die Verknüpfung zwischen einer Instanz und einem Dokument gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
aws ssm delete-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeleteAssociationunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die Verknüpfung zwischen einer Instanz und einem Dokument gelöscht. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Remove-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-
UpdateSSMAgent"
```

- Einzelheiten zur API finden Sie unter DeleteAssociation AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteDocument`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteDocument`.

CLI

AWS CLI

Um ein Dokument zu löschen

Im folgenden `delete-document` Beispiel wird ein Systems Manager Manager-Dokument gelöscht.

```
aws ssm delete-document \
    --name "Example"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Erstellen von Systems Manager Manager-Dokumenten im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeleteDocumentunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    // Deletes an AWS Systems Manager document.
    public static void deleteDoc(SsmClient ssmClient, String documentName) {
        try {
            DeleteDocumentRequest documentRequest =
DeleteDocumentRequest.builder()
                .name(documentName)
                .build();
```

```
            ssmClient.deleteDocument(documentRequest);
            System.out.println("The Systems Manager document was successfully
deleted.");

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie DeleteDocumentin der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Dokument gelöscht. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Remove-SSMDocument -Name "RunShellScript"
```

- Einzelheiten zur API finden Sie unter DeleteDocument AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteMaintenanceWindow`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteMaintenanceWindow`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit Systems Manager

CLI

AWS CLI

Um ein Wartungsfenster zu löschen

In diesem `delete-maintenance-window` Beispiel wird das angegebene Wartungsfenster entfernt.

```
aws ssm delete-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9"
```

Ausgabe:

```
{
    "WindowId":"mw-1a2b3c4d5e6f7g8h9"
}
```

Weitere Informationen finden Sie unter Löschen eines Wartungsfensters (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DeleteMaintenanceFenster in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void deleteMaintenanceWindow(SsmClient ssmClient, String winId)
{
       try {
           DeleteMaintenanceWindowRequest windowRequest =
DeleteMaintenanceWindowRequest.builder()
```

```
                .windowId(winId)
                .build();

            ssmClient.deleteMaintenanceWindow(windowRequest);
            System.out.println("The maintenance window was successfully
deleted.");

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie unter DeleteMaintenanceFenster in der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Wartungsfenster entfernt.

```
Remove-SSMMaintenanceWindow -WindowId "mw-06d59c1a07c022145"
```

Ausgabe:

```
mw-06d59c1a07c022145
```

- Einzelheiten zur API finden Sie unter DeleteMaintenanceFenster in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteParameter`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteParameter`.

CLI

AWS CLI

Um einen Parameter zu löschen

Im folgenden `delete-parameter` Beispiel wird der angegebene Einzelparameter gelöscht.

```
aws ssm delete-parameter \
    --name "MyParameter"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Arbeiten mit dem Parameterspeicher im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeleteParameterunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Parameter gelöscht. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Remove-SSMParameter -Name "helloWorld"
```

- Einzelheiten zur API finden Sie unter DeleteParameter AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeletePatchBaseline`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeletePatchBaseline`.

CLI

AWS CLI

Um eine Patch-Baseline zu löschen

Im folgenden `delete-patch-baseline` Beispiel wird die angegebene Patch-Baseline gelöscht.

```
aws ssm delete-patch-baseline \
    --baseline-id "pb-045f10b4f382baeda"
```

Ausgabe:

```
{
    "BaselineId": "pb-045f10b4f382baeda"
}
```

Weitere Informationen finden Sie unter Aktualisieren oder Löschen einer Patch-Baseline (Konsole) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DeletePatchBaseline in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline gelöscht.

```
Remove-SSMPatchBaseline -BaselineId "pb-045f10b4f382baeda"
```

Ausgabe:

```
pb-045f10b4f382baeda
```

- Einzelheiten zur API finden Sie unter DeletePatchBaseline in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeregisterManagedInstance`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeregisterManagedInstance`.

CLI

AWS CLI

Um die Registrierung einer verwalteten Instanz aufzuheben

Im folgenden `deregister-managed-instance` Beispiel wird die Registrierung der angegebenen verwalteten Instanz aufgehoben.

```
aws ssm deregister-managed-instance
    --instance-id "mi-08ab247cdfEXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Deregistering Managed Instances in a Hybrid Environment im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DeregisterManagedInstanz in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die Registrierung einer verwalteten Instanz aufgehoben. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Unregister-SSMManagedInstance -InstanceId "mi-08ab247cdf1046573"
```

- Einzelheiten zur API finden Sie unter DeregisterManagedInstance in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DeregisterPatchBaselineForPatchGroup** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeregisterPatchBaselineForPatchGroup`.

CLI

AWS CLI

Um eine Patch-Gruppe von einer Patch-Baseline abzumelden

Im folgenden `deregister-patch-baseline-for-patch-group` Beispiel wird die Registrierung der angegebenen Patchgruppe von der angegebenen Patch-Baseline aufgehoben.

```
aws ssm deregister-patch-baseline-for-patch-group \
    --patch-group "Production" \
    --baseline-id "pb-0ca44a362fEXAMPLE"
```

Ausgabe:

```
{
  "PatchGroup":"Production",
  "BaselineId":"pb-0ca44a362fEXAMPLE"
}
```

Weitere Informationen finden Sie unter Hinzufügen einer Patchgruppe zu einer Patch-Baseline im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeregisterPatchBaselineForPatchGroupunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die Registrierung einer Patchgruppe von einer Patch-Baseline aufgehoben.

```
Unregister-SSMPatchBaselineForPatchGroup -BaselineId "pb-045f10b4f382baeda" -
PatchGroup "Production"
```

Ausgabe:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Einzelheiten zur API finden Sie unter DeregisterPatchBaselineForPatchGroup AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeregisterTargetFromMaintenanceWindow`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeregisterTargetFromMaintenanceWindow`.

CLI

AWS CLI

Um ein Ziel aus einem Wartungsfenster zu entfernen

Im folgenden `deregister-target-from-maintenance-window` Beispiel wird das angegebene Ziel aus dem angegebenen Wartungsfenster entfernt.

```
aws ssm deregister-target-from-maintenance-window \
```

```
    --window-id "mw-ab12cd34ef56gh78" \
    --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

Ausgabe:

```
{
    "WindowId":"mw-ab12cd34ef56gh78",
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Weitere Informationen finden Sie unter Aktualisieren eines Wartungsfensters (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DeregisterTargetFromMaintenanceFenster in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Ziel aus einem Wartungsfenster entfernt.

```
Unregister-SSMTargetFromMaintenanceWindow -WindowTargetId
 "6ab5c208-9fc4-4697-84b7-b02a6cc25f7d" -WindowId "mw-06cf17cbefcb4bf4f"
```

Ausgabe:

```
WindowId             WindowTargetId
--------             --------------
mw-06cf17cbefcb4bf4f 6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

- Einzelheiten zur API finden Sie unter DeregisterTargetFromMaintenanceFenster in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DeregisterTaskFromMaintenanceWindow** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeregisterTaskFromMaintenanceWindow`.

CLI

AWS CLI

Um eine Aufgabe aus einem Wartungsfenster zu entfernen

Im folgenden `deregister-task-from-maintenance-window` Beispiel wird die angegebene Aufgabe aus dem angegebenen Wartungsfenster entfernt.

```
aws ssm deregister-task-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

Ausgabe:

```
{
    "WindowTaskId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",
    "WindowId":"mw-ab12cd34ef56gh78"
}
```

Weitere Informationen finden Sie unter Systems Manager Maintenance Windows Tutorials (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DeregisterTaskFromMaintenanceFenster in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Aufgabe aus einem Wartungsfenster entfernt.

```
Unregister-SSMTaskFromMaintenanceWindow -WindowTaskId "f34a2c47-ddfd-4c85-
a88d-72366b69af1b" -WindowId "mw-03a342e62c96d31b0"
```

Ausgabe:

```
WindowId             WindowTaskId
--------             ------------
mw-03a342e62c96d31b0 f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

- Einzelheiten zur API finden Sie unter DeregisterTaskFromMaintenanceFenster in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeActivations`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeActivations`.

CLI

AWS CLI

Um Aktivierungen zu beschreiben

Das folgende `describe-activations` Beispiel listet Details zu den Aktivierungen in Ihrem AWS Konto auf.

```
aws ssm describe-activations
```

Ausgabe:

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

Weitere Informationen finden Sie unter Schritt 4: Aktivierung einer verwalteten Instanz für eine Hybridumgebung erstellen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeActionsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieses Beispiel enthält Details zu den Aktivierungen in Ihrem Konto.

```
Get-SSMActivation
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeActivations AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAssociation`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAssociation`.

CLI

AWS CLI

Beispiel 1: Um Details zu einer Assoziation abzurufen

Das folgende `describe-association` Beispiel beschreibt die Assoziation für die angegebene Zuordnungs-ID.

```
aws ssm describe-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Ausgabe:

```
{
    "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"
    }
}
```

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So rufen Sie Details zu einer Zuordnung für eine bestimmte Instanz und ein bestimmtes Dokument ab

Das folgende `describe-association` Beispiel beschreibt die Zuordnung zwischen einer Instanz und einem Dokument.

```
aws ssm describe-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

Ausgabe:

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

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeAssociationunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieses Beispiel beschreibt die Assoziation zwischen einer Instanz und einem Dokument.

```
Get-SSMAssociation -InstanceId "i-0000293ffd8c57862" -Name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeAssociation AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAssociationExecutionTargets`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAssociationExecutionTargets`.

CLI

AWS CLI

Um Details zur Ausführung einer Assoziation abzurufen

Das folgende `describe-association-execution-targets` Beispiel beschreibt die angegebene Assoziationsausführung.

```
aws ssm describe-association-execution-targets \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"
```

Ausgabe:

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

Weitere Informationen finden Sie unter Zuordnungsverläufe anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeAssociationExecutionTargets AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Ressourcen-ID und ihr Ausführungsstatus angezeigt, die Teil der Ausführungsziele der Assoziation sind

```
Get-SSMAssociationExecutionTarget -AssociationId 123a45a0-
c678-9012-3456-78901234db5e -ExecutionId 123a45a0-c678-9012-3456-78901234db5e |
 Select-Object ResourceId, Status
```

Ausgabe:

```
ResourceId           Status
----------           ------
i-0b1b2a3456f7a890b  Success
i-01c12a45d6fc7a89f  Success
i-0a1caf234f56d7dc8  Success
i-012a3fd45af6dbcfe  Failed
i-0ddc1df23c4a5fb67  Success
```

Beispiel 2: Dieser Befehl überprüft die jeweilige Ausführung einer bestimmten Automatisierung seit gestern, der ein Befehlsdokument zugeordnet ist. Außerdem wird geprüft, ob die Ausführung der Assoziation fehlgeschlagen ist, und wenn ja, werden die Details zum Befehlsaufruf für die Ausführung zusammen mit der Instanz-ID angezeigt

```
$AssociationExecution= Get-SSMAssociationExecutionTarget -
AssociationId 1c234567-890f-1aca-a234-5a678d901cb0 -ExecutionId
 12345ca12-3456-2345-2b45-23456789012 |
    Where-Object {$_.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
    }
}
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeAssociationExecutionTargets AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAssociationExecutions`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAssociationExecutions`.

CLI

AWS CLI

Beispiel 1: Um Details zu allen Ausführungen für eine Assoziation abzurufen

Das folgende `describe-association-executions` Beispiel beschreibt alle Ausführungen der angegebenen Assoziation.

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Ausgabe:

```
{
    "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}"
        },
        ...
    ]
}
```

Weitere Informationen finden Sie unter Anzeigen von Zuordnungsverläufen im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: Um Details zu allen Ausführungen für eine Zuordnung nach einem bestimmten Datum und einer bestimmten Uhrzeit abzurufen

Das folgende `describe-association-executions` Beispiel beschreibt alle Ausführungen einer Assoziation nach dem angegebenen Datum und der angegebenen Uhrzeit.

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

Ausgabe:

```
{
    "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}"
        },
        ...
    ]
}
```

Weitere Informationen finden Sie unter Anzeigen von Zuordnungsverläufen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeAssociationAusführungen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Ausführungen für die angegebene Zuordnungs-ID zurückgegeben

```
Get-SSMAssociationExecution -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeAssociationAusführungen in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAutomationExecutions`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAutomationExecutions`.

CLI

AWS CLI

Um eine Automatisierungsausführung zu beschreiben

Im folgenden `describe-automation-executions` Beispiel werden Details zu einer Automatisierungsausführung angezeigt.

```
aws ssm describe-automation-executions \
    --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Ausgabe:

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

Weitere Informationen finden Sie unter Running a Simple Automation Workflow im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeAutomationAusführungen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle aktiven und beendeten Automatisierungsausführungen beschrieben, die mit Ihrem Konto verknüpft sind.

```
Get-SSMAutomationExecutionList
```

Ausgabe:

```
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]]}
```

Beispiel 2: In diesem Beispiel werden die Ausführungs-ID, das Dokument und der Start-/Endzeitstempel der Ausführung für Ausführungen angezeigt, bei denen es sich nicht um „Erfolg“ handelt AutomationExecutionStatus

```
Get-SSMAutomationExecutionList | Where-Object AutomationExecutionStatus
 -ne "Success" | Select-Object AutomationExecutionId, DocumentName,
 AutomationExecutionStatus, ExecutionStartTime, ExecutionEndTime | Format-Table -
AutoSize
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter Executions in Cmdlet Reference. DescribeAutomation AWS Tools for PowerShell

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAutomationStepExecutions`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAutomationStepExecutions`.

CLI

AWS CLI

Beispiel 1: Um alle Schritte für eine Automatisierungsausführung zu beschreiben

Das folgende `describe-automation-step-executions` Beispiel zeigt Details zu den Schritten einer Automatisierungsausführung.

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Ausgabe:

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

Beispiel 2: Um einen bestimmten Schritt für eine Automatisierungsausführung zu beschreiben

Das folgende `describe-automation-step-executions` Beispiel zeigt Details zu einem bestimmten Schritt einer Automatisierungsausführung.

```
aws ssm describe-automation-step-executions \
```

```
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
    --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE
```

Weitere Informationen finden Sie unter Schrittweises Ausführen eines Automatisierungsworkflows (Befehlszeile) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeAutomationStepExecutionsunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden Informationen über alle aktiven und beendeten Schrittausführungen in einem Automatisierungs-Workflow angezeigt.

```
Get-SSMAutomationStepExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object StepName, Action, StepStatus
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeAutomationStepExecutions AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAvailablePatches`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAvailablePatches`.

CLI

AWS CLI

Um verfügbare Patches zu erhalten

Im folgenden `describe-available-patches` Beispiel werden Details zu allen verfügbaren Patches für Windows Server 2019 abgerufen, die den MSRC-Schweregrad Kritisch haben.

```
aws ssm describe-available-patches \
    --filters "Key=PRODUCT,Values=WindowsServer2019"
 "Key=MSRC_SEVERITY,Values=Critical"
```

Ausgabe:

```
{
    "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"
        },
        ...
    ]
}
```

Um Details zu einem bestimmten Patch abzurufen

Im folgenden `describe-available-patches` Beispiel werden Details zum angegebenen Patch abgerufen.

```
aws ssm describe-available-patches \
    --filters "Key=PATCH_ID,Values=KB4480979"
```

Ausgabe:

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

Weitere Informationen finden Sie unter So funktionieren Patch Manager-Operationen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeAvailablePatches in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle verfügbaren Patches für Windows Server 2012 abgerufen, die den MSRC-Schweregrad Kritisch haben. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="PRODUCT";Values=@("WindowsServer2012")}
$filter2 = @{Key="MSRC_SEVERITY";Values=@("Critical")}

Get-SSMAvailablePatch -Filter $filter1,$filter2
```

Ausgabe:

```
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
...
```

Beispiel 2: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jeden Filter zu erstellen.

```
$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
```

Beispiel 3: In diesem Beispiel werden alle Updates abgerufen, die in den letzten 20 Tagen veröffentlicht wurden und für Produkte gelten, die 2019 entsprechen WindowsServer

```
Get-SSMAvailablePatch | Where-Object ReleaseDate -ge (Get-Date).AddDays(-20)
 | Where-Object Product -eq "WindowsServer2019" | Select-Object ReleaseDate,
 Product, Title
```

Ausgabe:

```
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)
```

- Einzelheiten zur API finden Sie unter DescribeAvailablePatches in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeDocument`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeDocument`.

CLI

AWS CLI

Um Details eines Dokuments anzuzeigen

Im folgenden `describe-document` Beispiel werden Details zu einem Systems Manager Manager-Dokument in Ihrem AWS Konto angezeigt.

```
aws ssm describe-document \
    --name "Example"
```

Ausgabe:

```
{
    "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": []
    }
}
```

Weitere Informationen finden Sie unter Erstellen von Systems Manager Manager-Dokumenten im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeDocumentunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden Informationen zu einem Dokument zurückgegeben.

```
Get-SSMDocumentDescription -Name "RunShellScript"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeDocument AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeDocumentPermission`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeDocumentPermission`.

CLI

AWS CLI

Um die Berechtigungen für Dokumente zu beschreiben

Im folgenden `describe-document-permission` Beispiel werden Berechtigungsdetails zu einem Systems Manager Manager-Dokument angezeigt, das öffentlich geteilt wird.

```
aws ssm describe-document-permission \
    --name "Example" \
    --permission-type "Share"
```

Ausgabe:

```
{
    "AccountIds": [
        "all"
    ],
    "AccountSharingInfoList": [
        {
            "AccountId": "all",
            "SharedDocumentVersion": "$DEFAULT"
        }
    ]
}
```

Weitere Informationen finden Sie unter Freigeben eines Systems Manager Manager-Dokuments im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeDocumentBerechtigungen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle Versionen eines Dokuments aufgeführt.

```
Get-SSMDocumentVersionList -Name "RunShellScript"
```

Ausgabe:

```
CreatedDate          DocumentVersion IsDefaultVersion Name
-----------          --------------- ---------------- ----
2/24/2017 5:25:13 AM 1               True             RunShellScript
```

- Einzelheiten zur API finden Sie unter DescribeDocumentPermission in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeEffectiveInstanceAssociations`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeEffectiveInstanceAssociations`.

CLI

AWS CLI

Um Details zu den effektiven Verknüpfungen für eine Instanz abzurufen

Im folgenden `describe-effective-instance-associations` Beispiel werden Details zu den effektiven Verknüpfungen für eine Instance abgerufen.

Befehl:

```
aws ssm describe-effective-instance-associations --instance-id
 "i-1234567890abcdef0"
```

Ausgabe:

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

- Einzelheiten zur API finden Sie unter DescribeEffectiveInstanceAssociations AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die effektiven Verknüpfungen für eine Instanz beschrieben.

```
Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5
```

Ausgabe:

```
AssociationId                        Content
-------------                        -------
d8617c07-2079-4c18-9847-1655fc2698b0 {...
```

Beispiel 2: In diesem Beispiel wird der Inhalt der effektiven Verknüpfungen für eine Instanz angezeigt.

```
(Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5).Content
```

Ausgabe:

```
{
```

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

- Einzelheiten zur API finden Sie unter DescribeEffectiveInstanceAssociations AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DescribeEffectivePatchesForPatchBaseline** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeEffectivePatchesForPatchBaseline`.

CLI

AWS CLI

Beispiel 1: Um alle Patches abzurufen, die durch eine benutzerdefinierte Patch-Baseline definiert sind

Im folgenden `describe-effective-patches-for-patch-baseline` Beispiel werden die durch eine benutzerdefinierte Patch-Baseline definierten Patches im aktuellen AWS Konto zurückgegeben. Beachten Sie, dass für eine benutzerdefinierte Baseline nur die ID für erforderlich ist`--baseline-id`.

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "pb-08b654cf9b9681f04"
```

Ausgabe:

```
{
    "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--"
}
```

Beispiel 2: Um alle Patches abzurufen, die durch eine AWS verwaltete Patch-Baseline definiert sind

Im folgenden `describe-effective-patches-for-patch-baseline` Beispiel werden die durch eine AWS verwaltete Patch-Baseline definierten Patches zurückgegeben. Beachten Sie, dass für eine AWS verwaltete Baseline der vollständige Baseline-ARN erforderlich ist für `--baseline-id`

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter So werden Sicherheitspatches ausgewählt im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeEffectivePatchesForPatchBaselineunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle Patch-Baselines mit einer maximalen Ergebnisliste von 1 aufgeführt.

```
Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1
```

Ausgabe:

```
Patch                                       PatchStatus
-----                                       -----------
Amazon.SimpleSystemsManagement.Model.Patch
 Amazon.SimpleSystemsManagement.Model.PatchStatus
```

Beispiel 2: In diesem Beispiel wird der Patchstatus für alle Patch-Baselines mit einer maximalen Ergebnisliste von 1 angezeigt.

```
(Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1).PatchStatus
```

Ausgabe:

```
ApprovalDate          DeploymentStatus
------------          ----------------
12/21/2010 6:00:00 PM APPROVED
```

- Einzelheiten zur API finden Sie unter DescribeEffectivePatchesForPatchBaseline AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeInstanceAssociationsStatus`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeInstanceAssociationsStatus`.

CLI

AWS CLI

Um den Status der Zuordnungen einer Instanz zu beschreiben

Dieses Beispiel zeigt Details zu den Zuordnungen für eine Instance.

Befehl:

```
aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"
```

Ausgabe:

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

- Einzelheiten zur API finden Sie DescribeInstanceAssociationsStatusunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieses Beispiel zeigt Details der Assoziationen für eine Instanz.

```
Get-SSMInstanceAssociationsStatus -InstanceId "i-0000293ffd8c57862"
```

Ausgabe:

```
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
```

Beispiel 2: In diesem Beispiel wird der Status der Instanzzuweisung für die angegebene Instanz-ID überprüft und außerdem der Ausführungsstatus dieser Zuordnungen angezeigt

```
Get-SSMInstanceAssociationsStatus -InstanceId i-012e3cb4df567e8aa | ForEach-
Object {Get-SSMAssociationExecution -AssociationId .AssociationId}
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeInstanceAssociationsStatus AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeInstanceInformation`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeInstanceInformation`.

CLI

AWS CLI

Beispiel 1: Um Informationen zu verwalteten Instanzen zu beschreiben

Im folgenden `describe-instance-information` Beispiel werden Details zu jeder Ihrer verwalteten Instanzen abgerufen.

```
aws ssm describe-instance-information
```

Beispiel 2: Um Informationen über eine bestimmte verwaltete Instanz zu beschreiben

Das folgende `describe-instance-information` Beispiel zeigt Details der verwalteten Instanz`i-028ea792daEXAMPLE`.

```
aws ssm describe-instance-information \
    --filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

Beispiel 3: Um Informationen über verwaltete Instanzen mit einem bestimmten Tag-Schlüssel zu beschreiben

Das folgende `describe-instance-information` Beispiel zeigt Details für verwaltete Instanzen, die über den Tag-Schlüssel verfügen`DEV`.

```
aws ssm describe-instance-information \
    --filters "Key=tag-key,Values=DEV"
```

Ausgabe:

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

Weitere Informationen finden Sie unter Verwaltete Instanzen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeInstanceInformationen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieses Beispiel zeigt Details zu jeder Ihrer Instanzen.

```
Get-SSMInstanceInformation
```

Ausgabe:

```
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
```

Beispiel 2: Dieses Beispiel zeigt, wie der Parameter -Filter verwendet wird, um Ergebnisse nur nach den AWS Systems Manager Manager-Instanzen in der Region **`us-east-1`** mit dem Wert **`AgentVersion`** von **`2.2.800.0`** zu filtern. Eine Liste der gültigen -Filter-Schlüsselwerte finden Sie im InstanceInformation API-Referenzthema (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
```

Ausgabe:

```
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
```

Beispiel 3: Dieses Beispiel zeigt, wie der InstanceInformationFilterList Parameter - verwendet wird, um Ergebnisse nur nach den AWS Systems Manager Manager-Instanzen in **`PlatformTypes`** der Region **`us-east-1`** mit **`Windows`** oder zu filtern**`Linux`**. Eine Liste der gültigen InstanceInformationFilterList Schlüsselwerte finden Sie im InstanceInformationFilter API-Referenzthema (https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_ InstanceInformation Filter.html).

```
$Filters = @{
   Key="PlatformTypes"
   ValueSet=("Windows","Linux")
}
Get-SSMInstanceInformation -Region us-east-1 -InstanceInformationFilterList
 $Filters
```

Ausgabe:

```
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
```

Beispiel 4: In diesem Beispiel werden von SSM verwaltete Instanzen und Exporte InstanceId LastPingDateTime sowie PlatformName in eine CSV-Datei aufgeführt. PingStatus

```
Get-SSMInstanceInformation | Select-Object InstanceId, PingStatus,
 LastPingDateTime, PlatformName | Export-Csv Instance-details.csv -
NoTypeInformation
```

- Einzelheiten zur API finden Sie unter DescribeInstanceInformationen in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeInstancePatchStates`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeInstancePatchStates`.

CLI

AWS CLI

Um die Status der Patch-Zusammenfassung für Instanzen abzurufen

In diesem `describe-instance-patch-states` Beispiel werden die Status der Patch-Zusammenfassung für eine Instanz abgerufen.

```
aws ssm describe-instance-patch-states \
    --instance-ids "i-1234567890abcdef0"
```

Ausgabe:

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

Weitere Informationen finden Sie unter About Patch Compliance im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeInstancePatchStatesunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Status der Patch-Zusammenfassung für eine Instanz abgerufen.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407"
```

Beispiel 2: In diesem Beispiel werden die Status der Patch-Zusammenfassung für zwei Instanzen abgerufen.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407","i-09a618aec652973a9"
```

- Einzelheiten zur API finden Sie unter DescribeInstancePatchStates AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DescribeInstancePatchStatesForPatchGroup** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeInstancePatchStatesForPatchGroup`.

CLI

AWS CLI

Beispiel 1: Um den Instanzstatus für eine Patch-Gruppe abzurufen

Im folgenden `describe-instance-patch-states-for-patch-group` Beispiel werden Details zu den Status der Patchzusammenfassung pro Instanz für die angegebene Patchgruppe abgerufen.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --patch-group "Production"
```

Ausgabe:

```
{
    "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
        }
    ]
}
```

Beispiel 2: Um den Instanzstatus für eine Patch-Gruppe mit mehr als fünf fehlenden Patches abzurufen

Im folgenden `describe-instance-patch-states-for-patch-group` Beispiel werden Details zum Status der Patch-Zusammenfassung für die angegebene Patchgruppe für Instances mit mehr als fünf fehlenden Patches abgerufen.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=MissingCount,Type=GreaterThan,Values=5 \
    --patch-group "Production"
```

Ausgabe:

```
{
    "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
        }
    ]
}
```

Beispiel 3: Um den Instanzstatus für eine Patchgruppe mit weniger als zehn Instanzen abzurufen, für die ein Neustart erforderlich ist

Im folgenden `describe-instance-patch-states-for-patch-group` Beispiel werden Details zum Status der Patch-Zusammenfassung für die angegebene Patchgruppe für Instances mit weniger als zehn Instanzen abgerufen, die einen Neustart erfordern.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \
    --patch-group "Production"
```

Ausgabe:

```
{
    "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
        }
    ]
}
```

Weitere Informationen finden Sie unter Grundlegendes zu den Werten für den Patch-Kompatibilitätsstatus im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeInstancePatchStatesForPatchGroup in AWS CLI Command Reference.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Status der Patch-Zusammenfassung pro Instanz für eine Patch-Gruppe abgerufen.

```
Get-SSMInstancePatchStatesForPatchGroup -PatchGroup "Production"
```

- API-Details finden Sie unter DescribeInstancePatchStatesForPatchGroup in AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeInstancePatches`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeInstancePatches`.

CLI

AWS CLI

Beispiel 1: Um die Details zum Patch-Status für eine Instanz abzurufen

Im folgenden `describe-instance-patches` Beispiel werden Details zu den Patches für die angegebene Instanz abgerufen.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0"
```

Ausgabe:

```
{
    "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--"
}
```

Beispiel 2: Um eine Liste von Patches mit dem Status Missing für eine Instanz abzurufen

Im folgenden `describe-instance-patches` Beispiel werden Informationen über Patches abgerufen, die sich für die angegebene Instanz im Status Missing befinden.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Missing
```

Ausgabe:

```
{
    "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--"
}
```

Weitere Informationen finden Sie unter About Patch Compliance States im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: Um eine Liste der Patches abzurufen, die seit einer bestimmten InstalledTime Instanz installiert wurden

Im folgenden `describe-instance-patches` Beispiel werden Informationen über Patches abgerufen, die seit einem bestimmten Zeitpunkt für die angegebene Instanz installiert wurden, indem die Verwendung von `--filters` und `--query` kombiniert wird.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Installed \
    --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"
```

Ausgabe:

```
{
    "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--"
}
```

- Einzelheiten zur API finden Sie unter DescribeInstancePatches in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Patch-Compliance-Details für eine Instanz abgerufen.

```
Get-SSMInstancePatch -InstanceId "i-08ee91c0b17045407"
```

- Einzelheiten zur API finden Sie unter Referenz zu DescribeInstancePatches in AWS Tools for PowerShell Cmdlets.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeMaintenanceWindowExecutionTaskInvocations`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeMaintenanceWindowExecutionTaskInvocations`.

CLI

AWS CLI

Um die spezifischen Aufgabenaufrufen für die Ausführung einer Aufgabe in einem Wartungsfenster auszuführen

Im folgenden `describe-maintenance-window-execution-task-invocations` Beispiel werden die Aufrufe für die angegebene Aufgabe aufgeführt, die im Rahmen der Ausführung des angegebenen Wartungsfensters ausgeführt wurden.

```
aws ssm describe-maintenance-window-execution-task-invocations \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"
```

Ausgabe:

```
{
    "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"
```

```
        }
    ]
}
```

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutionTaskInvocations AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Aufrufe für eine Aufgabe aufgeführt, die im Rahmen der Ausführung eines Wartungsfensters ausgeführt wurde.

```
Get-SSMMaintenanceWindowExecutionTaskInvocationList -TaskId "ac0c6ae1-
daa3-4a89-832e-d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Ausgabe:

```
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    :
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutionTaskInvocations AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeMaintenanceWindowExecutionTasks`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeMaintenanceWindowExecutionTasks`.

CLI

AWS CLI

Um alle Aufgaben aufzulisten, die mit der Ausführung eines Wartungsfensters verbunden sind

Das folgende `ssm describe-maintenance-window-execution-tasks` Beispiel listet die Aufgaben auf, die mit der Ausführung des angegebenen Wartungsfensters verknüpft sind.

```
aws ssm describe-maintenance-window-execution-tasks \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Ausgabe:

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

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutionAufgaben in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Aufgaben aufgeführt, die mit der Ausführung eines Wartungsfensters verbunden sind.

```
Get-SSMMaintenanceWindowExecutionTaskList -WindowExecutionId
 "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutionAufgaben in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeMaintenanceWindowExecutions`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeMaintenanceWindowExecutions`.

CLI

AWS CLI

Beispiel 1: Um alle Ausführungen für ein Wartungsfenster aufzulisten

Das folgende `describe-maintenance-window-executions` Beispiel listet alle Ausführungen für das angegebene Wartungsfenster auf.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE"
```

Ausgabe:

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

Beispiel 2: Um alle Ausführungen für ein Wartungsfenster vor einem bestimmten Datum aufzulisten

Im folgenden `describe-maintenance-window-executions` Beispiel werden alle Ausführungen für das angegebene Wartungsfenster vor dem angegebenen Datum aufgeführt.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"
```

Ausgabe:

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

Beispiel 3: Um alle Ausführungen für ein Wartungsfenster nach einem bestimmten Datum aufzulisten

Im folgenden `describe-maintenance-window-executions` Beispiel werden alle Ausführungen für das angegebene Wartungsfenster nach dem angegebenen Datum aufgeführt.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"
```

Ausgabe:

```
{
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
```

```
        "Status": "IN_PROGRESS",
        "StartTime": "2021-08-04T11:00:00.000000-07:00"
        }
    ]
}
```

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutions AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle Ausführungen für ein Wartungsfenster aufgeführt.

```
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d"
```

Ausgabe:

```
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
```

Beispiel 2: In diesem Beispiel werden alle Ausführungen für ein Wartungsfenster vor einem bestimmten Datum aufgeführt.

```
$option1 = @{Key="ExecutedBefore";Values=@("2016-11-04T05:00:00Z")}
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

Beispiel 3: In diesem Beispiel werden alle Ausführungen für ein Wartungsfenster nach einem bestimmten Datum aufgeführt.

```
$option1 = @{Key="ExecutedAfter";Values=@("2016-11-04T05:00:00Z")}
```

```
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutions AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeMaintenanceWindowTargets`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeMaintenanceWindowTargets`.

CLI

AWS CLI

Beispiel 1: Um alle Ziele für ein Wartungsfenster aufzulisten

Das folgende `describe-maintenance-window-targets` Beispiel listet alle Ziele für ein Wartungsfenster auf.

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-06cf17cbefEXAMPLE"
```

Ausgabe:

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

Beispiel 2: Um alle Ziele für ein Wartungsfenster aufzulisten, die einem bestimmten Besitzerinformationswert entsprechen

In diesem `describe-maintenance-window-targets` Beispiel werden alle Ziele für ein Wartungsfenster mit einem bestimmten Wert aufgeführt.

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-0ecb1226ddEXAMPLE" \
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

Ausgabe:

```
{
    "Targets": [
        {
            "WindowId": "mw-0ecb1226ddEXAMPLE",
            "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
```

```
                    "Key": "tag:Environment",
                    "Values": [
                        "Prod"
                    ]
                }
            ],
            "OwnerInformation": "CostCenter1",
            "Name": "ProdTarget1"
        }
    ]
}
```

Weitere Informationen finden Sie unter Informationen über Maintenance Windows (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeMaintenanceWindowTargetsunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle Ziele für ein Wartungsfenster aufgeführt.

```
Get-SSMMaintenanceWindowTarget -WindowId "mw-06cf17cbefcb4bf4f"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowTargets AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeMaintenanceWindowTasks`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeMaintenanceWindowTasks`.

CLI

AWS CLI

Beispiel 1: Um alle Aufgaben für ein Wartungsfenster aufzulisten

Das folgende `describe-maintenance-window-tasks` Beispiel listet alle Aufgaben für das angegebene Wartungsfenster auf.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-06cf17cbefEXAMPLE"
```

Ausgabe:

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

Beispiel 2: Um alle Aufgaben für ein Wartungsfenster aufzulisten, das das RunPowerShellScript Befehlsdokument AWS- aufruft

Das folgende `describe-maintenance-window-tasks` Beispiel listet alle Aufgaben für das angegebene Wartungsfenster auf, das das `AWS-RunPowerShellScript` Befehlsdokument aufruft.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

Ausgabe:

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

Beispiel 3: Um alle Aufgaben für ein Wartungsfenster aufzulisten, die eine Priorität von 3 haben

Das folgende `describe-maintenance-window-tasks` Beispiel listet alle Aufgaben für das angegebene Wartungsfenster auf, die den Wert `Priority` von haben`3`.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=3"
```

Ausgabe:

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

Beispiel 4: Um alle Aufgaben für ein Wartungsfenster aufzulisten, die eine Priorität von 1 haben, und verwenden Sie Run Command

In diesem `describe-maintenance-window-tasks` Beispiel werden alle Aufgaben für das angegebene Wartungsfenster aufgeführt, die einen Wert `Priority` von 1 und einen Verwendungszweck habenRun Command.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

Ausgabe:

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

Weitere Informationen finden Sie unter Informationen zu Wartungsfenstern (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeMaintenanceWindowTasksunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle Aufgaben für ein Wartungsfenster aufgeführt.

```
Get-SSMMaintenanceWindowTaskList -WindowId "mw-06cf17cbefcb4bf4f"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowTasks AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeMaintenanceWindows`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeMaintenanceWindows`.

CLI

AWS CLI

Beispiel 1: Um alle Wartungsfenster aufzulisten

Das folgende `describe-maintenance-windows` Beispiel listet alle Wartungsfenster in Ihrem AWS Konto in der aktuellen Region auf.

```
aws ssm describe-maintenance-windows
```

Ausgabe:

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

Beispiel 2: Um alle aktivierten Wartungsfenster aufzulisten

Das folgende `describe-maintenance-windows` Beispiel listet alle aktivierten Wartungsfenster auf.

```
aws ssm describe-maintenance-windows \
```

```
    --filters "Key=Enabled,Values=true"
```

Beispiel 3: Um Wartungsfenster aufzulisten, die einem bestimmten Namen entsprechen

In diesem `describe-maintenance-windows` Beispiel werden alle Wartungsfenster mit dem angegebenen Namen aufgeführt.

```
aws ssm describe-maintenance-windows \
    --filters "Key=Name,Values=MyMaintenanceWindow"
```

Weitere Informationen finden Sie unter Informationen über Maintenance Windows (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindows in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle Wartungsfenster Ihres Kontos aufgeführt.

```
Get-SSMMaintenanceWindowList
```

Ausgabe:

```
Cutoff   : 1
Duration : 4
Enabled  : True
Name     : My-First-Maintenance-Window
WindowId : mw-06d59c1a07c022145
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindows in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DescribeOpsItems** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeOpsItems`.

CLI

AWS CLI

Um eine Reihe von aufzulisten OpsItems

Im folgenden `describe-ops-items` Beispiel wird eine Liste aller offenen Konten OpsItems in Ihrem AWS Konto angezeigt.

```
aws ssm describe-ops-items \
    --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

Ausgabe:

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

Weitere Informationen finden Sie unter Arbeiten mit OpsItems im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribeOpsElemente in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void describeOpsItems(SsmClient ssmClient, String key) {
       try {
           OpsItemFilter filter = OpsItemFilter.builder()
               .key(OpsItemFilterKey.OPS_ITEM_ID)
               .values(key)
               .operator(OpsItemFilterOperator.EQUAL)
               .build();

           DescribeOpsItemsRequest itemsRequest =
DescribeOpsItemsRequest.builder()
               .maxResults(10)
               .opsItemFilters(filter)
               .build();

           DescribeOpsItemsResponse itemsResponse =
ssmClient.describeOpsItems(itemsRequest);
           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());
           }

       } catch (SsmException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }
```

- Einzelheiten zur API finden Sie unter DescribeOpsElemente in der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterSystems Manager mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeParameters`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeParameters`.

CLI

AWS CLI

Beispiel 1: Um alle Parameter aufzulisten

Das folgende `describe-parameters` Beispiel listet alle Parameter im AWS Girokonto und in der Region auf.

```
aws ssm describe-parameters
```

Ausgabe:

```
{
    "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": []
        }
    ]
}
```

Beispiel 2: Um alle Parameter aufzulisten, die bestimmten Metadaten entsprechen

In diesem `describe-parameters` Beispiel werden alle Parameter aufgeführt, die einem Filter entsprechen.

aws ssm describe-parameters --filters „Key=Type, Values=“ StringList

Ausgabe:

```
{
    "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": []
        }
    ]
}
```

Weitere Informationen finden Sie unter Suchen nach Systems Manager Manager-Parametern im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeParametersunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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);
        }
    }
}
```

- Einzelheiten zur API finden Sie DescribeParametersin der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieses Beispiel listet alle Parameter auf.

```
Get-SSMParameterList
```

Ausgabe:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:58:23 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
Type             : String
```

- Einzelheiten zur API finden Sie unter DescribeParameters AWS Tools for PowerShellCmdlet-Referenz.

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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(())
}
```

- Einzelheiten zur API finden Sie DescribeParametersin der API-Referenz zum AWS SDK für Rust.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterSystems Manager mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribePatchBaselines`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribePatchBaselines`.

CLI

AWS CLI

Beispiel 1: Um alle Patch-Baselines aufzulisten

Im folgenden `describe-patch-baselines` Beispiel werden Details für alle Patch-Baselines in Ihrem Konto in der aktuellen Region abgerufen.

```
aws ssm describe-patch-baselines
```

Ausgabe:

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

Beispiel 2: Um alle Patch-Baselines aufzulisten, die bereitgestellt werden von AWS

Das folgende `describe-patch-baselines` Beispiel listet alle Patch-Baselines auf, die von bereitgestellt werden. AWS

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[AWS]"
```

Beispiel 3: Um alle Patch-Baselines aufzulisten, die Ihnen gehören

Im folgenden `describe-patch-baselines` Beispiel werden alle benutzerdefinierten Patch-Baselines aufgeführt, die in Ihrem Konto in der aktuellen Region erstellt wurden.

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[Self]"
```

Weitere Informationen finden Sie unter Über vordefinierte und benutzerdefinierte Patch-Baselines im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribePatchBaselines in der Befehlsreferenz.AWS CLI

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle Patch-Baselines aufgeführt.

```
Get-SSMPatchBaseline
```

Ausgabe:

```
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...
```

Beispiel 2: In diesem Beispiel werden alle Patch-Baselines aufgeführt, die von bereitgestellt werden. AWS Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="OWNER";Values=@("AWS")}
```

Ausgabe:

```
Get-SSMPatchBaseline -Filter $filter1
```

Beispiel 3: In diesem Beispiel werden alle Patch-Baselines mit Ihnen als Eigentümer aufgeführt. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="OWNER";Values=@("Self")}
```

Ausgabe:

```
Get-SSMPatchBaseline -Filter $filter1
```

Beispiel 4: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jedes Tag zu erstellen.

```
$filter1 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter1.Key = "OWNER"
$filter1.Values = "AWS"

Get-SSMPatchBaseline -Filter $filter1
```

Ausgabe:

```
BaselineDescription                     BaselineId
                            BaselineName            DefaultBaselin


                                                    e
------------------                      ----------
                            ------------            --------------
Default Patch Baseline Provided by AWS. arn:aws:ssm:us-
west-2:123456789012:patchbaseline/pb-04fb4ae6142167966 AWS-DefaultPatchBaseline
 True
```

- Einzelheiten zur API finden Sie unter DescribePatchBaselines in der Cmdlet-Referenz.AWS Tools for PowerShell

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DescribePatchGroupState** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribePatchGroupState`.

CLI

AWS CLI

Um den Status einer Patch-Gruppe abzurufen

Im folgenden `describe-patch-group-state` Beispiel wird die allgemeine Zusammenfassung der Patch-Konformität für eine Patchgruppe abgerufen.

```
aws ssm describe-patch-group-state \
    --patch-group "Production"
```

Ausgabe:

```
{
    "Instances": 21,
    "InstancesWithCriticalNonCompliantPatches": 1,
    "InstancesWithFailedPatches": 2,
    "InstancesWithInstalledOtherPatches": 3,
    "InstancesWithInstalledPatches": 21,
    "InstancesWithInstalledPendingRebootPatches": 2,
    "InstancesWithInstalledRejectedPatches": 1,
    "InstancesWithMissingPatches": 3,
    "InstancesWithNotApplicablePatches": 4,
    "InstancesWithOtherNonCompliantPatches": 1,
    "InstancesWithSecurityNonCompliantPatches": 1,
    "InstancesWithUnreportedNotApplicablePatches": 2
}
```

Weitere Informationen finden Sie unter Über Patchgruppen__ < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html> und Grundlegendes zu den Werten für den Status der Patch-Konformität im Systems AWS Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribePatchGroupState AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die allgemeine Zusammenfassung der Patch-Konformität für eine Patch-Gruppe abgerufen.

```
Get-SSMPatchGroupState -PatchGroup "Production"
```

Ausgabe:

```
Instances                           : 4
InstancesWithFailedPatches          : 1
InstancesWithInstalledOtherPatches : 4
InstancesWithInstalledPatches       : 3
InstancesWithMissingPatches         : 0
InstancesWithNotApplicablePatches  : 0
```

- Einzelheiten zur API finden Sie unter DescribePatchGroupState AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribePatchGroups`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribePatchGroups`.

CLI

AWS CLI

Um Patch-Gruppenregistrierungen anzuzeigen

Das folgende `describe-patch-groups` Beispiel listet die Patch-Gruppenregistrierungen auf.

```
aws ssm describe-patch-groups
```

Ausgabe:

```
{
    "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
            }
        },
        ...
    ]
}
```

Weitere Informationen finden Sie unter Erstellen einer Patchgruppe__ < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html> und Hinzufügen einer Patchgruppe zu einer Patch-Baseline im Systems AWS Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DescribePatchGruppen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Registrierungen der Patchgruppen aufgeführt.

```
Get-SSMPatchGroup
```

Ausgabe:

```
BaselineIdentity                                                PatchGroup
----------------                                                ----------
Amazon.SimpleSystemsManagement.Model.PatchBaselineIdentity Production
```

- Einzelheiten zur API finden Sie unter DescribePatchGroups in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetAutomationExecution`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetAutomationExecution`.

CLI

AWS CLI

Um Details zu einer Automatisierungsausführung anzuzeigen

Im folgenden `get-automation-execution` Beispiel werden detaillierte Informationen zu einer Automatisierungsausführung angezeigt.

```
aws ssm get-automation-execution \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Ausgabe:

```
{
    "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
        }
    }
}
```

Weitere Informationen finden Sie unter Exemplarische Vorgehensweise: Patchen eines Linux-AMI (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter GetAutomationAusführung in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Details einer Automatisierungsausführung angezeigt.

```
Get-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

Ausgabe:

```
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...}
```

Beispiel 2: In diesem Beispiel werden die Schrittdetails für die angegebene Automatisierungsausführungs-ID aufgeführt

```
Get-SSMAutomationExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object -ExpandProperty StepExecutions | Select-Object
 StepName, Action, StepStatus, ValidNextSteps
```

Ausgabe:

```
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    {}
```

- Einzelheiten zur API finden Sie unter GetAutomationExecution in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetCommandInvocation`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetCommandInvocation`.

CLI

AWS CLI

Um die Details eines Befehlsaufrufs anzuzeigen

Das folgende `get-command-invocation` Beispiel listet alle Aufrufe des angegebenen Befehls auf der angegebenen Instanz auf.

```
aws ssm get-command-invocation \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --instance-id "i-1234567890abcdef0"
```

Ausgabe:

```
{
    "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
    }
}
```

Weitere Informationen finden Sie unter Understanding Command Statuses im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter GetCommandAufrufen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Details eines Befehls angezeigt, der auf einer Instanz ausgeführt wurde.

```
Get-SSMCommandInvocationDetail -InstanceId "i-0cb2b964d3e14fd9f" -CommandId
 "b8eac879-0541-439d-94ec-47a80d554f44"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter GetCommandInvocation in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **GetConnectionStatus** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetConnectionStatus`.

CLI

AWS CLI

Um den Verbindungsstatus einer verwalteten Instanz anzuzeigen

In diesem `get-connection-status` Beispiel wird der Verbindungsstatus der angegebenen verwalteten Instanz zurückgegeben.

```
aws ssm get-connection-status \
    --target i-1234567890abcdef0
```

Ausgabe:

```
{
    "Target": "i-1234567890abcdef0",
    "Status": "connected"
}
```

- Einzelheiten zur API finden Sie unter GetConnectionStatus in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird der Session Manager-Verbindungsstatus für eine Instanz abgerufen, um festzustellen, ob sie verbunden und bereit ist, Session Manager-Verbindungen zu empfangen.

```
Get-SSMConnectionStatus -Target i-0a1caf234f12d3dc4
```

Ausgabe:

```
Status    Target
------    ------
```

```
Connected i-0a1caf234f12d3dc4
```

- Einzelheiten zur API finden Sie unter GetConnectionStatus in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetDefaultPatchBaseline`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetDefaultPatchBaseline`.

CLI

AWS CLI

Beispiel 1: Um die Standard-Windows-Patch-Baseline anzuzeigen

Im folgenden `get-default-patch-baseline` Beispiel werden Details für die Standard-Patch-Baseline für Windows Server abgerufen.

```
aws ssm get-default-patch-baseline
```

Ausgabe:

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

Beispiel 2: So zeigen Sie die Standard-Patch-Baseline für Amazon Linux an

Im folgenden `get-default-patch-baseline` Beispiel werden Details für die Standard-Patch-Baseline für Amazon Linux abgerufen.

```
aws ssm get-default-patch-baseline \
    --operating-system AMAZON_LINUX
```

Ausgabe:

```
{
    "BaselineId": "pb-047c6eb9c8fc12345",
    "OperatingSystem": "AMAZON_LINUX"
}
```

Weitere Informationen finden Sie unter Über vordefinierte und benutzerdefinierte Patch-Baselines__ < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html> und Eine bestehende Patch-Baseline als Standard festlegen im Systems AWS Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetDefaultPatchBaselinein AWS CLI der Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die Standard-Patch-Baseline angezeigt.

```
Get-SSMDefaultPatchBaseline
```

Ausgabe:

```
arn:aws:ssm:us-west-2:123456789012:patchbaseline/pb-04fb4ae6142167966
```

- Einzelheiten zur API finden Sie unter GetDefaultPatchBaseline AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetDeployablePatchSnapshotForInstance`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetDeployablePatchSnapshotForInstance`.

CLI

AWS CLI

Um den aktuellen Snapshot für die Patch-Baseline abzurufen, verwendet eine Instanz

Im folgenden `get-deployable-patch-snapshot-for-instance` Beispiel werden Details für den aktuellen Snapshot für die angegebene Patch-Baseline abgerufen, die von einer Instanz verwendet wird. Dieser Befehl muss von der Instanz aus mit den Anmeldeinformationen der Instanz ausgeführt werden. Um sicherzustellen, dass er die Instance-Anmeldeinformationen verwendet, führen Sie ihn aus `aws configure` und geben Sie nur die Region Ihrer Instance an. Lassen Sie die `Secret Key` Felder `Access Key` und leer.

Tipp: Verwenden Sie`uuidgen`, um eine zu generieren`snapshot-id`.

```
aws ssm get-deployable-patch-snapshot-for-instance \
    --instance-id "i-1234567890abcdef0" \
    --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

Ausgabe:

```
{
    "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"
}
```

Weitere Informationen finden Sie unter Parametername: Snapshot-ID im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetDeployablePatchSnapshotForInstanceunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird der aktuelle Snapshot für die von einer Instance verwendete Patch-Baseline angezeigt. Dieser Befehl muss von der Instance aus mit den Anmeldeinformationen der Instanz ausgeführt werden. Um sicherzustellen, dass die Instanzanmeldedaten verwendet werden, übergibt das Beispiel ein **`Amazon.Runtime.InstanceProfileAWSCredentials`** Objekt an den Credentials-Parameter.

```
$credentials = [Amazon.Runtime.InstanceProfileAWSCredentials]::new()
Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f" -Credentials $credentials
```

Ausgabe:

```
InstanceId          SnapshotDownloadUrl
----------          -------------------
i-0cb2b964d3e14fd9f https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...1692/4681775b-098f-4435...
```

Beispiel 2: Dieses Beispiel zeigt, wie Sie die vollständigen Daten abrufen können SnapshotDownloadUrl. Dieser Befehl muss von der Instanz aus mit den Instanzanmeldedaten ausgeführt werden. Um sicherzustellen, dass die Instanzanmeldedaten verwendet werden, konfiguriert das Beispiel die PowerShell Sitzung für die Verwendung eines **`Amazon.Runtime.InstanceProfileAWSCredentials`** Objekts.

```
Set-AWSCredential -Credential
 ([Amazon.Runtime.InstanceProfileAWSCredentials]::new())
(Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f").SnapshotDownloadUrl
```

Ausgabe:

```
https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...
```

- Einzelheiten zur API finden Sie unter GetDeployablePatchSnapshotForInstance AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetDocument`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetDocument`.

CLI

AWS CLI

Um den Inhalt des Dokuments abzurufen

Im folgenden `get-document` Beispiel wird der Inhalt eines Systems Manager Manager-Dokuments angezeigt.

```
aws ssm get-document \
    --name "AWS-RunShellScript"
```

Ausgabe:

```
{
    "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"
}
```

Weitere Informationen finden Sie unter AWS Systems Manager Manager-Dokumente im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetDocumentunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird der Inhalt eines Dokuments zurückgegeben.

```
Get-SSMDocument -Name "RunShellScript"
```

Ausgabe:

```
Content
-------
{...
```

Beispiel 2: In diesem Beispiel wird der vollständige Inhalt eines Dokuments angezeigt.

```
(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 }}"
         }
      }
   ]
}
```

- Einzelheiten zur API finden Sie unter GetDocument AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetInventory`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetInventory`.

CLI

AWS CLI

Um Ihr Inventar einzusehen

In diesem Beispiel werden die benutzerdefinierten Metadaten für Ihr Inventar abgerufen.

Befehl:

```
aws ssm get-inventory
```

Ausgabe:

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

- Einzelheiten zur API finden Sie GetInventoryin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die benutzerdefinierten Metadaten für Ihr Inventar abgerufen.

```
Get-SSMInventory
```

Ausgabe:

```
Data
     Id
----
     --
{[AWS:InstanceInformation,
 Amazon.SimpleSystemsManagement.Model.InventoryResultItem]} i-0cb2b964d3e14fd9f
```

- Einzelheiten zur API finden Sie unter GetInventory AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetInventorySchema`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetInventorySchema`.

CLI

AWS CLI

Um Ihr Inventarschema einzusehen

In diesem Beispiel wird eine Liste von Inventartypnamen für das Konto zurückgegeben.

Befehl:

```
aws ssm get-inventory-schema
```

Ausgabe:

```
{
  "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--"
}
```

Um das Inventarschema für einen bestimmten Inventartyp anzuzeigen

In diesem Beispiel wird das Inventarschema für den AWS Inventartyp „AWS Komponente“ zurückgegeben.

Befehl:

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- Einzelheiten zur API finden Sie unter GetInventorySchema in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Liste von Inventartypnamen für das Konto zurückgegeben.

```
Get-SSMInventorySchema
```

- Einzelheiten zur API finden Sie unter GetInventorySchema in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetMaintenanceWindow`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetMaintenanceWindow`.

CLI

AWS CLI

Um Informationen über ein Wartungsfenster zu erhalten

Im folgenden `get-maintenance-window` Beispiel werden Details zum angegebenen Wartungsfenster abgerufen.

```
aws ssm get-maintenance-window \
    --window-id "mw-03eb9db428EXAMPLE"
```

Ausgabe:

```
{
    "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"
}
```

Weitere Informationen finden Sie unter Informationen zu Wartungsfenstern (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter GetMaintenanceFenster in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden Details zu einem Wartungsfenster abgerufen.

```
Get-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter GetMaintenanceFenster in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **GetMaintenanceWindowExecution** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetMaintenanceWindowExecution`.

CLI

AWS CLI

Um Informationen über die Ausführung einer Aufgabe im Wartungsfenster zu erhalten

Das folgende `get-maintenance-window-execution` Beispiel listet Informationen über eine Aufgabe auf, die im Rahmen der Ausführung des angegebenen Wartungsfensters ausgeführt wurde.

```
aws ssm get-maintenance-window-execution \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Ausgabe:

```
{
    "Status": "SUCCESS",
    "TaskIds": [
        "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
    ],
    "StartTime": 1487692834.595,
    "EndTime": 1487692835.051,
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
}
```

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter GetMaintenanceWindowExecution AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden Informationen über eine Aufgabe aufgeführt, die im Rahmen der Ausführung eines Wartungsfensters ausgeführt wurde.

```
Get-SSMMaintenanceWindowExecution -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter GetMaintenanceWindowExecution AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetMaintenanceWindowExecutionTask`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetMaintenanceWindowExecutionTask`.

CLI

AWS CLI

Um Informationen über die Ausführung einer Aufgabe im Wartungsfenster zu erhalten

Im folgenden `get-maintenance-window-execution-task` Beispiel werden Informationen zu einer Aufgabe aufgeführt, die Teil der Ausführung des angegebenen Wartungsfensters ist.

```
aws ssm get-maintenance-window-execution-task \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```

Ausgabe:

```
{
    "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"
}
```

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter GetMaintenanceWindowExecutionTask in AWS CLI Command Reference.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden Informationen zu einer Aufgabe aufgeführt, die Teil einer Ausführung im Rahmen eines Wartungsfensters war.

```
Get-SSMMaintenanceWindowExecutionTask -TaskId "ac0c6ae1-daa3-4a89-832e-
d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter GetMaintenanceWindowExecutionTask in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetParameterHistory`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetParameterHistory`.

CLI

AWS CLI

Um einen Werteverlauf für einen Parameter abzurufen

Im folgenden `get-parameter-history` Beispiel wird der Verlauf der Änderungen für den angegebenen Parameter einschließlich seines Werts aufgeführt.

```
aws ssm get-parameter-history \
    --name "MyStringParameter"
```

Ausgabe:

```
{
    "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": []
        }
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit Parameterversionen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter GetParameterVerlauf in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird der Werteverlauf für einen Parameter aufgeführt.

```
Get-SSMParameterHistory -Name "Welcome"
```

Ausgabe:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:55:25 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
Type             : String
Value            : helloWorld
```

- Einzelheiten zur API finden Sie unter GetParameterHistory in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetParameters`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetParameters`.

CLI

AWS CLI

Beispiel 1: Um die Werte für einen Parameter aufzulisten

Das folgende `get-parameters` Beispiel listet die Werte für die drei angegebenen Parameter auf.

```
aws ssm get-parameters \
    --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"
```

Ausgabe:

```
{
    "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"
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit dem Parameterspeicher im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: Um Namen und Werte mehrerer Parameter mit der Option ``--query`` aufzulisten

Das folgende `get-parameters` Beispiel listet die Namen und Werte für die angegebenen Parameter auf.

```
aws ssm get-parameters \
    --names MyStringParameter MyStringListParameter \
    --query "Parameters[*].{Name:Name,Value:Value}"
```

Ausgabe:

```
[
    {
        "Name": "MyStringListParameter",
        "Value": "alpha,beta,gamma"
    },
```

```
    {
        "Name": "MyStringParameter",
        "Value": "Vidi"
    }
]
```

Weitere Informationen finden Sie unter Arbeiten mit dem Parameterspeicher im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: So zeigen Sie den Wert eines Parameters mithilfe von Beschriftungen an

Das folgende `get-parameter` Beispiel listet den Wert für den angegebenen Einzelparameter mit einer angegebenen Bezeichnung auf.

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

Ausgabe:

```
{
    "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": []
}
```

Weitere Informationen finden Sie unter Arbeiten mit Parameterbeschriftungen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetParametersunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Werte für einen Parameter aufgeführt.

```
Get-SSMParameterValue -Name "Welcome"
```

Ausgabe:

```
InvalidParameters Parameters
----------------- ----------
{}                {Welcome}
```

Beispiel 2: In diesem Beispiel werden die Details des Werts aufgeführt.

```
(Get-SSMParameterValue -Name "Welcome").Parameters
```

Ausgabe:

```
Name    Type   Value
----    ----   -----
Welcome String Good day, Sunshine!
```

- Einzelheiten zur API finden Sie unter GetParameters AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **GetPatchBaseline** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetPatchBaseline`.

CLI

AWS CLI

Um eine Patch-Baseline anzuzeigen

Im folgenden `get-patch-baseline` Beispiel werden die Details für die angegebene Patch-Baseline abgerufen.

```
aws ssm get-patch-baseline \
    --baseline-id "pb-0123456789abcdef0"
```

Ausgabe:

```
{
    "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": []
}
```

Weitere Informationen finden Sie unter About Patch Baselines im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter GetPatchBaseline in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Details für eine Patch-Baseline angezeigt.

```
Get-SSMPatchBaselineDetail -BaselineId "pb-03da896ca3b68b639"
```

Ausgabe:

```
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 : {}
```

- Einzelheiten zur API finden Sie unter GetPatchBaseline in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **GetPatchBaselineForPatchGroup** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetPatchBaselineForPatchGroup`.

CLI

AWS CLI

Um die Patch-Baseline für eine Patch-Gruppe anzuzeigen

Im folgenden `get-patch-baseline-for-patch-group` Beispiel werden Details zur Patch-Baseline für die angegebene Patchgruppe abgerufen.

```
aws ssm get-patch-baseline-for-patch-group \
    --patch-group "DEV"
```

Ausgabe:

```
{
    "PatchGroup": "DEV",
    "BaselineId": "pb-0123456789abcdef0",
    "OperatingSystem": "WINDOWS"
}
```

Weitere Informationen finden Sie unter Erstellen einer Patchgruppe__ < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html> und Hinzufügen einer Patchgruppe zu einer Patch-Baseline im Systems AWS Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter GetPatchBaselineForPatchGroup AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die Patch-Baseline für eine Patch-Gruppe angezeigt.

```
Get-SSMPatchBaselineForPatchGroup -PatchGroup "Production"
```

Ausgabe:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Einzelheiten zur API finden Sie unter GetPatchBaselineForPatchGroup AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListAssociationVersions`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListAssociationVersions`.

CLI

AWS CLI

Um alle Versionen einer Assoziation für eine bestimmte Zuordnungs-ID aufzulisten

Das folgende `list-association-versions` Beispiel listet alle Versionen der angegebenen Assoziationen auf.

```
aws ssm list-association-versions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Ausgabe:

```
{
```

```
"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"
        }
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter ListAssociationVersionen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle Versionen der bereitgestellten Assoziation abgerufen.

```
Get-SSMAssociationVersionList -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Ausgabe:

```
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}
```

- Einzelheiten zur API finden Sie unter ListAssociationVersionen in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListAssociations`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListAssociations`.

CLI

AWS CLI

Beispiel 1: Um Ihre Assoziationen für eine bestimmte Instanz aufzulisten

Das folgende Beispiel für list-associations listet alle Assoziationen mit dem UpdateSSMAgent auf AssociationName.

```
aws ssm list-associations /
    --association-filter-list "key=AssociationName,value=UpdateSSMAgent"
```

Ausgabe:

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

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So listen Sie Ihre Verknüpfungen für ein bestimmtes Dokument auf

Das folgende Beispiel für Listenzuordnungen listet alle Verknüpfungen für das angegebene Dokument auf.

```
aws ssm list-associations /
    --association-filter-list "key=Name,value=AWS-UpdateSSMAgent"
```

Ausgabe:

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

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ListAssociationsunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle Assoziationen für eine Instanz aufgeführt. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="InstanceId";Value=@("i-0000293ffd8c57862")}
Get-SSMAssociationList -AssociationFilterList $filter1
```

Ausgabe:

```
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}
```

Beispiel 2: In diesem Beispiel werden alle Verknüpfungen für ein Konfigurationsdokument aufgeführt. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter2 = @{Key="Name";Value=@("AWS-UpdateSSMAgent")}
Get-SSMAssociationList -AssociationFilterList $filter2
```

Ausgabe:

```
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}
```

Beispiel 3: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jeden Filter zu erstellen.

```
$filter1 = New-Object Amazon.SimpleSystemsManagement.Model.AssociationFilter
$filter1.Key = "InstanceId"
$filter1.Value = "i-0000293ffd8c57862"

Get-SSMAssociationList -AssociationFilterList $filter1
```

Ausgabe:

```
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}
```

- Einzelheiten zur API finden Sie unter ListAssociations AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListCommandInvocations`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListCommandInvocations`.

CLI

AWS CLI

Um die Aufrufe eines bestimmten Befehls aufzulisten

Das folgende `list-command-invocations` Beispiel listet alle Aufrufe eines Befehls auf.

```
aws ssm list-command-invocations \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --details
```

Ausgabe:

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


Weitere Informationen finden Sie unter Understanding Command Statuses im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter ListCommandAufrufe in AWS CLI der Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieses Beispiel listet alle Aufrufe eines Befehls auf.

```
Get-SSMCommandInvocation -CommandId "b8eac879-0541-439d-94ec-47a80d554f44" -
Detail $true
```

Ausgabe:

```
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        :
```

Beispiel 2: In diesem Beispiel wird der Aufruf der Befehls-ID CommandPlugins e1eb2e3c-ed4c-5123-45c1-234f5612345f aufgeführt

```
Get-SSMCommandInvocation -CommandId e1eb2e3c-ed4c-5123-45c1-234f5612345f -Detail:
$true | Select-Object -ExpandProperty CommandPlugins
```

Ausgabe:

```
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
```

- Einzelheiten zur API ListCommandfinden Sie AWS Tools for PowerShell unter Invocations in Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListCommands`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListCommands`.

CLI

AWS CLI

Beispiel 1: Um den Status eines bestimmten Befehls abzurufen

Im folgenden `list-commands` Beispiel wird der Status des angegebenen Befehls abgerufen und angezeigt.

```
aws ssm list-commands \
    --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"
```

Beispiel 2: Um den Status von Befehlen abzurufen, die nach einem bestimmten Datum angefordert wurden

Im folgenden `list-commands` Beispiel werden die Details von Befehlen abgerufen, die nach dem angegebenen Datum angefordert wurden.

```
aws ssm list-commands \
    --filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

Beispiel 3: Um alle Befehle aufzulisten, die in einem AWS Konto angefordert wurden

Das folgende `list-commands` Beispiel listet alle Befehle auf, die von Benutzern im aktuellen AWS Konto und in der Region angefordert wurden.

```
aws ssm list-commands
```

Ausgabe:

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

Weitere Informationen finden Sie unter Ausführen von Befehlen mit Systems Manager Run Command im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ListCommandsunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieses Beispiel listet alle angeforderten Befehle auf.

```
Get-SSMCommand
```

Ausgabe:

```
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            : {}
```

Beispiel 2: In diesem Beispiel wird der Status eines bestimmten Befehls abgerufen.

```
Get-SSMCommand -CommandId "4b75a163-d39a-4d97-87c9-98ae52c6be35"
```

Beispiel 3: In diesem Beispiel werden alle SSM-Befehle abgerufen, die nach dem 2019-04-01T 00:00:00 Z aufgerufen wurden

```
Get-SSMCommand -Filter @{Key="InvokedAfter";Value="2019-04-01T00:00:00Z"} |
 Select-Object CommandId, DocumentName, Status, RequestedDateTime | Sort-Object -
Property RequestedDateTime -Descending
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter Cmdlet-Referenz. ListCommandsAWS Tools for PowerShell

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListComplianceItems`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListComplianceItems`.

CLI

AWS CLI

Um Compliance-Artikel für eine bestimmte Instanz aufzulisten

In diesem Beispiel werden alle Konformitätselemente für die angegebene Instanz aufgeführt.

Befehl:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance"
```

Ausgabe:

```
{
  "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--"
}
```

Um Konformitätselemente für eine bestimmte Instanz und Zuordnungs-ID aufzulisten

In diesem Beispiel werden alle Konformitätselemente für die angegebene Instanz und Zuordnungs-ID aufgeführt.

Befehl:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --filters
```

```
 "Key=ComplianceType,Values=Association,Type=EQUAL"
 "Key=Id,Values=e4c2ed6d-516f-41aa-aa2a-0123456789ab,Type=EQUAL"
```

Um Compliance-Elemente für eine Instanz nach einem bestimmten Datum und einer bestimmten Uhrzeit aufzulisten

In diesem Beispiel werden alle Compliance-Elemente für eine Instanz nach dem angegebenen Datum und der angegebenen Uhrzeit aufgeführt.

Befehl:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --filters
 "Key=ExecutionTime,Values=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

- Einzelheiten zur API finden Sie unter ListComplianceElemente in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die Liste der Compliance-Elemente für die angegebene Ressourcen-ID und den angegebenen Ressourcentyp aufgeführt, wobei nach dem Compliance-Typ „Association“ gefiltert wird

```
Get-SSMComplianceItemList -ResourceId i-1a2caf345f67d0dc2 -ResourceType
 ManagedInstance -Filter @{Key="ComplianceType";Values="Association"}
```

Ausgabe:

```
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            :
```

- Einzelheiten zur API finden Sie unter Referenz zu ListComplianceElementen in AWS Tools for PowerShell Cmdlets.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListComplianceSummaries`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListComplianceSummaries`.

CLI

AWS CLI

Um Konformitätszusammenfassungen für alle Konformitätstypen aufzulisten

In diesem Beispiel werden Konformitätszusammenfassungen für alle Compliance-Typen in Ihrem Konto aufgeführt.

Befehl:

```
aws ssm list-compliance-summaries
```

Ausgabe:

```
{
  "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=="
}
```

Um Konformitätszusammenfassungen für einen bestimmten Konformitätstyp aufzulisten

In diesem Beispiel wird die Konformitätszusammenfassung für den Kompatibilitätstyp Patch aufgeführt.

Befehl:

```
aws ssm list-compliance-summaries --filters
 "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Einzelheiten zur API finden Sie unter ListComplianceZusammenfassungen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Zusammenfassung der Anzahl der konformen und nicht konformen Ressourcen für alle Compliance-Typen zurückgegeben.

```
Get-SSMComplianceSummaryList
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter ListComplianceZusammenfassungen in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListDocumentVersions`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListDocumentVersions`.

CLI

AWS CLI

Um Dokumentversionen aufzulisten

Das folgende `list-document-versions` Beispiel listet alle Versionen eines Systems Manager Manager-Dokuments auf.

```
aws ssm list-document-versions \
    --name "Example"
```

Ausgabe:

```
{
    "DocumentVersions": [
        {
            "Name": "Example",
            "DocumentVersion": "1",
            "CreatedDate": 1583257938.266,
            "IsDefaultVersion": true,
            "DocumentFormat": "YAML",
            "Status": "Active"
        }
    ]
}
```

Weitere Informationen finden Sie unter Senden von Befehlen, die den Dokumentversionsparameter verwenden im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter ListDocumentVersionen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die Berechtigungsliste für ein Dokument zurückgegeben.

```
Get-SSMDocumentPermission -Name "RunShellScript" -PermissionType "Share"
```

Ausgabe:

```
all
```

- Einzelheiten zur API finden Sie unter ListDocumentVersions in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListDocuments`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListDocuments`.

CLI

AWS CLI

Beispiel 1: Um Dokumente aufzulisten

Das folgende `list-documents` Beispiel listet Dokumente auf, die dem anfragenden Konto gehören und mit dem benutzerdefinierten Tag versehen sind.

```
aws ssm list-documents \
    --filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

Ausgabe:

```
{
    "DocumentIdentifiers": [
        {
```

```
            "Name": "Example",
            "Owner": "29884EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Automation",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": [
                {
                    "Key": "DocUse",
                    "Value": "Testing"
                }
            ]
        }
    ]
}
```

Weitere Informationen finden Sie unter AWS Systems Manager Manager-Dokumente im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So listen Sie gemeinsam genutzte Dokumente auf

Das folgende `list-documents` Beispiel listet gemeinsam genutzte Dokumente auf, einschließlich privater geteilter Dokumente, die nicht Eigentum von sind AWS.

```
aws ssm list-documents \
    --filters Key=Name,Values=sharedDocNamePrefix  Key=Owner,Values=Private
```

Ausgabe:

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "12345EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
```

```
            "DocumentType": "Command",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": []
        }
    ]
}
```

Weitere Informationen finden Sie unter AWS Systems Manager Manager-Dokumente im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ListDocumentsunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Listet alle Konfigurationsdokumente in Ihrem Konto auf.

```
Get-SSMDocumentList
```

Ausgabe:

```
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
...
```

Beispiel 2: In diesem Beispiel werden alle Automatisierungsdokumente abgerufen, deren Name mit „Platform“ übereinstimmt

```
Get-SSMDocumentList -DocumentFilterList @{Key="DocumentType";Value="Automation"}
 | Where-Object Name -Match "Platform"
```

Ausgabe:

```
DocumentFormat  : JSON
DocumentType    : Automation
DocumentVersion : 7
Name            : KT-Get-Platform
Owner           : 987654123456
PlatformTypes   : {Windows, Linux}
SchemaVersion   : 0.3
Tags            : {}
TargetType      :
VersionName     :
```

- Einzelheiten zur API finden Sie unter ListDocumentsCmdlet-Referenz.AWS Tools for PowerShell

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListInventoryEntries`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListInventoryEntries`.

CLI

AWS CLI

Beispiel 1: So zeigen Sie bestimmte Inventartypeinträge für eine Instanz an

Im folgenden `list-inventory-entries` Beispiel werden die Inventareinträge für den AWS Inventartyp:Application für eine bestimmte Instanz aufgeführt.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "AWS:Application"
```

Ausgabe:

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

Beispiel 2: So zeigen Sie benutzerdefinierte Inventareinträge an, die einer Instanz zugewiesen sind

Das folgende `list-inventory-entries` Beispiel listet einen benutzerdefinierten Inventareintrag auf, der einer Instanz zugewiesen ist.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "Custom:RackInfo"
```

Ausgabe:

```
{
```

```
  "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"
    }
  ]
}
```

- Einzelheiten zur API finden Sie unter ListInventoryEinträge in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden alle benutzerdefinierten Inventareinträge für eine Instanz aufgeführt.

```
Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo"
```

Ausgabe:

```
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
```

Beispiel 2: In diesem Beispiel werden die Details aufgeführt.

```
(Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo").Entries
```

Ausgabe:

```
Key          Value
```

```
---          -----
RackLocation Bay B/Row C/Rack D/Shelf E
```

- Einzelheiten zur API finden Sie unter ListInventoryEinträge in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListResourceComplianceSummaries`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListResourceComplianceSummaries`.

CLI

AWS CLI

Um die Anzahl der Compliance-Anforderungen auf Ressourcenebene aufzulisten

In diesem Beispiel wird die Anzahl der Konformitäten auf Ressourcenebene zusammenfassend aufgeführt.

Befehl:

```
aws ssm list-resource-compliance-summaries
```

Ausgabe:

```
{
  "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--"
}
```

Um Compliance-Zusammenfassungen auf Ressourcenebene für einen bestimmten Konformitätstyp aufzulisten

In diesem Beispiel werden Konformitätszusammenfassungen auf Ressourcenebene für den Kompatibilitätstyp Patch aufgeführt.

Befehl:

```
aws ssm list-resource-compliance-summaries --filters
 "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Einzelheiten zur API finden Sie unter Befehlsreferenz [ListResourceComplianceSummaries](#).AWS CLI

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Zusammenfassung der Anzahl auf Ressourcenebene abgerufen. Die Zusammenfassung enthält Informationen über den Status „konform“ und „nicht konform“ sowie detaillierte Angaben zum Schweregrad von Produkten, die „Windows10" entsprechen. Da der MaxResult Standardwert 100 ist, wenn der Parameter nicht angegeben

ist und dieser Wert nicht gültig ist, wird der MaxResult Parameter hinzugefügt und der Wert auf 50 gesetzt.

```
$FilterValues = @{
  "Key"="Product"
        "Type"="EQUAL"
        "Values"="Windows10"
}
        Get-SSMResourceComplianceSummaryList -Filter $FilterValues -MaxResult 50
```

- Einzelheiten zur API finden Sie unter ListResourceComplianceSummaries AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListTagsForResource`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListTagsForResource`.

CLI

AWS CLI

Um die Tags aufzulisten, die auf eine Patch-Baseline angewendet wurden

Im folgenden `list-tags-for-resource` Beispiel werden die Tags für eine Patch-Baseline aufgeführt.

```
aws ssm list-tags-for-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0"
```

Ausgabe:

```
{
    "TagList": [
        {
            "Key": "Environment",
```

```
            "Value": "Production"
        },
        {
            "Key": "Region",
            "Value": "EMEA"
        }
    ]
}
```

Weitere Informationen finden Sie unter AWS Ressourcen taggen in der AWS allgemeinen Referenz.

- Einzelheiten zur API finden Sie ListTagsForResourcein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden die Tags für ein Wartungsfenster aufgelistet.

```
Get-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow"
```

Ausgabe:

```
Key   Value
---   -----
Stack Production
```

- Einzelheiten zur API finden Sie unter ListTagsForResource AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ModifyDocumentPermission`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ModifyDocumentPermission`.

CLI

AWS CLI

Um Dokumentberechtigungen zu ändern

Im folgenden `modify-document-permission` Beispiel wird ein Systems Manager Manager-Dokument öffentlich freigegeben.

```
aws ssm modify-document-permission \
    --name "Example" \
    --permission-type "Share" \
    --account-ids-to-add "All"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Freigeben eines Systems Manager Manager-Dokuments im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter ModifyDocumentBerechtigungen in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden allen Konten für ein Dokument „Teilen“ -Berechtigungen hinzugefügt. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Edit-SSMDocumentPermission -Name "RunShellScript" -PermissionType "Share" -
AccountIdsToAdd all
```

Beispiel 2: In diesem Beispiel werden einem bestimmten Konto für ein Dokument „Teilen“ -Berechtigungen hinzugefügt. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Edit-SSMDocumentPermission -Name "RunShellScriptNew" -PermissionType "Share" -
AccountIdsToAdd "123456789012"
```

- Einzelheiten zur API finden Sie unter ModifyDocumentPermission in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **PutComplianceItems** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`PutComplianceItems`.

CLI

AWS CLI

Um einen Konformitätstyp und Konformitätsdetails für eine bestimmte Instanz zu registrieren

In diesem Beispiel wird der Konformitätstyp `Custom:AVCheck` für die angegebene verwaltete Instanz registriert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Befehl:

```
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"
```

- Einzelheiten zur API finden Sie unter PutComplianceElemente in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein benutzerdefiniertes Compliance-Element für die angegebene verwaltete Instanz geschrieben

```
$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"
```

- Einzelheiten zur API finden Sie unter Referenz zu PutComplianceElementen in AWS Tools for PowerShell Cmdlets.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`PutInventory`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`PutInventory`.

CLI

AWS CLI

Um einer Instanz Kundenmetadaten zuzuweisen

In diesem Beispiel werden einer Instance Informationen zum Rack-Standort zugewiesen. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Befehl (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"}]}]'
```

Befehl (Windows):

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --items
 "TypeName=Custom:RackInfo,SchemaVersion=1.0,CaptureTime=2019-01-22T10:01:01Z,Content=[{R
 B/Row C/Rack D/Shelf F'}]"
```

- Einzelheiten zur API finden Sie PutInventoryin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden einer Instanz Informationen zum Rack-Standort zugewiesen. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
$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
```

- Einzelheiten zur API finden Sie unter PutInventory AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`PutParameter`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`PutParameter`.

CLI

AWS CLI

Beispiel 1: Um einen Parameterwert zu ändern

Im folgenden `put-parameter` Beispiel wird der Wert des angegebenen Parameters geändert.

```
aws ssm put-parameter \
    --name "MyStringParameter" \
    --type "String" \
    --value "Vici" \
    --overwrite
```

Ausgabe:

```
{
    "Version": 2,
    "Tier": "Standard"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), 'Verwalten von Parameterstufen`__ und Arbeiten mit Parameterrichtlinien im Systems AWS Manager Manager-Benutzerhandbuch. < https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>

Beispiel 2: So erstellen Sie einen erweiterten Parameter

Im folgenden `put-parameter` Beispiel wird ein erweiterter Parameter erstellt.

```
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
```

Ausgabe:

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), 'Verwalten von Parameterstufen`__ und Arbeiten mit Parameterrichtlinien im Systems AWS Manager Manager-Benutzerhandbuch. < https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>

Beispiel 3: So konvertieren Sie einen Standardparameter in einen erweiterten Parameter

Das folgende `put-parameter` Beispiel konvertiert einen vorhandenen Standardparameter in einen erweiterten Parameter.

```
aws ssm put-parameter \
    --name "MyConvertedParameter" \
    --value "abc123" \
    --type "String" \
    --tier Advanced \
    --overwrite
```

Ausgabe:

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), 'Verwalten von Parameterstufen`__ und Arbeiten mit Parameterrichtlinien im Systems AWS Manager Manager-Benutzerhandbuch. < https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>

Beispiel 4: So erstellen Sie einen Parameter mit angehängter Richtlinie

Im folgenden `put-parameter` Beispiel wird ein erweiterter Parameter mit einer angehängten Parameterrichtlinie erstellt.

```
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\"}}]"
```

Ausgabe:

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), 'Verwalten von Parameterstufen`__ und Arbeiten mit Parameterrichtlinien im Systems AWS Manager Manager-Benutzerhandbuch. < https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>

Beispiel 5: So fügen Sie einem vorhandenen Parameter eine Richtlinie hinzu

Im folgenden `put-parameter` Beispiel wird eine Richtlinie an einen vorhandenen erweiterten Parameter angehängt.

```
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
```

Ausgabe:

```
{
```

```
    "Version": 2,
    "Tier": "Advanced"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), 'Verwalten von Parameterstufen`__ und Arbeiten mit Parameterrichtlinien im Systems AWS Manager Manager-Benutzerhandbuch. < https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>

- Einzelheiten zur API finden Sie unter PutParameterBefehlsreferenz.AWS CLI

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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);
        }
    }
}
```

- Einzelheiten zur API finden Sie PutParameterin der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Parameter erstellt. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "helloWorld"
```

Beispiel 2: In diesem Beispiel wird ein Parameter geändert. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "Good day, Sunshine!" -
Overwrite $true
```

- Einzelheiten zur API finden Sie unter PutParameter AWS Tools for PowerShellCmdlet-Referenz.

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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(())
```

```
}
```

- Einzelheiten zur API finden Sie PutParameterin der API-Referenz zum AWS SDK für Rust.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterSystems Manager mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **RegisterDefaultPatchBaseline** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`RegisterDefaultPatchBaseline`.

CLI

AWS CLI

Um die Standard-Patch-Baseline festzulegen

Im folgenden `register-default-patch-baseline` Beispiel wird die angegebene benutzerdefinierte Patch-Baseline als Standard-Patch-Baseline für den unterstützten Betriebssystemtyp registriert.

```
aws ssm register-default-patch-baseline \
    --baseline-id "pb-abc123cf9bEXAMPLE"
```

Ausgabe:

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

Im folgenden `register-default-patch-baseline` Beispiel wird die von AWS für CentOS bereitgestellte Standard-Patch-Baseline als Standard-Patch-Baseline registriert.

```
aws ssm register-default-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0574b43a65ea646ed"
```

Ausgabe:

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

Weitere Informationen finden Sie unter Über vordefinierte und benutzerdefinierte Patch-Baselines im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter RegisterDefaultPatchBaseline AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline als Standard-Patch-Baseline registriert.

```
Register-SSMDefaultPatchBaseline -BaselineId "pb-03da896ca3b68b639"
```

Ausgabe:

```
pb-03da896ca3b68b639
```

- Einzelheiten zur API finden Sie unter RegisterDefaultPatchBaseline AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`RegisterPatchBaselineForPatchGroup`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`RegisterPatchBaselineForPatchGroup`.

CLI

AWS CLI

Um eine Patch-Baseline für eine Patch-Gruppe zu registrieren

Im folgenden `register-patch-baseline-for-patch-group` Beispiel wird eine Patch-Baseline für eine Patchgruppe registriert.

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-045f10b4f382baeda" \
    --patch-group "Production"
```

Ausgabe:

```
{
    "BaselineId": "pb-045f10b4f382baeda",
    "PatchGroup": "Production"
}
```

Weitere Informationen finden Sie unter Erstellen einer Patchgruppe__ < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html> und Hinzufügen einer Patchgruppe zu einer Patch-Baseline im Systems AWS Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter RegisterPatchBaselineForPatchGroup AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline für eine Patch-Gruppe registriert.

```
Register-SSMPatchBaselineForPatchGroup -BaselineId "pb-03da896ca3b68b639" -
PatchGroup "Production"
```

Ausgabe:

```
BaselineId           PatchGroup
----------           ----------
```

```
pb-03da896ca3b68b639 Production
```

- Einzelheiten zur API finden Sie unter RegisterPatchBaselineForPatchGroup AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`RegisterTargetWithMaintenanceWindow`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`RegisterTargetWithMaintenanceWindow`.

CLI

AWS CLI

Beispiel 1: Um ein einzelnes Ziel mit einem Wartungsfenster zu registrieren

Im folgenden `register-target-with-maintenance-window` Beispiel wird eine Instanz mit einem Wartungsfenster registriert.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \
    --owner-information "Single instance" \
    --resource-type "INSTANCE"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Beispiel 2: Um mehrere Ziele mithilfe von Instanz-IDs für ein Wartungsfenster zu registrieren

Im folgenden `register-target-with-maintenance-window` Beispiel werden zwei Instanzen mit einem Wartungsfenster registriert, indem ihre Instanz-IDs angegeben werden.

```
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"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Beispiel 3: Um Ziele mithilfe von Ressourcen-Tags für ein Wartungsfenster zu registrieren

Im folgenden `register-target-with-maintenance-window` Beispiel werden Instanzen mit einem Wartungsfenster registriert, indem Ressourcen-Tags angegeben werden, die auf die Instanzen angewendet wurden.

```
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"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Beispiel 4: Um Ziele mithilfe einer Gruppe von Tag-Schlüsseln zu registrieren

Im folgenden `register-target-with-maintenance-window` Beispiel werden Instanzen registriert, denen unabhängig von ihren Schlüsselwerten ein oder mehrere Tag-Schlüssel zugewiesen wurden.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
```

```
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Beispiel 5: Um Ziele mit einem Ressourcengruppennamen zu registrieren

Im folgenden `register-target-with-maintenance-window` Beispiel wird eine angegebene Ressourcengruppe unabhängig vom darin enthaltenen Ressourcentyp registriert.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Weitere Informationen finden Sie unter Registrieren einer Zielinstanz mit dem Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter RegisterTargetWithMaintenanceFenster in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Instanz mit einem Wartungsfenster registriert.

```
$option1 = @{Key="InstanceIds";Values=@("i-0000293ffd8c57862")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Ausgabe:

```
d8e47760-23ed-46a5-9f28-927337725398
```

Beispiel 2: In diesem Beispiel werden mehrere Instanzen mit einem Wartungsfenster registriert.

```
$option1 =
 @{Key="InstanceIds";Values=@("i-0000293ffd8c57862","i-0cb2b964d3e14fd9f")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Ausgabe:

```
6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

Beispiel 3: In diesem Beispiel wird mithilfe von EC2-Tags eine Instance mit einem Wartungsfenster registriert.

```
$option1 = @{Key="tag:Environment";Values=@("Production")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Production Web Servers" -ResourceType "INSTANCE"
```

Ausgabe:

```
2994977e-aefb-4a71-beac-df620352f184
```

- Einzelheiten zur API finden Sie unter RegisterTargetWithMaintenanceWindow in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`RegisterTaskWithMaintenanceWindow`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`RegisterTaskWithMaintenanceWindow`.

CLI

AWS CLI

Beispiel 1: Um eine Automatisierungsaufgabe mit einem Wartungsfenster zu registrieren

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Automatisierungsaufgabe mit einem Wartungsfenster registriert, das auf eine Instanz ausgerichtet ist.

```
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"
```

Ausgabe:

```
{
    "WindowTaskId":"11144444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So registrieren Sie eine Lambda-Aufgabe mit einem Wartungsfenster

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Lambda-Task mit einem Wartungsfenster registriert, das auf eine Instanz ausgerichtet ist.

```
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"
```

Ausgabe:

```
{
    "WindowTaskId":"22244444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: So registrieren Sie eine Run Command-Aufgabe mit einem Wartungsfenster

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Run Command-Aufgabe mit einem Wartungsfenster registriert, das auf eine Instanz ausgerichtet ist.

```
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
```

Ausgabe:

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 4: So registrieren Sie eine Step Functions Functions-Aufgabe mit einem Wartungsfenster

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Step Functions Functions-Aufgabe mit einem Wartungsfenster registriert, das auf eine Instanz ausgerichtet ist.

```
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"
```

Ausgabe:

```
{
    "WindowTaskId":"44444444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 5: So registrieren Sie eine Aufgabe mithilfe einer Windows-Wartungsziel-ID

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Aufgabe mithilfe einer Ziel-ID für das Wartungsfenster registriert. Die Ziel-ID des Wartungsfensters war in der Ausgabe des `aws ssm register-target-with-maintenance-window` Befehls enthalten. Sie können sie auch aus der Ausgabe des `aws ssm describe-maintenance-window-targets` Befehls abrufen.

```
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
```

Ausgabe:

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter RegisterTaskWithMaintenanceFenster in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Aufgabe mit einem Wartungsfenster unter Verwendung einer Instanz-ID registriert. Die Ausgabe ist die Task-ID.

```
$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
```

Ausgabe:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Beispiel 2: In diesem Beispiel wird eine Aufgabe mit einem Wartungsfenster unter Verwendung einer Ziel-ID registriert. Die Ausgabe ist die Task-ID.

```
$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
```

Ausgabe:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Beispiel 3: Dieses Beispiel erstellt ein Parameterobjekt für das Run-Befehlsdokument **AWS-RunPowerShellScript** und erstellt eine Aufgabe mit einem bestimmten Wartungsfenster unter Verwendung der Ziel-ID. Die Rückgabeausgabe ist die Aufgaben-ID.

```
$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
```

Ausgabe:

```
f1e2ef34-5678-12e3-456a-12334c5c6cbe
```

Beispiel 4: In diesem Beispiel wird eine AWS Systems Manager Automation-Aufgabe mithilfe eines Dokuments mit dem Namen registriert**`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"
```

- Einzelheiten zur API finden Sie unter RegisterTaskWithMaintenanceWindow in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **RemoveTagsFromResource** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`RemoveTagsFromResource`.

CLI

AWS CLI

Um ein Tag aus einer Patch-Baseline zu entfernen

Im folgenden `remove-tags-from-resource` Beispiel werden Tags aus einer Patch-Baseline entfernt.

```
aws ssm remove-tags-from-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0" \
    --tag-keys "Region"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter AWS Ressourcen taggen in der AWS allgemeinen Referenz.

- Einzelheiten zur API finden Sie RemoveTagsFromResourcein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Tag aus einem Wartungsfenster entfernt. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Remove-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -TagKey "Production"
```

- Einzelheiten zur API finden Sie unter RemoveTagsFromResource AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **SendCommand** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`SendCommand`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit Systems Manager

CLI

AWS CLI

Beispiel 1: Um einen Befehl auf einer oder mehreren Remote-Instances auszuführen

Im folgenden `send-command` Beispiel wird ein `echo` Befehl auf einer Zielinstanz ausgeführt.

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --parameters 'commands=["echo HelloWorld"]' \
    --targets "Key=instanceids,Values=i-1234567890abcdef0" \
    --comment "echo HelloWorld"
```

Ausgabe:

```
{
    "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
        }
    }
}
```

Weitere Informationen finden Sie unter Ausführen von Befehlen mit Systems Manager Run Command im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So rufen Sie IP-Informationen über eine Instanz ab

Im folgenden `send-command` Beispiel werden die IP-Informationen über eine Instanz abgerufen.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig"
```

In Beispiel 1 finden Sie eine Beispielausgabe.

Weitere Informationen finden Sie unter Ausführen von Befehlen mit Systems Manager Run Command im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: So führen Sie einen Befehl für Instanzen mit bestimmten Tags aus

Im folgenden `send-command` Beispiel wird ein Befehl auf Instanzen ausgeführt, die den Tag-Schlüssel „ENV“ und den Wert „Dev“ haben.

```
aws ssm send-command \
    --targets "Key=tag:ENV,Values=Dev" \
    --document-name "AWS-RunShellScript" \
    --parameters "commands=ifconfig"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Ausführen von Befehlen mit Systems Manager Run Command im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 4: So führen Sie einen Befehl aus, der SNS-Benachrichtigungen sendet

Im folgenden `send-command` Beispiel wird ein Befehl ausgeführt, der SNS-Benachrichtigungen für alle Benachrichtigungsereignisse und den `Command` Benachrichtigungstyp sendet.

```
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"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Ausführen von Befehlen mit Systems Manager Run Command im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 5: Um einen Befehl auszuführen, der an S3 ausgegeben wird und CloudWatch

Im folgenden `send-command` Beispiel wird ein Befehl ausgeführt, der Befehlsdetails an einen S3-Bucket und eine CloudWatch Logs-Protokollgruppe ausgibt.

```
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"
```

In Beispiel 1 finden Sie eine Beispielausgabe.

Weitere Informationen finden Sie unter Ausführen von Befehlen mit Systems Manager Run Command im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 6: So führen Sie Befehle auf mehreren Instanzen mit unterschiedlichen Tags aus

Im folgenden `send-command` Beispiel wird ein Befehl für Instanzen mit zwei verschiedenen Tag-Schlüsseln und -Werten ausgeführt.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Ausführen von Befehlen mit Systems Manager Run Command im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 7: So zielen Sie auf mehrere Instances mit demselben Tag-Schlüssel ab

Im folgenden `send-command` Beispiel wird ein Befehl für Instanzen ausgeführt, die denselben Tag-Schlüssel, aber unterschiedliche Werte haben.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev,Test
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Ausführen von Befehlen mit Systems Manager Run Command im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 8: So führen Sie einen Befehl aus, der ein geteiltes Dokument verwendet

Im folgenden `send-command` Beispiel wird ein gemeinsam verwendetes Dokument auf einer Zielinstanz ausgeführt.

```
aws ssm send-command \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
 \
    --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Verwenden gemeinsam genutzter SSM-Dokumente im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter SendCommand AWS CLIBefehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   // Sends a SSM command to a managed node.
   public static String sendSSMCommand(SsmClient ssmClient, String documentName,
String instanceId) throws InterruptedException {
       // Before we use Document to send a command - make sure it is active.
       boolean isDocumentActive = false;
       DescribeDocumentRequest request = DescribeDocumentRequest.builder()
           .name(documentName)
           .build();

       while (!isDocumentActive) {
           DescribeDocumentResponse response =
ssmClient.describeDocument(request);
```

```
            String documentStatus = response.document().statusAsString();
            if (documentStatus.equals("Active")) {
                System.out.println("The Systems Manager document is active and
ready to use.");
                isDocumentActive = true;
            } else {
                System.out.println("The Systems Manager document is not active.
Status: " + documentStatus);
                try {
                    // Add a delay to avoid making too many requests.
                    Thread.sleep(5000); // Wait for 5 seconds before checking
again
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }

        // Create the SendCommandRequest.
        SendCommandRequest commandRequest = SendCommandRequest.builder()
            .documentName(documentName)
            .instanceIds(instanceId)
            .build();

        // Send the command.
        SendCommandResponse commandResponse =
ssmClient.sendCommand(commandRequest);
        String commandId = commandResponse.command().commandId();
        System.out.println("The command Id is " + commandId);

        // Wait for the command execution to complete.
        GetCommandInvocationRequest invocationRequest =
GetCommandInvocationRequest.builder()
            .commandId(commandId)
            .instanceId(instanceId)
            .build();

        System.out.println("Wait 5 secs");
        TimeUnit.SECONDS.sleep(5);

        // Retrieve the command execution details.
        GetCommandInvocationResponse commandInvocationResponse =
ssmClient.getCommandInvocation(invocationRequest);
```

```
        // 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);
        }
        return commandId;
    }
```

- Einzelheiten zur API finden Sie SendCommandin der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Echo-Befehl auf einer Zielinstanz ausgeführt.

```
Send-SSMCommand -DocumentName "AWS-RunPowerShellScript" -Parameter @{commands =
 "echo helloWorld"} -Target @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
```

Ausgabe:

```
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}
```

Beispiel 2: Dieses Beispiel zeigt, wie ein Befehl ausgeführt wird, der verschachtelte Parameter akzeptiert.

```
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
```

- Einzelheiten zur API finden Sie unter SendCommand AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`StartAutomationExecution`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`StartAutomationExecution`.

CLI

AWS CLI

Beispiel 1: Um ein Automatisierungsdokument auszuführen

Im folgenden `start-automation-execution` Beispiel wird ein Automatisierungsdokument ausgeführt.

```
aws ssm start-automation-execution \
    --document-name "AWS-UpdateLinuxAmi" \
    --parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

Ausgabe:

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

Weitere Informationen finden Sie unter Manuelles Ausführen eines Automatisierungsworkflows im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So führen Sie ein gemeinsam genutztes Automatisierungsdokument aus

Im folgenden `start-automation-execution` Beispiel wird ein gemeinsam genutztes Automatisierungsdokument ausgeführt.

```
aws ssm start-automation-execution \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

Ausgabe:

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

Weitere Informationen finden Sie unter Verwenden gemeinsam genutzter SSM-Dokumente im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter StartAutomationAusführung in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird ein Dokument ausgeführt, das eine Automatisierungsrolle, eine AMI-Quell-ID und eine Amazon EC2 EC2-Instance-Rolle angibt.

```
Start-SSMAutomationExecution -DocumentName AWS-UpdateLinuxAmi -
Parameter @{'AutomationAssumeRole'='arn:aws:iam::123456789012:role/
SSMAutomationRole';'SourceAmiId'='ami-
f173cc91';'InstanceIamRole'='EC2InstanceRole'}
```

Ausgabe:

```
3a532a4f-0382-11e7-9df7-6f11185f6dd1
```

- Einzelheiten zur API finden Sie unter StartAutomationExecution in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`StopAutomationExecution`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`StopAutomationExecution`.

CLI

AWS CLI

Um eine Automatisierungsausführung zu beenden

Im folgenden `stop-automation-execution` Beispiel wird ein Automatisierungsdokument gestoppt.

```
aws ssm stop-automation-execution
    --automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Manuelles Ausführen eines Automatisierungs-Workflows im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter StopAutomationAusführung in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Automatisierungsausführung gestoppt. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Stop-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

- Einzelheiten zur API finden Sie unter StopAutomationExecution in AWS Tools for PowerShell Cmdlet Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`UpdateAssociation`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`UpdateAssociation`.

CLI

AWS CLI

Beispiel 1: So aktualisieren Sie eine Dokumentverknüpfung

Im folgenden `update-association` Beispiel wird eine Verknüpfung mit einer neuen Dokumentversion aktualisiert.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --document-version "\$LATEST"
```

Ausgabe:

```
{
    "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
    }
}
```

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So aktualisieren Sie den Zeitplanausdruck einer Assoziation

Im folgenden `update-association` Beispiel wird der Zeitplanausdruck für die angegebene Zuordnung aktualisiert.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --schedule-expression "cron(0 0 0/4 1/1 * ? *)"
```

Ausgabe:

```
{
    "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
    }
}
```

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie UpdateAssociationunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Verknüpfung mit einer neuen Dokumentversion aktualisiert.

```
Update-SSMAssociation -AssociationId "93285663-92df-44cb-9f26-2292d4ecc439" -
DocumentVersion "1"
```

Ausgabe:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           :
Date                 : 3/1/2017 6:22:21 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

- Einzelheiten zur API finden Sie unter UpdateAssociation AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **UpdateAssociationStatus** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`UpdateAssociationStatus`.

CLI

AWS CLI

Um den Zuordnungsstatus zu aktualisieren

Im folgenden `update-association-status` Beispiel wird der Zuordnungsstatus der Verknüpfung zwischen einer Instanz und einem Dokument aktualisiert.

```
aws ssm update-association-status \
    --name "AWS-UpdateSSMAgent" \
    --instance-id "i-1234567890abcdef0" \
    --association-status
 "Date=1424421071.939,Name=Pending,Message=temp_status_change,AdditionalInfo=Additional-
Config-Needed"
```

Ausgabe:

```
{
    "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
    }
}
```

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter UpdateAssociationStatus in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird der Zuordnungsstatus der Zuordnung zwischen einer Instanz und einem Konfigurationsdokument aktualisiert.

```
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"
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter UpdateAssociationStatus in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`UpdateDocument`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`UpdateDocument`.

CLI

AWS CLI

Um eine neue Version eines Dokuments zu erstellen

Das folgende `update-document` Beispiel erstellt eine neue Version eines Dokuments, wenn es auf einem Windows-Computer ausgeführt wird. Das von angegebene Dokument `--document` muss im JSON-Format vorliegen. Beachten Sie, dass darauf verwiesen werden `file://` muss, gefolgt vom Pfad der Inhaltsdatei. Aufgrund der Tatsache, dass der `--document-version` Parameter $ am Anfang steht, müssen Sie unter Windows den Wert in doppelte Anführungszeichen setzen. Unter Linux, macOS oder an einer PowerShell Eingabeaufforderung müssen Sie den Wert in einfache Anführungszeichen setzen.

Windows-Version:

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version "$LATEST"
```

Linux/Mac-Version:

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version '$LATEST'
```

Ausgabe:

```
{
  "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"
  }
}
```

- Einzelheiten zur API finden Sie UpdateDocumentin AWS CLI der Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dadurch wird eine neue Version eines Dokuments mit dem aktualisierten Inhalt der von Ihnen angegebenen JSON-Datei erstellt. Das Dokument muss im JSON-Format sein. Sie können die Dokumentversion mit dem Cmdlet „Get-SSM DocumentVersion List“ abrufen.

```
Update-SSMDocument -Name RunShellScript -DocumentVersion "1" -Content (Get-
Content -Raw "c:\temp\RunShellScript.json")
```

Ausgabe:

```
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
```

- Einzelheiten zur API finden Sie unter UpdateDocumentCmdlet-Referenz.AWS Tools for PowerShell

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`UpdateDocumentDefaultVersion`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`UpdateDocumentDefaultVersion`.

CLI

AWS CLI

Um die Standardversion eines Dokuments zu aktualisieren

Im folgenden `update-document-default-version` Beispiel wird die Standardversion eines Systems Manager Manager-Dokuments aktualisiert.

```
aws ssm update-document-default-version \
    --name "Example" \
```

```
    --document-version "2"
```

Ausgabe:

```
{
    "Description": {
        "Name": "Example",
        "DefaultVersion": "2"
    }
}
```

Weitere Informationen finden Sie unter Writing SSM Document Content im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter UpdateDocumentDefaultVersion AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dadurch wird die Standardversion eines Dokuments aktualisiert. Sie können die verfügbaren Dokumentversionen mit dem Cmdlet „Get-SSM DocumentVersion List“ abrufen.

```
Update-SSMDocumentDefaultVersion -Name "RunShellScript" -DocumentVersion "2"
```

Ausgabe:

```
DefaultVersion Name
-------------- ----
2              RunShellScript
```

- Einzelheiten zur API finden Sie unter UpdateDocumentDefaultVersionCmdlet-Referenz.AWS Tools for PowerShell

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **UpdateMaintenanceWindow** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`UpdateMaintenanceWindow`.

CLI

AWS CLI

Beispiel 1: Um ein Wartungsfenster zu aktualisieren

Im folgenden `update-maintenance-window` Beispiel wird der Name eines Wartungsfensters aktualisiert.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --name "My-Renamed-MW"
```

Ausgabe:

```
{
    "Cutoff": 1,
    "Name": "My-Renamed-MW",
    "Schedule": "cron(0 16 ? * TUE *)",
    "Enabled": true,
    "AllowUnassociatedTargets": true,
    "WindowId": "mw-1a2b3c4d5e6f7g8h9",
    "Duration": 4
}
```

Beispiel 2: Um ein Wartungsfenster zu deaktivieren

Das folgende `update-maintenance-window` Beispiel deaktiviert ein Wartungsfenster.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --no-enabled
```

Beispiel 3: Um ein Wartungsfenster zu aktivieren

Das folgende `update-maintenance-window` Beispiel aktiviert ein Wartungsfenster.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --enabled
```

Weitere Informationen finden Sie unter Aktualisieren eines Wartungsfensters (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter UpdateMaintenanceFenster in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   // Update the maintenance window schedule
   public static void updateSSMMaintenanceWindow(SsmClient ssmClient, String id,
String name) {
       try {
           UpdateMaintenanceWindowRequest updateRequest =
UpdateMaintenanceWindowRequest.builder()
                .windowId(id)
                .allowUnassociatedTargets(true)
                .duration(24)
                .enabled(true)
                .name(name)
                .schedule("cron(0 0 ? * MON *)")
                .build();

           ssmClient.updateMaintenanceWindow(updateRequest);
           System.out.println("The Systems Manager maintenance window was
successfully updated.");

       } catch (SsmException e) {
           System.err.println(e.getMessage());
           System.exit(1);
```

```
        }
    }
```

- Einzelheiten zur API finden Sie unter UpdateMaintenanceFenster in der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird der Name eines Wartungsfensters aktualisiert.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Name "My-Renamed-
MW"
```

Ausgabe:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

Beispiel 2: In diesem Beispiel wird ein Wartungsfenster aktiviert.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $true
```

Ausgabe:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

Beispiel 3: In diesem Beispiel wird ein Wartungsfenster deaktiviert.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $false
```

Ausgabe:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : False
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

- Einzelheiten zur API finden Sie unter UpdateMaintenanceFenster in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`UpdateManagedInstanceRole`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`UpdateManagedInstanceRole`.

CLI

AWS CLI

Um die IAM-Rolle einer verwalteten Instanz zu aktualisieren

Im folgenden `update-managed-instance-role` Beispiel wird das IAM-Instanzprofil einer verwalteten Instanz aktualisiert.

```
aws ssm update-managed-instance-role \
    --instance-id "mi-08ab247cdfEXAMPLE" \
    --iam-role "ExampleRole"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Schritt 4: Erstellen eines IAM-Instanzprofils für Systems Manager im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter UpdateManagedInstanceRole AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die Rolle einer verwalteten Instanz aktualisiert. Es erfolgt keine Ausgabe, wenn der Befehl erfolgreich ist.

```
Update-SSMManagedInstanceRole -InstanceId "mi-08ab247cdf1046573" -IamRole
 "AutomationRole"
```

- Einzelheiten zur API finden Sie unter UpdateManagedInstanceRole AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`UpdateOpsItem`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`UpdateOpsItem`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit Systems Manager

CLI

AWS CLI

Um ein zu aktualisieren OpsItem

Im folgenden `update-ops-item` Beispiel werden die Beschreibung, Priorität und Kategorie für ein aktualisiert OpsItem. Darüber hinaus gibt der Befehl ein SNS-Thema an, an das die Benachrichtigungen gesendet werden, wenn dieses bearbeitet oder geändert OpsItem wird.

```
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"
```

Ausgabe:

```
This command produces no output.
```

Weitere Informationen finden Sie unter Arbeiten mit OpsItems im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter UpdateOpsElement in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public static void resolveOpsItem(SsmClient ssmClient, String opsID) {
        try {
            UpdateOpsItemRequest opsItemRequest = UpdateOpsItemRequest.builder()
                .opsItemId(opsID)
                .status(OpsItemStatus.RESOLVED)
                .build();

            ssmClient.updateOpsItem(opsItemRequest);

        } catch (SsmException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie unter UpdateOpsArtikel in der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterSystems Manager mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **UpdatePatchBaseline** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`UpdatePatchBaseline`.

CLI

AWS CLI

Beispiel 1: Um eine Patch-Baseline zu aktualisieren

Im folgenden `update-patch-baseline` Beispiel werden der angegebenen Patch-Baseline die beiden angegebenen Patches als abgelehnt und ein Patch als genehmigt hinzugefügt.

```
aws ssm update-patch-baseline \
        --baseline-id "pb-0123456789abcdef0" \
        --rejected-patches "KB2032276" "MS10-048" \
        --approved-patches "KB2124261"
```

Ausgabe:

```
{
    "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": []
}
```

Beispiel 2: Um eine Patch-Baseline umzubenennen

Im folgenden `update-patch-baseline` Beispiel wird die angegebene Patch-Baseline umbenannt.

```
aws ssm update-patch-baseline \
    --baseline-id "pb-0713accee01234567" \
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

Weitere Informationen finden Sie unter Aktualisieren oder Löschen einer Patch-Baseline`__ im Systems AWS Manager Manager-Benutzerhandbuch. < https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-baseline-update-or-delete.html>

- Einzelheiten zur API finden Sie unter UpdatePatchBaseline in der Befehlsreferenz.AWS CLI

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel werden einer vorhandenen Patch-Baseline zwei Patches als abgelehnt und ein Patch als genehmigt hinzugefügt.

```
Update-SSMPatchBaseline -BaselineId "pb-03da896ca3b68b639" -RejectedPatch
 "KB2032276","MS10-048" -ApprovedPatch "KB2124261"
```

Ausgabe:

```
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}
```

- Einzelheiten zur API finden Sie unter UpdatePatchBaseline in der AWS Tools for PowerShell Cmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Systems Manager mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Szenarien für Systems Manager mit AWS SDKs

Die folgenden Codebeispiele zeigen Ihnen, wie Sie allgemeine Szenarien in Systems Manager mit AWS SDKs implementieren. Diese Szenarien zeigen Ihnen, wie Sie bestimmte Aufgaben ausführen,

indem Sie mehrere Funktionen in Systems Manager aufrufen. Jedes Szenario enthält einen Link zu GitHub, über den Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Beispiele

- Erste Schritte mit Systems Manager mithilfe eines AWS SDK

Erste Schritte mit Systems Manager mithilfe eines AWS SDK

Das folgende Codebeispiel zeigt, wie Sie mit Systems Manager Manager-Wartungsfenstern, Dokumenten und arbeiten OpsItems.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ssm.SsmClient;
import software.amazon.awssdk.services.ssm.model.CommandInvocation;
import software.amazon.awssdk.services.ssm.model.CommandInvocationStatus;
import software.amazon.awssdk.services.ssm.model.CreateDocumentRequest;
import software.amazon.awssdk.services.ssm.model.CreateDocumentResponse;
import software.amazon.awssdk.services.ssm.model.CreateMaintenanceWindowRequest;
import software.amazon.awssdk.services.ssm.model.CreateMaintenanceWindowResponse;
import software.amazon.awssdk.services.ssm.model.CreateOpsItemRequest;
import software.amazon.awssdk.services.ssm.model.CreateOpsItemResponse;
import software.amazon.awssdk.services.ssm.model.DeleteDocumentRequest;
import software.amazon.awssdk.services.ssm.model.DeleteMaintenanceWindowRequest;
import software.amazon.awssdk.services.ssm.model.DeleteOpsItemRequest;
import software.amazon.awssdk.services.ssm.model.DescribeDocumentRequest;
import software.amazon.awssdk.services.ssm.model.DescribeDocumentResponse;
import
 software.amazon.awssdk.services.ssm.model.DescribeMaintenanceWindowsRequest;
import
 software.amazon.awssdk.services.ssm.model.DescribeMaintenanceWindowsResponse;
import software.amazon.awssdk.services.ssm.model.DescribeOpsItemsRequest;
```

```
import software.amazon.awssdk.services.ssm.model.DescribeOpsItemsResponse;
import software.amazon.awssdk.services.ssm.model.DocumentAlreadyExistsException;
import software.amazon.awssdk.services.ssm.model.DocumentType;
import software.amazon.awssdk.services.ssm.model.GetCommandInvocationRequest;
import software.amazon.awssdk.services.ssm.model.GetCommandInvocationResponse;
import software.amazon.awssdk.services.ssm.model.GetOpsItemRequest;
import software.amazon.awssdk.services.ssm.model.GetOpsItemResponse;
import software.amazon.awssdk.services.ssm.model.ListCommandInvocationsRequest;
import software.amazon.awssdk.services.ssm.model.ListCommandInvocationsResponse;
import software.amazon.awssdk.services.ssm.model.MaintenanceWindowFilter;
import software.amazon.awssdk.services.ssm.model.MaintenanceWindowIdentity;
import software.amazon.awssdk.services.ssm.model.OpsItemDataValue;
import software.amazon.awssdk.services.ssm.model.OpsItemFilter;
import software.amazon.awssdk.services.ssm.model.OpsItemFilterKey;
import software.amazon.awssdk.services.ssm.model.OpsItemFilterOperator;
import software.amazon.awssdk.services.ssm.model.OpsItemStatus;
import software.amazon.awssdk.services.ssm.model.OpsItemSummary;
import software.amazon.awssdk.services.ssm.model.SendCommandRequest;
import software.amazon.awssdk.services.ssm.model.SendCommandResponse;
import software.amazon.awssdk.services.ssm.model.SsmException;
import software.amazon.awssdk.services.ssm.model.UpdateMaintenanceWindowRequest;
import software.amazon.awssdk.services.ssm.model.UpdateOpsItemRequest;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;

/**
 * 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/setup.html
 *
 *
 * This Java program performs these tasks:
 * 1. Creates an AWS Systems Manager maintenance window with a default name or a
 user-provided name.
 * 2. Modifies the maintenance window schedule.
```

```
 * 3. Creates a Systems Manager document with a default name or a user-provided
 name.
 * 4. Sends a command to a specified EC2 instance using the created Systems
 Manager document and displays the time when the command was invoked.
 * 5. Creates a Systems Manager OpsItem with a predefined title, source,
 category, and severity.
 * 6. Updates and resolves the created OpsItem.
 * 7. Deletes the Systems Manager maintenance window, OpsItem, and document.
 */

public class SSMScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");
    public static void main(String[] args) throws InterruptedException {
        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);
        String documentName;
        String windowName;
        String instanceId = args[0];
        String title = "Disk Space Alert" ;
        String source = "EC2" ;
        String category = "Performance" ;
        String severity = "2" ;
```

```
        Region region = Region.US_EAST_1;
        SsmClient ssmClient = SsmClient.builder()
            .region(region)
            .build();

        System.out.println(DASHES);
        System.out.println("""
            Welcome to the AWS Systems Manager SDK Getting Started scenario.
            This program demonstrates how to interact with Systems Manager using
the AWS SDK for Java (v2).
            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...
            Please hit Enter
            """);
        scanner.nextLine();
        System.out.println(DASHES);

        System.out.println("Create a Systems Manager 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 = createMaintenanceWindow(ssmClient, windowName);
        System.out.println(DASHES);

        System.out.println("Modify the maintenance window by changing the
schedule");
        System.out.println("Please hit Enter");
        scanner.nextLine();
        updateSSMMaintenanceWindow(ssmClient, winId, windowName);
        System.out.println(DASHES);

        System.out.println("Create a document that defines the actions that
Systems Manager performs on your EC2 instance.");
        System.out.println("Please enter the document name (default is
ssmdocument):");
        String doc = scanner.nextLine();
        documentName = doc.isEmpty() ? "ssmdocument" : doc;
```

```
        createSSMDoc(ssmClient, documentName);

        System.out.println("Now we are going to run a command on an EC2 instance
that echoes 'Hello, world!'");
        System.out.println("Please hit Enter");
        scanner.nextLine();
        String commandId = sendSSMCommand(ssmClient, documentName, instanceId);
        System.out.println(DASHES);

        System.out.println("Lets get the time when the specific command was sent
to the specific managed node");
        System.out.println("Please hit Enter");
        scanner.nextLine();
        displayCommands(ssmClient, commandId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("""
             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.
            """);

        System.out.println("Please hit Enter");
        scanner.nextLine();
        String opsItemId = createSSMOpsItem(ssmClient, title, source, category,
severity);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Now we will update  the OpsItem "+opsItemId);
        System.out.println("Please hit Enter");
        scanner.nextLine();
        String description = "An update to "+opsItemId ;
        updateOpsItem(ssmClient, opsItemId, title, description);
```

```
        System.out.println("Now we will get the status of the OpsItem
"+opsItemId);
        System.out.println("Please hit Enter");
        scanner.nextLine();
        describeOpsItems(ssmClient, opsItemId);
        System.out.println("Now we will resolve the OpsItem "+opsItemId);
        System.out.println("Please hit Enter");
        scanner.nextLine();
        resolveOpsItem(ssmClient, opsItemId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Would you like to delete the Systems Manager
resources? (y/n)");
        String delAns = scanner.nextLine().trim();
        if (delAns.equalsIgnoreCase("y")) {
            System.out.println("You selected to delete the resources.");
            System.out.print("Press Enter to continue...");
            scanner.nextLine();
            deleteOpsItem(ssmClient, opsItemId);
            deleteMaintenanceWindow(ssmClient, winId);
            deleteDoc(ssmClient, documentName);
        } else {
            System.out.println("The Systems Manager resources will not be
deleted");
        }
        System.out.println(DASHES);

        System.out.println("This concludes the Systems Manager SDK Getting
Started scenario.");
        System.out.println(DASHES);
    }

    // Displays the date and time when the specific command was invoked.
    public static void displayCommands(SsmClient ssmClient, String commandId) {
        try {
            ListCommandInvocationsRequest commandInvocationsRequest =
ListCommandInvocationsRequest.builder()
                .commandId(commandId)
                .build();

            ListCommandInvocationsResponse response =
ssmClient.listCommandInvocations(commandInvocationsRequest);
            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()));
            }

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Create an SSM OpsItem
    public static String createSSMOpsItem(SsmClient ssmClient, String title,
String source, String category, String severity) {
        try {
            CreateOpsItemRequest opsItemRequest = CreateOpsItemRequest.builder()
                .description("Created by the Systems Manager Java API")
                .title(title)
                .source(source)
                .category(category)
                .severity(severity)
                .build();

            CreateOpsItemResponse itemResponse =
ssmClient.createOpsItem(opsItemRequest);
            return itemResponse.opsItemId();

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
       return "";
    }

    // Update the AWS SSM OpsItem.
    public static void updateOpsItem(SsmClient ssmClient, 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());
```

```
        try {
            UpdateOpsItemRequest request = UpdateOpsItemRequest.builder()
                .opsItemId(opsItemId)
                .title(title)
                .operationalData(operationalData)
                .status(getOpsItem(ssmClient, opsItemId))
                .description(description)
                .build();

            ssmClient.updateOpsItem(request);

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void resolveOpsItem(SsmClient ssmClient, String opsID) {
        try {
            UpdateOpsItemRequest opsItemRequest = UpdateOpsItemRequest.builder()
                .opsItemId(opsID)
                .status(OpsItemStatus.RESOLVED)
                .build();

            ssmClient.updateOpsItem(opsItemRequest);

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Gets a specific OpsItem.
    private static OpsItemStatus getOpsItem(SsmClient ssmClient, String
opsItemId) {
        GetOpsItemRequest itemRequest = GetOpsItemRequest.builder()
            .opsItemId(opsItemId)
            .build();

        try {
            GetOpsItemResponse response = ssmClient.getOpsItem(itemRequest);
            return response.opsItem().status();
```

```
        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return null;
    }

    // Sends a SSM command to a managed node.
    public static String sendSSMCommand(SsmClient ssmClient, String documentName,
String instanceId) throws InterruptedException {
        // Before we use Document to send a command - make sure it is active.
        boolean isDocumentActive = false;
        DescribeDocumentRequest request = DescribeDocumentRequest.builder()
            .name(documentName)
            .build();

        while (!isDocumentActive) {
            DescribeDocumentResponse response =
ssmClient.describeDocument(request);
            String documentStatus = response.document().statusAsString();
            if (documentStatus.equals("Active")) {
                System.out.println("The Systems Manager document is active and
ready to use.");
                isDocumentActive = true;
            } else {
                System.out.println("The Systems Manager document is not active.
Status: " + documentStatus);
                try {
                    // Add a delay to avoid making too many requests.
                    Thread.sleep(5000); // Wait for 5 seconds before checking
again
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }

        // Create the SendCommandRequest.
        SendCommandRequest commandRequest = SendCommandRequest.builder()
            .documentName(documentName)
            .instanceIds(instanceId)
            .build();

        // Send the command.
```

```
        SendCommandResponse commandResponse =
ssmClient.sendCommand(commandRequest);
        String commandId = commandResponse.command().commandId();
        System.out.println("The command Id is " + commandId);

        // Wait for the command execution to complete.
        GetCommandInvocationRequest invocationRequest =
GetCommandInvocationRequest.builder()
            .commandId(commandId)
            .instanceId(instanceId)
            .build();

        System.out.println("Wait 5 secs");
        TimeUnit.SECONDS.sleep(5);

        // Retrieve the command execution details.
        GetCommandInvocationResponse commandInvocationResponse =
ssmClient.getCommandInvocation(invocationRequest);

        // 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);
        }
        return commandId;
    }

    // Deletes an AWS Systems Manager document.
    public static void deleteDoc(SsmClient ssmClient, String documentName) {
        try {
            DeleteDocumentRequest documentRequest =
DeleteDocumentRequest.builder()
                .name(documentName)
                .build();

            ssmClient.deleteDocument(documentRequest);
            System.out.println("The Systems Manager document was successfully
deleted.");

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
```

```
        }
    }

    public static void deleteMaintenanceWindow(SsmClient ssmClient, String winId)
{
        try {
            DeleteMaintenanceWindowRequest windowRequest =
DeleteMaintenanceWindowRequest.builder()
                .windowId(winId)
                .build();

            ssmClient.deleteMaintenanceWindow(windowRequest);
            System.out.println("The maintenance window was successfully
deleted.");

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Update the maintenance window schedule
    public static void updateSSMMaintenanceWindow(SsmClient ssmClient, String id,
String name) {
        try {
            UpdateMaintenanceWindowRequest updateRequest =
UpdateMaintenanceWindowRequest.builder()
                .windowId(id)
                .allowUnassociatedTargets(true)
                .duration(24)
                .enabled(true)
                .name(name)
                .schedule("cron(0 0 ? * MON *)")
                .build();

            ssmClient.updateMaintenanceWindow(updateRequest);
            System.out.println("The Systems Manager maintenance window was
successfully updated.");

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

```
   public static String createMaintenanceWindow(SsmClient ssmClient, String
winName) {
       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();

       try {
           CreateMaintenanceWindowResponse response =
ssmClient.createMaintenanceWindow(request);
           String maintenanceWindowId = response.windowId();
           System.out.println("The maintenance window id is " +
maintenanceWindowId);
           return maintenanceWindowId;

       } catch (DocumentAlreadyExistsException e) {
           System.err.println("The maintenance window already exists. Moving
on.");
       } catch (SsmException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }

       MaintenanceWindowFilter filter = MaintenanceWindowFilter.builder()
           .key("name")
           .values(winName)
           .build();

       DescribeMaintenanceWindowsRequest winRequest =
DescribeMaintenanceWindowsRequest.builder()
           .filters(filter)
           .build();

       String windowId = "";
       DescribeMaintenanceWindowsResponse response =
ssmClient.describeMaintenanceWindows(winRequest);
       List<MaintenanceWindowIdentity> windows = response.windowIdentities();
       if (!windows.isEmpty()) {
```

```
            windowId = windows.get(0).windowId();
            System.out.println("Window ID: " + windowId);
        } else {
            System.out.println("Window not found.");
        }
       return windowId;
   }

   // Create an AWS SSM document to use in this scenario.
   public static void createSSMDoc(SsmClient ssmClient, String docName) {
        // Create JSON for the content
        String jsonData = """
            {
                "schemaVersion": "2.2",
                "description": "Run a simple shell command",
                "mainSteps": [
                    {
                        "action": "aws:runShellScript",
                        "name": "runEchoCommand",
                        "inputs": {
                          "runCommand": [
                            "echo 'Hello, world!'"
                          ]
                        }
                      }
                    ]
                }
            """;

        try {
            CreateDocumentRequest request = CreateDocumentRequest.builder()
                .content(jsonData)
                .name(docName)
                .documentType(DocumentType.COMMAND)
                .build();

            // Create the document.
            CreateDocumentResponse response = ssmClient.createDocument(request);
            System.out.println("The status of the document is " +
response.documentDescription().status());

        } catch (DocumentAlreadyExistsException e) {
            System.err.println("The document already exists. Moving on." );
        } catch (SsmException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeOpsItems(SsmClient ssmClient, String key) {
        try {
            OpsItemFilter filter = OpsItemFilter.builder()
                .key(OpsItemFilterKey.OPS_ITEM_ID)
                .values(key)
                .operator(OpsItemFilterOperator.EQUAL)
                .build();

            DescribeOpsItemsRequest itemsRequest =
DescribeOpsItemsRequest.builder()
                .maxResults(10)
                .opsItemFilters(filter)
                .build();

            DescribeOpsItemsResponse itemsResponse =
ssmClient.describeOpsItems(itemsRequest);
            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());
            }

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void deleteOpsItem(SsmClient ssmClient, String opsId) {
        try {
            DeleteOpsItemRequest deleteOpsItemRequest =
DeleteOpsItemRequest.builder()
                .opsItemId(opsId)
                .build();

            ssmClient.deleteOpsItem(deleteOpsItemRequest);
            System.out.println(opsId +" Opsitem was deleted");

        } catch (SsmException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- API-Details finden Sie in den folgenden Themen der AWS SDK for Java 2.x -API-Referenz.
  - CommandInvocations
  - CreateDocument
  - CreateMaintenanceFenster
  - CreateOpsArtikel
  - DeleteMaintenanceFenster
  - SendCommand
  - UpdateOpsArtikel

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterSystems Manager mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Überwachung AWS Systems Manager

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit AWS Systems Manager und Leistung Ihrer AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etappenübergreifenden Ausfall debuggen können. Bevor Sie mit der Überwachung von Systems Manager beginnen, sollten Sie einen Überwachungsplan mit Antworten auf die folgenden Fragen erstellen:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen werden überwacht?
- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungstools werden verwendet?
- Wer führt die Überwachungsaufgaben aus?
- Wer soll benachrichtigt werden, wenn Fehler auftreten?

Nachdem Sie Ihre Überwachungsziele festgelegt und Ihren Überwachungsplan erstellt haben, legen Sie im nächsten Schritt einen Ausgangswert für normale Systems Manager-Leistung in Ihrer Umgebung fest. Sie sollten die Systems Manager-Leistung zu verschiedenen Zeiten und unter verschiedenen Belastungsbedingungen messen. Wenn Sie Systems Manager überwachen, sollten Sie einen Verlauf der von Ihnen gesammelten Überwachungsdaten speichern. Sie können die aktuelle Systems Manager-Leistung mit diesen historischen Daten zur Identifikation normaler Leistungsmuster und Leistungsanomalien sowie zur Erstellung von Verfahren für deren Handhabung vergleichen.

Beispielsweise können Sie die Erfolge oder Fehlschläge von Vorgängen wie Automation-Workflows, die Anwendung von Patch-Baselines, Wartungsfensterereignisse und die Konfigurations-Compliance überwachen. Automatisierung ist eine Fähigkeit von. AWS Systems Manager

Sie können auch die CPU-Auslastung, die Festplatten-I/O und die Netzwerkauslastung Ihrer verwalteten Knoten überwachen. Wenn die Leistung außerhalb der festgelegten Grundwerte liegt, müssen Sie den Knoten eventuell neu konfigurieren oder optimieren, um die CPU-Nutzung zu verringern, die Festplatten-I/O zu verbessern oder den Netzwerkverkehr zu reduzieren. Weitere Informationen zur Überwachung von EC2-Instances finden Sie unter Monitor Amazon EC2 im Amazon EC2 EC2-Benutzerhandbuch.

Themen

- Überwachungstools
- Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent)
- Senden von SSM Agent-Protokollen an CloudWatch Logs
- Überwachung der Ereignisse Ihrer Änderungsanfragen
- Überwachung Ihrer Automatisierungen
- Überwachen von Run Command-Metriken mit Amazon CloudWatch
- AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail
- Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs
- Konfiguration von Amazon CloudWatch Logs für Run Command
- Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge
- Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen

Überwachungstools

Der Inhalt dieses Kapitels enthält Informationen zur Verwendung der Tools, die zur Überwachung Ihres Systems Manager und anderer AWS Ressourcen zur Verfügung stehen. Eine vollständigere Liste der Tools finden Sie unter Protokollieren und Überwachen in AWS Systems Manager.

Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent)

Sie können den CloudWatch Amazon-Agenten konfigurieren und verwenden, um Metriken und Protokolle von Ihren Knoten zu sammeln, anstatt AWS Systems Manager Agent (SSM Agent) für diese Aufgaben zu verwenden. Der CloudWatch Agent ermöglicht es Ihnen, mehr Metriken auf EC2-Instances zu sammeln, als mitSSM Agent. Darüber hinaus können Sie mit dem Agenten Metriken von lokalen Servern sammeln. CloudWatch

Sie können die Agentenkonfigurationseinstellungen auch im Systems Manager speichern, Parameter Store um sie mit dem CloudWatch Agenten zu verwenden. Parameter Storeist eine Fähigkeit von AWS Systems Manager.

> **Note**
>
> AWS Systems Manager unterstützt nur unter 64-Bit-Versionen von Windows die Migration vom SSM Agent Unified CloudWatch Agent zum Sammeln von Protokollen und Messdaten. Informationen zur Einrichtung des Unified CloudWatch Agents auf anderen Betriebssystemen und vollständige Informationen zur Verwendung des CloudWatch Agenten finden Sie unter Sammeln von Metriken und Protokollen von Amazon EC2 EC2-Instances und lokalen Servern mit dem CloudWatch Agenten im _CloudWatch Amazon-Benutzerhandbuch.
> Sie können den CloudWatch Agenten auf anderen unterstützten Betriebssystemen verwenden, aber Sie können Systems Manager nicht verwenden, um eine Tool-Migration durchzuführen.

SSM Agent schreibt Informationen zu Ausführungen, geplanten Aktionen, Fehlern und dem Zustandsstatus in Protokolldateien auf jedem Knoten. Die manuelle Verbindung mit einem Knoten, um Protokolldateien anzuzeigen und zum Beheben eines Problems mit dem SSM Agent ist zeitaufwendig. Für eine effizientere Knotenüberwachung können Sie entweder SSM Agent sich selbst oder den CloudWatch Agenten so konfigurieren, dass er diese Protokolldaten an Amazon CloudWatch Logs sendet.

> **Important**
>
> Der Unified CloudWatch Agent wurde SSM Agent als Tool zum Senden von Protokolldaten an Amazon CloudWatch Logs ersetzt. Das SSM Agent-aws:cloudWatch-Plugin wird nicht unterstützt. Wir empfehlen, nur den Unified CloudWatch Agent für Ihre Protokollerfassungsprozesse zu verwenden. Weitere Informationen finden Sie unter den folgenden Themen:
>
> - Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent)
> - Migrieren Sie die Erfassung von Windows Server-Knotenprotokollen auf den CloudWatch Agenten
> - Erfassung von Metriken, Protokollen und Traces mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.

Mithilfe von CloudWatch Logs können Sie Protokolldaten in Echtzeit überwachen, Protokolldaten suchen und filtern, indem Sie einen oder mehrere Metrikfilter erstellen, und historische Daten

archivieren und abrufen, wenn Sie sie benötigen. Weitere Informationen zu CloudWatch Logs finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

Die Konfiguration eines Agenten zum Senden von Protokolldaten an Amazon CloudWatch Logs bietet die folgenden Vorteile:

- Zentrale Speicherung von Protokolldateien alle SSM Agent-Protokolldateien.
- Schnellerer Zugriff auf Dateien zur Fehleranalyse.
- Unbegrenzte Protokolldatei-Aufbewahrung (konfigurierbar).
- Die Verwaltung und der Zugriff auf Protokolle ist unabhängig vom Status des Knotens möglich.
- Zugriff auf andere CloudWatch Funktionen wie Metriken und Alarme.

Informationen zur Überwachung von Session Manager-Aktivitäten finden Sie unter Prüfen von Sitzungsaktivitäten und Protokollierung von Sitzungsaktivitäten aktivieren und deaktivieren.

Migrieren Sie die Erfassung von Windows Server-Knotenprotokollen auf den CloudWatch Agenten

Wenn Sie SSM Agent auf unterstützten Windows Server Knoten SSM Agent Protokolldateien an Amazon CloudWatch Logs senden, können Sie Systems Manager verwenden, SSM Agent um vom CloudWatch Agenten als Protokollerfassungstool zu migrieren und Ihre Konfigurationseinstellungen zu migrieren.

Der CloudWatch Agent wird auf 32-Bit-Versionen von nicht unterstütztWindows Server.

Bei 64-Bit-EC2-Instances für Windows Server können Sie die Migration zum CloudWatch Agenten automatisch oder manuell durchführen. Bei On-Premises-Servern und virtuellen Maschinen muss der Prozess manuell ausgeführt werden.

> **Note**
>
> Während des Migrationsprozesses werden die an gesendeten Daten CloudWatch möglicherweise unterbrochen oder dupliziert. Ihre Metriken und Protokolldaten werden CloudWatch nach Abschluss der Migration erneut korrekt aufgezeichnet.

Wir empfehlen, die Migration auf einer begrenzten Anzahl von Knoten zu testen, bevor eine gesamte Flotte auf den CloudWatch Agenten migriert wird. Nach der Migration können Sie die Protokollerfassung wieder mit SSM Agent ausführen, wenn Sie dies bevorzugen.

Important

In den folgenden Fällen können Sie mit den in diesem Thema beschriebenen Schritten nicht zum CloudWatch Agenten migrieren:

- Die bestehende Konfiguration für SSM Agent gibt mehrere Regionen an.
- Die bestehende Konfiguration für SSM Agent gibt mehrere Sätze von Anmeldeinformationen für Zugriffsschlüssel und geheime Schlüssel an.

In diesen Fällen ist es erforderlich, die Protokollerfassung zu deaktivieren SSM Agent und den CloudWatch Agenten ohne Migrationsprozess zu installieren. Weitere Informationen finden Sie in den folgenden Themen im CloudWatch Amazon-Benutzerhandbuch:

- Den CloudWatch Agenten installieren
- Installation des CloudWatch Agenten auf lokalen Servern

Bevor Sie beginnen

Bevor Sie mit der Migration zum CloudWatch Agenten für die Protokollerfassung beginnen, stellen Sie sicher, dass die Knoten, auf denen Sie die Migration durchführen werden, die folgenden Anforderungen erfüllen:

- Das Betriebssystem ist eine 64-Bit-Version von Windows Server.
- SSM Agent 2.2.93.0 oder höher ist auf dem Knoten installiert.
- SSM Agent ist für die Überwachung auf dem Knoten konfiguriert.

Themen

- Automatische Migration zum Agenten CloudWatch
- Manuelles Migrieren zum Agenten CloudWatch

Automatische Migration zum Agenten CloudWatch

Windows ServerNur für EC2-Instances können Sie die AWS Systems Manager Konsole oder die AWS Command Line Interface (AWS CLI) verwenden, um automatisch zum CloudWatch Agenten als Tool zur Protokollerfassung zu migrieren.

> **Note**
>
> AWS Systems Manager unterstützt die Migration vom Unified CloudWatch Agent SSM Agent zum Sammeln von Protokollen und Messdaten nur in 64-Bit-Versionen von Windows. Informationen zur Einrichtung des Unified CloudWatch Agents auf anderen Betriebssystemen und vollständige Informationen zur Verwendung des CloudWatch Agenten finden Sie unter Sammeln von Metriken und Protokollen von Amazon EC2 EC2-Instances und lokalen Servern mit dem CloudWatch Agenten im _CloudWatch Amazon-Benutzerhandbuch.
> Sie können den CloudWatch Agenten auf anderen unterstützten Betriebssystemen verwenden, aber Sie können Systems Manager nicht verwenden, um eine Tool-Migration durchzuführen.

Überprüfen Sie nach erfolgreicher Migration Ihre Ergebnisse, CloudWatch um sicherzustellen, dass Sie die erwarteten Metriken, Protokolle oder Windows-Ereignisprotokolle erhalten. Wenn Sie mit den Ergebnissen zufrieden sind, können Sie optional folgende Aktion durchführen: Speichern Sie die CloudWatch Agentenkonfigurationseinstellungen in Parameter Store. Wenn die Migration nicht erfolgreich verlaufen ist oder die Ergebnisse nicht den Erwartungen entsprechen, können Sie Rollback zur Protokollerfassung mit SSM Agent ausprobieren.

> **Note**
>
> Wenn Sie eine Quellkonfigurationsdatei migrieren möchten, die einen `{hostname}`-Eintrag enthält, sollten Sie daran denken, dass der `{hostname}`-Eintrag den Wert des Felds ändern kann, falls die Migration abgeschlossen ist. Nehmen wir beispielsweise an, dass der folgende `"LogStream": "{hostname}"` Eintrag einem Server namens MyLogServer001 zugeordnet ist.
>
> ```
> {
> "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}"
     }
}
```

Nach der Migration wird dieser Eintrag einer Domäne wie ip-11-1-1-11.production zugeordnet. ExampleCompany.com. Um den lokalen hostname-Wert beizubehalten, geben Sie `{local_hostname}` anstelle von `{hostname}` an.

Um automatisch zum CloudWatch Agenten (Konsole) zu migrieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command und anschließend Run command (Befehl ausführen) aus.
3. Wählen Sie in der Liste Command document (Befehlsdokument) die Option `AmazonCloudWatch-MigrateCloudWatchAgent` aus.
4. Wählen Sie für Status die Option Enabled.
5. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

6. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

7. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

8. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

9. Wählen Sie Ausführen aus.

Um automatisch zum CloudWatch Agenten zu migrieren ()AWS CLI

- Führen Sie den folgenden Befehl aus.

```
aws ssm send-command --document-name AmazonCloudWatch-MigrateCloudWatchAgent --
targets Key=instanceids,Values=ID1,ID2,ID3
```

ID1, *ID2* und *ID3* stellen die IDs der Knoten dar, die Sie aktualisieren möchten, wie beispielsweise i-02573cafcfEXAMPLE.

Manuelles Migrieren zum Agenten CloudWatch

Gehen Sie für lokale Windows Server Knoten oder EC2-Instances für wie folgt vorWindows Server, um die Protokollerfassung manuell auf den CloudWatch Amazon-Agenten zu migrieren.

> **Note**
>
> Wenn Sie eine Quellkonfigurationsdatei migrieren möchten, die einen `{hostname}`-Eintrag enthält, sollten Sie daran denken, dass der `{hostname}`-Eintrag den Wert des Felds ändern kann, falls die Migration abgeschlossen ist. Nehmen wir zum Beispiel an, dass der folgende *`"LogStream": "{hostname}"`* Eintrag einem Server namens MyLog Server001 zugeordnet ist.
>
> ```
> {
> "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}"
> ```

```
    }
}
```

Nach der Migration wird dieser Eintrag einer Domäne wie ip-11-1-1-11.production zugeordnet. ExampleCompany.com. Um den lokalen hostname-Wert beizubehalten, geben Sie `{local_hostname}` anstelle von `{hostname}` an.

Erstens: Um den CloudWatch Agenten (Konsole) zu installieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command und anschließend Run command (Befehl ausführen) aus.
3. Wählen Sie in der Liste Command document (Befehlsdokument) die Option `AWS-ConfigureAWSPackage` aus.
4. Für Action (Aktion), wählen Sie `Install` aus.
5. Geben Sie unter Name **`AmazonCloudWatchAgent`** ein.
6. Geben Sie unter Version **`latest`** ein, wenn dies nicht bereits standardmäßig bereitgestellt wird.
7. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

8. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

10. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Wählen Sie Ausführen aus.

2) Aktualisieren des JSON-Formats der Konfigurationsdaten

- Um die JSON-Formatierung der vorhandenen Konfigurationseinstellungen für den CloudWatch Agenten zu aktualisieren Run Command, verwenden Sie eine Funktion von AWS Systems Manager oder melden Sie sich direkt mit einer RDP-Verbindung beim Knoten an, um die folgenden PowerShell Windows-Befehle nacheinander auf dem Knoten auszuführen.

```
cd ${Env:ProgramFiles}\\Amazon\\AmazonCloudWatchAgent
```

```
.\\amazon-cloudwatch-agent-config-wizard.exe --isNonInteractiveWindowsMigration
```

{Env:ProgramFiles} steht normalerweise `C:\Program Files` für den Speicherort, an dem sich das Amazon-Verzeichnis befindet, das den CloudWatch Agenten enthält.

Drei: Um den CloudWatch Agenten zu konfigurieren und zu starten (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command und anschließend Run command (Befehl ausführen) aus.
3. Wählen Sie in der Liste Command document (Befehlsdokument) die Option `AWS-RunPowerShellScript` aus.
4. Geben Sie unter Commands (Befehle) die beiden folgenden Befehle ein.

```
cd ${Env:ProgramFiles}\Amazon\AmazonCloudWatchAgent
```

```
.\amazon-cloudwatch-agent-ctl.ps1 -a fetch-config -m ec2 -c file:config.json -s
```

{Env:ProgramFiles} steht normalerweise `C:\Program Files` für den Speicherort, an dem sich das Amazon-Verzeichnis befindet, das den CloudWatch Agenten enthält.

5. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

6. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

7. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

8. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

   Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

9. Wählen Sie Ausführen aus.

4) Deaktivieren der Protokollerfassung im SSM Agent (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command und anschließend Run command (Befehl ausführen) aus.
3. Wählen Sie in der Liste Command document (Befehlsdokument) die Option `AWS-ConfigureCloudWatch` aus.
4. Wählen Sie unter Status die Option Disabled (Deaktiviert) aus.
5. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

6. Wählen Sie unter Status die Option `Disabled` aus.
7. Für Rate control (Ratenregelung):
   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.
8. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der

für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

9. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

   Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

10. Wählen Sie Ausführen aus.

    Nachdem Sie diese Schritte abgeschlossen haben, überprüfen Sie Ihre Logs, CloudWatch um sicherzustellen, dass Sie die erwarteten Metriken, Protokolle oder Windows-Ereignisprotokolle erhalten. Wenn die Ergebnisse zufriedenstellend sind, können Sie optional Speichern Sie die CloudWatch Agentenkonfigurationseinstellungen in Parameter Store. Wenn die Migration nicht erfolgreich verlaufen ist oder die Ergebnisse nicht den Erwartungen entsprechen, können Sie Rollback zur Protokollerfassung mit SSM Agent ausprobieren.

Speichern Sie die CloudWatch Agentenkonfigurationseinstellungen in Parameter Store

Sie können den Inhalt einer CloudWatch Agent-Konfigurationsdatei in speichernParameter Store. Wenn diese Konfigurationsdaten in einem Parameter angegeben werden, können mehrere Knoten ihre Konfigurations-Einstellungen daraus ableiten und Sie müssen keine Konfigurationsdateien auf Ihren Knoten erstellen oder manuell aktualisieren. Sie können beispielsweise den Inhalt des Parameters in Konfigurationsdateien auf mehreren Knoten schreiben oder eine Funktion von verwenden State Manager AWS Systems Manager, um Konfigurationsabweichungen in den CloudWatch Agentenkonfigurationseinstellungen über eine Flotte von Knoten hinweg zu vermeiden. Run Command

Wenn Sie den Assistenten zur CloudWatch Agentenkonfiguration ausführen, können Sie festlegen, dass der Assistent Ihre Konfigurationseinstellungen als neuen Parameter in speichertParameter Store. Informationen zur Ausführung des Assistenten für die CloudWatch Agentenkonfiguration finden

Sie unter Erstellen der CloudWatch Agentenkonfigurationsdatei mit dem Assistenten im CloudWatch Amazon-Benutzerhandbuch.

Wenn Sie den Assistenten ausgeführt, aber nicht die Option zum Speichern der Einstellungen als Parameter ausgewählt haben, oder wenn Sie die CloudWatch Agenten-Konfigurationsdatei manuell erstellt haben, können Sie die Daten, die als Parameter auf Ihrem Knoten gespeichert werden sollen, in der folgenden Datei abrufen.

```
${Env:ProgramFiles}\Amazon\AmazonCloudWatchAgent\config.json
```

{Env:ProgramFiles} steht normalerweise `C:\Program Files` für den Speicherort, an dem sich das Amazon-Verzeichnis befindet, das den CloudWatch Agenten enthält.

Wir empfehlen, ein Backup des JSON-Formats in dieser Datei an einem anderen Speicherort als den Knoten selbst zu speichern.

Weitere Informationen zum Erstellen eines Parameters finden Sie unter Erstellen von Systems Manager-Parametern.

Weitere Informationen über den CloudWatch Agenten finden Sie im CloudWatch Amazon-Benutzerhandbuch unter Erfassung von Metriken und Protokollen von Amazon EC2 EC2-Instances und lokalen Servern mit dem CloudWatch Agenten.

Rollback zur Protokollerfassung mit SSM Agent

Wenn Sie wieder SSM Agent für die Protokollerfassung verwenden möchten, führen Sie die folgenden Schritte aus.

1) Abrufen von Konfigurationsdaten aus SSM Agent

1. Suchen Sie die Inhalte der SSM Agent-Config-Datei auf dem Knoten, auf dem Sie erneut Protokolle mit dem SSM Agent erfassen möchten. Diese JSON-Datei befindet sich in der Regel am folgenden Speicherort:

   ```
   ${Env:ProgramFiles}\\Amazon\\SSM\\Plugins\\awsCloudWatch\
   \AWS.EC2.Windows.CloudWatch.json
   ```

   {Env:ProgramFiles} steht in der Regel für den Speicherort, an dem sich das `Amazon` Verzeichnis befindet. `C:\Program Files`
2. Kopieren Sie diese Daten für die Verwendung in einem späteren Schritt in eine Textdatei.

Wir empfehlen, ein Backup der JSON-Datei an einem anderen Speicherort als den Knoten selbst zu speichern.

Zweitens: Um den CloudWatch Agenten (Konsole) zu deinstallieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command und anschließend Run command (Befehl ausführen) aus.
3. Wählen Sie in der Liste Command document (Befehlsdokument) die Option `AWS-ConfigureAWSPackage` aus.
4. Wählen Sie für Action (Aktion) die Option Uninstall (Deinstallieren) aus.
5. Geben Sie unter Name **`AmazonCloudWatchAgent`** ein.
6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

8. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

9. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

   Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

10. Wählen Sie Ausführen aus.

Drei: Aktivieren der Protokollerfassung in SSM Agent(Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Run Command und anschließend Run command (Befehl ausführen) aus.
3. Wählen Sie in der Liste Command document (Befehlsdokument) die Option `AWS-ConfigureCloudWatch` aus.
4. Wählen Sie unter Status die Option `Enabled` aus.
5. Fügen Sie unter Properties (Eigenschaften) den Inhalt der alten Konfigurationsdaten ein, die Sie in der Textdatei gespeichert haben.
6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

8. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

9. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

   Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

10. Wählen Sie Ausführen aus.

Senden von SSM Agent-Protokollen an CloudWatch Logs

AWS Systems Manager-Agent (SSM Agent) ist eine Amazon-Software, die auf Ihren EC2 Instances, Edge-Geräten, On-Premises-Servern und virtuellen Maschinen (VMs) ausgeführt wird, die für Systems Manager konfiguriert sind. SSM Agent verarbeitet Anfragen vom Systems Manager Service in der Cloud und konfiguriert Ihren Computer wie in der Anfrage angegeben. Mehr über SSM Agent erfahren Sie unter Arbeiten mit SSM Agent.

Darüber hinaus können Sie mit den folgenden Schritten SSM Agent so konfigurieren, dass er Protokolldaten an Amazon CloudWatch Logs sendet.

Bevor Sie beginnen

Erstellen einer Protokollgruppe in CloudWatch Logs. Weitere Informationen finden Sie unter Erste Schritte mit CloudWatch Logs im Benutzerhandbuch zu Amazon CloudWatch Logs.

Konfiguration von SSM Agent zum Senden von Protokollen an CloudWatch.

1. Melden Sie sich bei einem Knoten an und suchen Sie die folgende Datei:

   Linux

   Bei den meisten Linux-Knotentypen: `/etc/amazon/ssm/seelog.xml.template`.

   Auf Ubuntu Server 20.10 STR und 20.04, 18.04 und 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. Ändern des Dateinamens von `seelog.xml.template` in `seelog.xml`

   > **Note**
   >
   > Auf Ubuntu Server 20.10 STR und 20.04, 18.04 und 16.04 LTS muss die Datei `seelog.xml` im Verzeichnis `/etc/amazon/ssm/` erstellt werden. Sie können dieses Verzeichnis und diese Datei mit den folgenden Befehlen erstellen.
   >
   > ```
   > 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. Öffnen Sie die Datei `seelog.xml` in einem Texteditor und suchen Sie nach folgendem Abschnitt.

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. Bearbeiten Sie die Datei und fügen Sie nach dem schließenden </filter>-Tag ein benutzerdefiniertes Namenselement hinzu. Im folgenden Beispiel wird der benutzerdefinierte Name als `cloudwatch_receiver` angegeben.

   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. Speichern Sie Ihre Änderungen und starten Sie dann SSM Agent oder den Knoten neu.
6. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
7. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus und wählen Sie dann den Namen der Protokollgruppe aus.

> **Tip**
>
> Der Protokoll-Stream für SSM Agent-Protokolldatei-Daten wird nach Knoten-ID organisiert.

Überwachung der Ereignisse Ihrer Änderungsanfragen

Nachdem Sie die Integration mit AWS CloudTrail Lake aktiviert und einen Ereignisdatenspeicher erstellt haben, können Sie überprüfbare Details zu den Änderungsanforderungen einsehen, die in Ihrem Konto oder Ihrer Organisation ausgeführt werden. Dazu gehören Details wie die folgenden:

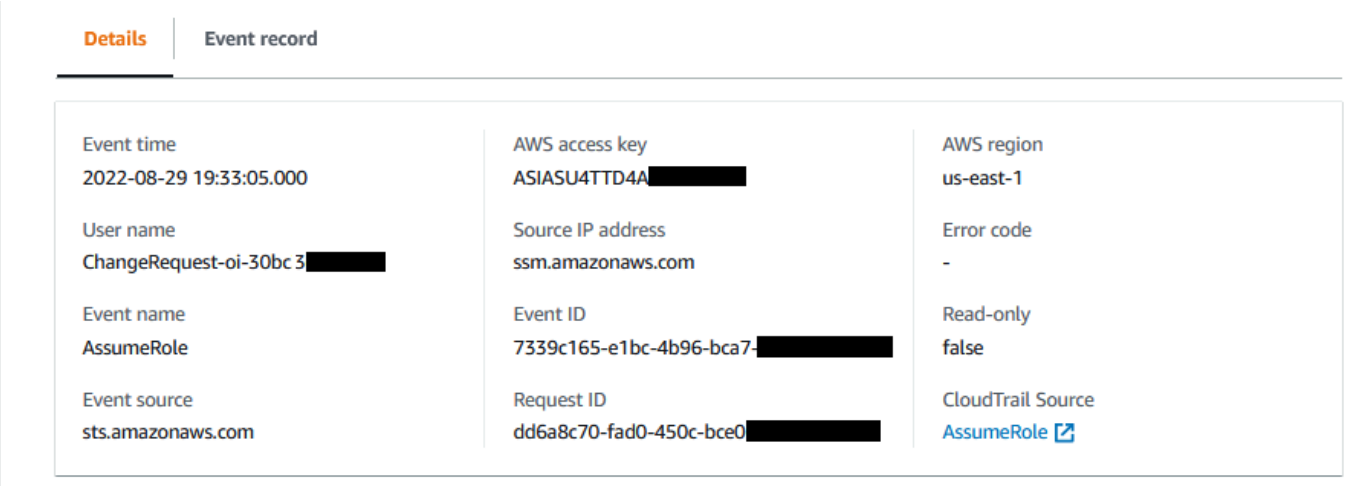
- Die Identität des Benutzers, der die Änderungsanfrage initiiert hat
- Der AWS-Regionen Ort, an dem die Änderungen vorgenommen wurden
- Die Quell-IP-Adresse für die Anfrage
- Der für die Anfrage verwendete AWS Zugriffsschlüssel
- Die Ausführung der API-Aktionen für die Änderungsanfrage
- Die für diese Aktionen enthaltenen Anfrageparameter

- Die während des Vorgangs aktualisierten Ressourcen

Im Folgenden finden Sie Beispiele für Ereignisdetails, die Sie für eine Änderungsanforderung anzeigen können, nachdem Sie den Ereignisdatenspeicher in AWS CloudTrail Lake erstellt haben.

Details

Das folgende Image zeigt die allgemeinen Informationen zu einer Änderungsanfrage, die auf der Registerkarte Details verfügbar sind. Zu diesen Details gehören Informationen wie der Zeitpunkt des Beginns der Änderungsanfrage, die ID des Benutzers, der die Änderungsanfrage initiiert hat, die betroffenen AWS-Region sowie die mit der Anfrage verknüpften Ereignis-ID und Anfrage-ID.



Event record

Die folgende Abbildung zeigt die Struktur des JSON-Inhalts, der von CloudTrail Lake für ein Änderungsanforderungsereignis bereitgestellt wird. Diese Daten werden in einem Änderungsauftrag auf der Registerkarte Event record (Ereignisdatensatz) bereitgestellt.

Important

Wenn Sie Change Manager für eine Organisation verwenden, können Sie das folgende Verfahren ausführen, während Sie entweder beim Verwaltungskonto oder beim delegierten Administratorkonto für Change Manager angemeldet sind.
Um jedoch das delegierte Administratorkonto für diese Schritte zu verwenden, muss dasselbe delegierte Administratorkonto sowohl für als auch CloudTrail angegeben werden. Change Manager
Wenn Sie sich beim Verwaltungskonto für anmeldenChange Manager, können Sie das delegierte Administratorkonto für CloudTrail auf der CloudTrail Seite Einstellungen hinzufügen oder ändern. Dies muss erfolgen, bevor das delegierte Administratorkonto einen Ereignisdatenspeicher zur Verwendung durch die gesamte Organisation erstellen kann.

Um die CloudTrail Lake-Ereignisverfolgung zu aktivieren, gehen Sie von Change Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.

3. Wählen Sie die Registerkarte Requests (Anforderungen).
4. Wählen Sie eine bereits vorhandene Anfrage aus und wählen Sie dann die Registerkarte Associated events (Zugeordnete Ereignisse).
5. Wählen Sie Enable CloudTrail Lake aus.
6. Folgen Sie den Schritten unter Erstellen eines Ereignisdatenspeichers für CloudTrail Ereignisse im AWS CloudTrail Benutzerhandbuch.

   Um sicherzustellen, dass die Ereignisdaten für Ihre Änderungsanfragen gespeichert werden, treffen Sie die folgenden Auswahlen, während Sie das Verfahren abschließen:

   - Behalten Sie für Ereignistyp die AWS Standardereignisse und CloudTrailEreignisse bei.
   - Wenn Sie Change Manager mit einer Organisation verwenden, wählen Sie die Option Für alle Konten in meiner Organisation aktivieren aus.
   - Deaktivieren Sie bei Verwaltungsereignissen das Kontrollkästchen Schreiben nicht.

   Andere Optionen, die Sie beim Erstellen Ihres Ereignisdatenspeichers auswählen, wirken sich nicht auf die Speicherung von Ereignisdaten für Ihre Änderungsanfragen aus.

Überwachung Ihrer Automatisierungen

Metriken sind das grundlegende Konzept in Amazon CloudWatch. Eine Metrik stellt eine zeitlich angeordnete Gruppe von Datenpunkten, die an CloudWatch veröffentlicht werden, dar. Eine Metrik können Sie sich als eine zu überwachende Variable und die Datenpunkte als die Werte dieser Variablen im Laufe der Zeit vorstellen.

Automation ist eine Funktion von AWS Systems Manager. Systems Manager veröffentlicht Metriken zur Automation-Nutzung in CloudWatch. So können Sie Alarme basierend auf diesen Metriken festlegen.

So zeigen Sie Metriken in der CloudWatch-Konsole an

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie SSM aus.
4. Wählen Sie auf der Registerkarte Metrics (Metriken) erst Usage (Nutzung) und dann By AWS Resource (Nach Ressource) aus.

5. Geben Sie in das Suchfeld neben der Liste der Metriken SSM ein.

So zeigen Sie Automation-Metriken mit der AWS CLI an

Öffnen Sie eine Eingabeaufforderung und verwenden Sie den folgenden Befehl.

```
aws cloudwatch list-metrics \
    --namespace "AWS/Usage"
```

Automation-Metriken

Systems Manager sendet die folgenden Automation-Metriken an CloudWatch.

Metrik	Beschreibung
`ConcurrentAutomationUsage`	Die Anzahl der Automatisierungen, die gleichzeitig in AWS-Konto und AWS-Region ausgeführt werden können.
`QueuedAutomationUsage`	Die Anzahl der derzeit in der Warteschlange befindlichen Automatisierungen, die noch nicht gestartet wurden und den Status `Pending` aufweisen.

Weitere Informationen zum Arbeiten mit CloudWatch-Metriken finden Sie in den folgenden Themen im Amazon CloudWatch-Benutzerhandbuch:

- Metriken
- Verwenden von Amazon-CloudWatch-Metriken
- Verwenden von Amazon CloudWatch-Alarmen

Überwachen von Run Command-Metriken mit Amazon CloudWatch

Metriken sind das grundlegende Konzept in Amazon CloudWatch. Eine Metrik stellt eine zeitlich angeordnete Gruppe von Datenpunkten, die an CloudWatch veröffentlicht werden, dar. Sie können sich eine Metrik als eine zu überwachende Variable und die Datenpunkte als die Werte dieser Variablen im Laufe der Zeit vorstellen.

AWS Systems Manager veröffentlicht Metriken über den Status von Run Command-Befehlen an CloudWatch, wodurch Sie Alarme auf Basis dieser Metriken einstellen können. Run Command ist eine Funktion von AWS Systems Manager. Diese Statistiken werden für einen längeren Zeitraum aufgezeichnet, damit Sie auf historische Informationen zugreifen können und eine bessere Übersicht über die Erfolgsrate der in Ihrem AWS-Kontoausgeführten Befehle erhalten.

Zu den Terminalstatuswerten für Befehle, für die Sie Metriken verfolgen können, gehören `Success`, `Failed` und `Delivery Timed Out`. Für ein SSM-Befehlsdokument, das stündlich ausgeführt werden soll, können Sie beispielsweise einen Alarm konfigurieren, der Sie benachrichtigt, wenn der Status `Success` für eine dieser Stunden nicht gemeldet wird. Weitere Informationen zu Befehlsstatuswerten finden Sie unter Grundlegendes zu Befehlsstatus.

Anzeigen von Metriken in der CloudWatch-Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie im Bereich Alarms by AWS service (Alarme nach Service) für Services (Services) SSM-Run Command aus.

So zeigen Sie Metriken mit der a AWS CLI

Öffnen Sie eine Eingabeaufforderung und verwenden Sie den folgenden Befehl.

```
aws cloudwatch list-metrics --namespace "AWS/SSM-RunCommand"
```

Verwenden Sie den folgenden Befehl, um alle verfügbaren Metriken aufzulisten.

```
aws cloudwatch list-metrics
```

Systems Manager Run Command-Metriken und -Dimensionen

Systems Manager sendet einmal pro Minute Run Command-Befehlsmetriken an CloudWatch.

Systems Manager sendet die folgenden Befehlsmetriken an CloudWatch.

> **Note**
>
> Diese Metriken verwenden `Count` als Einheit, daher sind `Sum` und `SampleCount` die nützlichsten Statistiken.

Metrik	Beschreibung
`CommandsDeliveryTimedOut`	Die Anzahl der Befehle, die den Terminalstatus `Delivery Timed Out` haben.
`CommandsFailed`	Die Anzahl der Befehle, die den Terminalstatus `Failed` haben.
`CommandsSucceeded`	Die Anzahl der Befehle, die den Terminalstatus `Success` haben.

Weitere Informationen zum Arbeiten mit CloudWatch-Metriken finden Sie in den folgenden Themen im Amazon CloudWatch-Benutzerhandbuch:

- Metriken
- Verwenden von Amazon-CloudWatch-Metriken
- Verwenden von Amazon CloudWatch-Alarmen

AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail

AWS Systems Manager ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem ausgeführten Aktionen bereitstellt AWS-Service. CloudTrail erfasst API-Aufrufe für Systems Manager als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Systems Manager Manager-Konsole und Codeaufrufen an die Systems Manager Manager-API-Operationen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Systems Manager gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln.

Jeder Ereignis- oder Protokolleintrag enthält Informationen, anhand derer Sie feststellen können, wer die Anfrage gestellt hat.

- Root-Benutzer des AWS-Kontos
- Temporäre Sicherheitsanmeldedaten von einer AWS Identity and Access Management (IAM-) Rolle oder einem Verbundbenutzer.
- Langfristige Sicherheits-Anmeldeinformation eines IAM-Benutzers.
- Anfragen, die im Namen eines IAM Identity Center-Benutzers gestellt wurden.
- Noch ein AWS-Service.

Weitere Informationen finden Sie unter dem CloudTrailUserIdentity-Element.

CloudTrail ist in Ihrem aktiv AWS-Konto , wenn Sie das Konto erstellen, und Sie haben automatisch Zugriff auf den CloudTrail Eventverlauf. Der CloudTrail Ereignisverlauf bietet eine einsehbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem. AWS-Region Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter Arbeiten mit dem CloudTrail Ereignisverlauf. Für die Anzeige des Eventverlaufs CloudTrail fallen keine Gebühren an.

Für eine fortlaufende Aufzeichnung der Ereignisse in AWS-Konto den letzten 90 Tagen erstellen Sie einen Trail- oder CloudTrailLake-Event-Datenspeicher.

CloudTrail Pfade

Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Alle mit dem erstellten Pfade AWS Management Console sind regionsübergreifend. Sie können einen Pfad mit einer oder mehreren Regionen erstellen, indem Sie den verwenden. AWS CLI Es wird empfohlen, einen Trail mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten in Ihrem Konto AWS-Regionen erfassen. Wenn du einen Trail mit nur einer Region erstellst, kannst du dir nur die Ereignisse ansehen, die in den Trails protokolliert wurden. AWS-Region Weitere Informationen zu Trails finden Sie unter Einen Trail für Sie erstellen AWS-Konto und Einen Trail für eine Organisation erstellen im AWS CloudTrail Benutzerhandbuch.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos an Ihren Amazon S3 S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3 S3-Speichergebühren an. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preise. Informationen zu Amazon-S3-Preisen finden Sie unter Amazon S3-Preise.

CloudTrail Datenspeicher für Ereignisse in Lake

CloudTrail Mit Lake können Sie SQL-basierte Abfragen für Ihre Ereignisse ausführen. CloudTrail Lake konvertiert bestehende Ereignisse im zeilenbasierten JSON-Format in das Apache ORC-Format. ORC ist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst, bei denen es sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von erweiterten Ereignisselektoren auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, steuern, welche Ereignisse bestehen bleiben und für Sie zur Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie unter Arbeiten mit AWS CloudTrail Lake im AWS CloudTrail Benutzerhandbuch.

CloudTrail Für das Speichern und Abfragen von Ereignisdaten in Lake fallen Kosten an. Beim Erstellen eines Ereignisdatenspeichers wählen Sie die Preisoption aus, die für den Ereignisdatenspeicher genutzt werden soll. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungsdauer für den Ereignisdatenspeicher. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preise.

Systems Manager Manager-Datenereignisse in CloudTrail

Datenereignisse liefern Informationen über die Ressourcenoperationen, die auf oder in einer Ressource ausgeführt werden (z. B. das Erstellen oder Öffnen eines Steuerkanals). Sie werden auch als Vorgänge auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Protokolliert standardmäßig CloudTrail keine Datenereignisse. Der CloudTrail Ereignisverlauf zeichnet keine Datenereignisse auf.

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preisgestaltung.

Sie können Datenereignisse für die Systems Manager Manager-Ressourcentypen mithilfe der CloudTrail Konsole oder CloudTrail API-Operationen protokollieren. AWS CLI Weitere Informationen zum Protokollieren von Datenereignissen finden Sie unter Protokollieren von Datenereignissen mit der AWS Management Console und Protokollieren von Datenereignissen mit dem AWS Command Line Interface im AWS CloudTrail Benutzerhandbuch.

In der folgenden Tabelle sind die Systems Manager Manager-Ressourcentypen aufgeführt, für die Sie Datenereignisse protokollieren können. In der Spalte Datenereignistyp (Konsole) wird der Wert angezeigt, den Sie in der Liste Datenereignistyp auf der CloudTrail Konsole auswählen können.

In der Wertspalte resources.type wird der `resources.type` Wert angezeigt, den Sie angeben würden, wenn Sie erweiterte Event-Selektoren mithilfe der AWS CLI APIs oder konfigurieren würden. CloudTrail In der CloudTrail Spalte „Protokollierte Daten-APIs“ werden die API-Aufrufe angezeigt, die CloudTrail für den Ressourcentyp protokolliert wurden.

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten-APIs, bei denen die Anmeldung erfolgt CloudTrail
Systems Manager	`AWS::SSMMessages::ControlChannel`	• `CreateControlChannel` • `OpenControlChannel` Weitere Informationen zu diesen Vorgängen finden Sie unter Von Amazon Message Gateway Service definierte Aktionen in der Service Authorization Reference.
Von Systems Manager verwalteter Knoten	`AWS::SSM::ManagedNode`	• `RequestManagedInstanceRoleToken` — Dieses Ereignis wird generiert, wenn der Systems Manager Agent (SSM Agent), der auf einem von Systems Manager verwalteten Knoten ausgeführt wird, Anmeldeinformationen vom Systems Manager-Anmeldeinformationsdienst anfordert.

Sie können erweiterte Ereignisauswahlen so konfigurieren, dass sie nach den `resources.ARN` Feldern, und filtern `eventNamereadOnly`, sodass nur die Ereignisse protokolliert werden, die für Sie wichtig sind. Weitere Informationen zu diesen Feldern finden Sie AdvancedFieldSelectorin der AWS CloudTrail API-Referenz.

Systems Manager Manager-Verwaltungsereignisse in CloudTrail

Verwaltungsereignisse enthalten Informationen zu Verwaltungsvorgängen, die an Ressourcen in Ihrem ausgeführt werden AWS-Konto. Sie werden auch als Vorgänge auf Steuerebene bezeichnet. CloudTrail Protokolliert standardmäßig Verwaltungsereignisse.

Systems Manager protokolliert alle Operationen auf der Steuerungsebene CloudTrail als Verwaltungsereignisse. Die API-Operationen von Systems Manager sind in der AWS Systems Manager API-Referenz dokumentiert. Beispielsweise generieren Aufrufe der `StartSession` Aktionen `CreateMaintenanceWindowsPutInventory`,`SendCommand`, und Einträge in den CloudTrail Protokolldateien. Ein Beispiel für die Einrichtung CloudTrail zur Überwachung eines Systems Manager Manager-API-Aufrufs finden Sie unterÜberwachung der Sitzungsaktivität mit Amazon EventBridge (Konsole).

Beispiele Systems Manager Manager-Ereignisse

Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten API-Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass Ereignisse nicht in einer bestimmten Reihenfolge angezeigt werden.

Beispiele:

- Beispiele für Verwaltungsereignisse
- Beispiele für Datenereignisse

Beispiele für Verwaltungsereignisse

Beispiel 1: **`DeleteDocument`**

Das folgende Beispiel zeigt ein CloudTrail Ereignis, das den `DeleteDocument` Vorgang mit einem Dokument demonstriert, das `example-Document` in der Region USA Ost (Ohio) (us-east-2) benannt ist.

```
{
    "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"
}
```


Beispiel 2: **`StartConnection`**

Das folgende Beispiel zeigt ein CloudTrail Ereignis für einen Benutzer, der eine RDP-Verbindung über Fleet Manager die Region USA Ost (Ohio) (us-east-2) startet. Die zugrunde liegende API-Aktion ist `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"
}
```

Beispiele für Datenereignisse

Beispiel 1: **`CreateControlChannel`**

Das folgende Beispiel zeigt ein CloudTrail Ereignis, das den `CreateControlChannel` Vorgang demonstriert.

```
{
  "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"
}
```


Beispiel 2: **`RequestManagedInstanceRoleToken`**

Das folgende Beispiel zeigt ein CloudTrail Ereignis, das den `RequestManagedInstanceRoleToken` Vorgang demonstriert.


```
{
    "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"
```

```
}
```

Informationen zu CloudTrail Datensatzinhalten finden Sie im AWS CloudTrail Benutzerhandbuch unter CloudTrailDatensatzinhalt.

Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs

Automation, eine Funktion von AWS Systems Manager, ist in Amazon CloudWatch Logs integriert. Sie können die Ausgabe von `aws:executeScript`-Aktionen in Ihren Runbooks an die von Ihnen angegebene Protokollgruppe senden. Systems Manager erstellt keine Protokollgruppe oder Protokoll-Streams für Dokumente, die `aws:executeScript`-Aktionen nicht verwenden. Wenn das Dokument `aws:executeScript` verwendet, bezieht sich die an CloudWatch Logs gesendete Ausgabe nur auf diese Aktionen. Sie können die `aws:executeScript`-Aktionsausgabe, die in der Protokollgruppe „CloudWatch Logs“ für Debug- und Fehlerbehebungszwecke gespeichert ist, verwenden. Wenn Sie eine Protokollgruppe auswählen, die verschlüsselt ist, wird die `aws:executeScript`-Aktionsausgabe ebenfalls verschlüsselt. Protokollierungsausgabe von `aws:executeScript`-Aktionen ist eine Einstellung auf Kontoebene.

Um Aktionsausgaben an CloudWatch Logs für Amazon-eigene Runbooks zu senden, muss der Benutzer oder die Rolle, der bzw. die die Automatisierung ausführt, über Berechtigungen für die folgenden Vorgänge verfügen:

- `logs:CreateLogGroup`
- `logs:CreateLogStream`
- `logs:DescribeLogGroups`
- `logs:DescribeLogStreams`
- `logs:PutLogEvents`

Für Runbooks, die Sie besitzen, müssen der IAM-Servicerolle (oder AssumeRole), die Sie zum Ausführen des Runbooks verwenden, dieselben Berechtigungen hinzugefügt werden.

Senden einer Aktionsausgabe an CloudWatch Logs (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Registerkarte Preferences (Präferenzen) und anschließend Edit (Bearbeiten) aus.
4. Aktivieren Sie das Kontrollkästchen neben Send output to CloudWatch Logs (Senden von Ausgaben an CloudWatch Logs).
5. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Encrypt log data (Verschlüsseln von Protokolldaten). Wenn diese Funktion aktiviert ist, werden die Protokolldaten mithilfe des serverseitigen Verschlüsselungsschlüssels, der für die Protokollgruppe angegeben wurde, verschlüsselt. Wenn Sie die an CloudWatch Logs gesendeten Protokolldaten nicht verschlüsseln möchten, deaktivieren Sie das Kontrollkästchen. Deaktivieren Sie das Kontrollkästchen, wenn die Verschlüsselung für die Protokollgruppe nicht zulässig ist.
6. Wählen Sie für CloudWatch Logs log group (CloudWatch Logs-Protokollgruppe) einen der folgenden Schritte aus, um die vorhandene CloudWatch Logs-Protokollgruppe in Ihrem AWS-Konto anzugeben, an die Sie Aktionsausgaben senden wollen:

   - Send output to the default log group (Ausgabe an die Standardprotokollgruppe senden) - Wenn die Standard-Protokollgruppe nicht vorhanden ist (`/aws/ssm/automation/executeScript`), erstellt Automation diese für Sie.
   - Choose from a list of log groups (Aus einer Liste von Protokollgruppen auswählen): Wählen Sie eine Protokollgruppe, die bereits in Ihrem Konto erstellt wurde, um die Aktionsausgabe zu speichern.
   - Enter a log group name (Eingabe eines Protokollgruppennamens): Geben Sie in das Textfeld den Namen einer Protokollgruppe ein, die bereits in Ihrem Konto angelegt wurde, um die Aktionsausgabe zu speichern.
7. Wählen Sie Save (Speichern).

Senden einer Aktionsausgabe an CloudWatch Logs (Befehlszeile)

1. Öffnen Sie das bevorzugte Befehlszeilen-Tool und führen Sie den folgenden Befehl aus, um das Aktionsausgabeziel zu aktualisieren.

   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"
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die Protokollgruppe anzugeben, an die die Aktionsausgabe gesendet werden soll.

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"
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

3. Führen Sie den folgenden Befehl aus, um den aktuellen Durchsatz für Serviceeinstellungen für Einstellungen für die Aktionsprotokollierung für Automation im aktuellen AWS-Konto und in der AWS-Region anzuzeigen.

   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"
   ```

   Der Befehl gibt Informationen wie die folgenden zurück.

   ```
   {
       "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"
       }
   ```

```
}
```

Konfiguration von Amazon CloudWatch Logs für Run Command

Wenn Sie einen Befehl mit Run Command einer Funktion von senden AWS Systems Manager, können Sie angeben, wohin die Befehlsausgabe gesendet werden soll. Standardmäßig gibt Systems Manager nur die ersten 24.000 Zeichen der Befehlsausgabe zurück. Wenn Sie alle Details der Befehlsausgabe anzeigen möchten, können Sie einen Amazon Simple Storage Service (Amazon S3)-Bucket angeben. Oder Sie können Amazon CloudWatch Logs angeben. Wenn Sie CloudWatch Logs angeben, Run Command werden in regelmäßigen Abständen alle Befehlsausgaben und CloudWatch Fehlerprotokolle an Logs gesendet. Sie können Ausgabeprotokolle nahezu in Echtzeit überwachen, nach bestimmten Ausdrücken, Werten oder Mustern suchen und Alarme basierend auf der Suche erstellen.

Wenn Sie Ihren verwalteten Knoten für die Verwendung der AWS Identity and Access Management (IAM) verwalteten Richtlinien `AmazonSSMManagedInstanceCore` und konfiguriert haben`CloudWatchAgentServerPolicy`, benötigt Ihr Knoten keine zusätzliche Konfiguration, um die Ausgabe an CloudWatch Logs zu senden. Wählen Sie diese Option, wenn Sie Befehle von der Konsole aus senden, oder fügen Sie den `cloud-watch-output-config` Abschnitt und den `CloudWatchOutputEnabled` Parameter hinzu, wenn Sie die AWS Command Line Interface (AWS CLI) AWS Tools for Windows PowerShell, oder eine API-Operation verwenden. Der `cloud-watch-output-config`-Abschnitt und der `CloudWatchOutputEnabled`-Parameter sind später in diesem Thema noch ausführlicher beschrieben.

Informationen zum Hinzufügen von Richtlinien zu einem Instanzprofil für EC2-Instances finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen. Informationen zum Hinzufügen von Richtlinien zu einer Servicerolle für lokale Server und virtuelle Maschinen, die Sie als verwaltete Knoten verwenden möchten, finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Dienstrolle in Hybrid- und Multicloud-Umgebungen.

Wenn Sie auf Ihren Knoten eine benutzerdefinierte Richtlinie verwenden, aktualisieren Sie die Richtlinie auf jedem Knoten, damit Systems Manager Ausgaben und CloudWatch Protokolle an Logs senden kann. Fügen Sie Ihrer benutzerdefinierten Richtlinie die folgenden Richtlinienobjekte hinzu. Weitere Informationen zum Aktualisieren einer IAM-Richtlinie finden Sie unter Editing IAM policies (Bearbeiten von IAM-Richtlinien) im IAM-Benutzerhandbuch.

```
{
```

```
    "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/*"
},
```

CloudWatch Logs angeben, wenn Sie Befehle senden

Um CloudWatch Logs als Ausgabe anzugeben, wenn Sie einen Befehl aus dem senden AWS Management Console, wählen Sie im Abschnitt CloudWatch Ausgabeoptionen die Option Ausgabe aus. Optional können Sie den Namen der CloudWatch Protokollgruppe angeben, an die Sie die Befehlsausgabe senden möchten. Wenn Sie keinen Gruppennamen angeben, erstellt Systems Manager automatisch eine Protokollgruppe für Sie. Die Protokollgruppe verwendet das folgende Bezeichnungsformat: `/aws/ssm/`*`SystemsManagerDocumentName`*

Wenn Sie Befehle mithilfe von ausführen AWS CLI, geben Sie den `cloud-watch-output-config` Abschnitt in Ihrem Befehl an. Dieser Abschnitt ermöglicht Ihnen, den `CloudWatchOutputEnabled`-Parameter und optional den `CloudWatchLogGroupName`-Parameter anzugeben. Ein Beispiel.

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}'
```

Befehlsausgabe in CloudWatch Logs anzeigen

Sobald der Befehl ausgeführt wird, sendet Systems Manager die Ausgabe nahezu in Echtzeit an CloudWatch Logs. Die Ausgabe in CloudWatch Logs verwendet das folgende Format:

CommandID/*InstanceID*/*PluginID*/stdout

CommandID/*InstanceID*/*PluginID*/stderr

Die Ausgabe der Ausführung wird alle 30 Sekunden hochgeladen, oder wenn der Puffer mehr als 200 KB umfasst (je nachdem, was eher eintritt).

> **Note**
>
> Log Streams werden nur erstellt, wenn Ausgabedaten verfügbar sind. Wenn es beispielsweise keine Fehlerdaten für eine Ausführung gibt, wird der stderr-Stream nicht erstellt.

Hier ist ein Beispiel für die Befehlsausgabe, wie sie in CloudWatch Logs angezeigt wird.

```
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
```

Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge

Amazon EventBridge ist ein Serverless-Ereignisbus-Service, mit dem Sie Ihre Anwendungen mit Daten aus verschiedenen Quellen verbinden können. EventBridge stellt einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, Software-as-a-Service-(SaaS)-Anwendungen und

AWS-Services und leitet diese Daten dann an Ziele wie AWS Lambda weiter. Sie können Routing-Regeln einrichten, um festzulegen, wohin Ihre Daten gesendet werden. Auf diese Weise können Sie Anwendungsarchitekturen erstellen, die in Echtzeit auf alle Ihre Datenquellen reagieren. EventBridge ermöglicht es Ihnen, ereignisgesteuerte Architekturen zu erstellen, die lose gekoppelt und verteilt sind.

EventBridge wurde früher als Amazon CloudWatch Events bezeichnet. EventBridge enthält neue Funktionen, mit denen Sie Ereignisse aus SaaS-Partnern und Ihren eigenen Anwendungen empfangen können. Vorhandene Benutzer von CloudWatch Events können auf ihren vorhandenen Standardbus, ihre Regeln und ihre Ereignisse in der neuen EventBridge-Konsole und in der CloudWatch Events-Konsole zugreifen. EventBridge verwendet dieselbe CloudWatch Ereignis-API, sodass die gesamte vorhandene CloudWatch Events-API-Nutzung unverändert bleibt.

EventBridge kann Ereignisse aus Dutzenden von AWS-Services zu Ihren Regeln und Zielen von über 20 AWS-Services hinzufügen.

EventBridge bietet Unterstützung für AWS Systems Manager-Ereignisse und Systems Manager-Ziele.

Unterstützte Systems Manager Ereignistypen

Unter den vielen Typen von Systems Manager Ereignissen, die EventBridge erkennen kann, sind:

- Ein Wartungsfenster, das ausgeschaltet wird.
- Ein erfolgreiches Abschließen eines Automation-Workflows Automation ist eine Funktion von AWS Systems Manager.
- Ein verwalteter Knoten, der außerhalb der Patch-Compliance liegt.
- Ein Parameterwert, der aktualisiert wird.

EventBridge unterstützt Ereignisse aus den folgenden AWS Systems Manager-Funktionen:

- Automatisierung (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Change Calendar (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- -Compliance
- Inventory (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Maintenance Windows (Ereignisse werden auf bestmögliche Weise ausgegeben.)

- Parameter Store (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Run Command (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- State Manager (Ereignisse werden auf bestmögliche Weise ausgegeben.)

Ausführliche Details zu unterstützten Systems Manager Ereignistypen finden Sie unter Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager und EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager.

Unterstützte Systems Manager Zieltypen

EventBridge unterstützt die folgenden drei Systems Manager-Funktionen als Ziele einer Ereignisregel:

- Ausführen eines Automation-Workflows
- Ausführen eines Run Command-Befehlsdokuments Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Erstellen eines OpsCenter OpsItem

Die vorgeschlagenen Möglichkeiten, wie Sie diese Ziele verwenden können, finden Sie unter Beispielszenarien: Systems-Manager-Ziele in Amazon-EventBridge-Regeln.

Weitere Informationen zu den ersten Schritten mit EventBridge und Einrichtungsregeln finden Sie unter Erste Schritte mit Amazon EventBridge im Benutzerhandbuch zu Amazon EventBridge. Weitere Informationen zum Arbeiten mit EventBridge finden Sie unter Amazon EventBridge-Benutzerhandbuch .

Themen

- Konfigurieren von EventBridge für Systems Manager-Ereignisse
- EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager
- Beispielszenarien: Systems-Manager-Ziele in Amazon-EventBridge-Regeln

Konfigurieren von EventBridge für Systems Manager-Ereignisse

Sie können Amazon EventBridge verwenden, um ein Zielereignis durchzuführen, wenn unterstützte AWS Systems Manager-Statusänderungen, Zustandsänderungen oder andere Bedingungen

auftreten. Sie können eine Regel erstellen, die ausgeführt wird, sobald ein Statusübergang oder ein Übergang zu einem oder mehreren Status stattfindet, die für sie von Interesse sind.

Das folgende Verfahren enthält allgemeine Schritte zum Erstellen einer EventBridge-Regel, die aktiviert wird, wenn ein bestimmtes Ereignis von Systems Manager ausgelöst wird. Eine Liste der Verfahren in diesem Benutzerhandbuch, die bestimmte Szenarien behandelt, finden Sie unter Weiter Informationen am Ende dieses Themas.

> **Note**
>
> Wenn ein Service in Ihrem AWS-Konto ein Ereignis ausgibt, wird es stets an den Standard-Event-Bus Ihres Kontos weitergeleitet. Um eine Regel zu schreiben, die bei Ereignissen aus AWS-Services in Ihrem Konto reagiert, ordnen Sie die Regel dem Standard-Event-Bus zu. Sie können eine Regel für einen benutzerdefinierten Event Bus erstellen, der nach Ereignissen aus AWS-Services sucht. Diese Regel wird jedoch nur ausgelöst, wenn Sie ein solches Ereignis über die kontoübergreifende Ereignisbereitstellung aus einem anderen Konto erhalten. Weitere Informationen finden Sie unter Senden und Empfangen von Amazon-EventBridge-Ereignissen zwischen AWS-Konten im Benutzerhandbuch zu Amazon EventBridge.

Konfigurieren von EventBridge für Systems Manager-Ereignisse

1. Öffnen Sie die Amazon EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.
3. Wählen Sie Create rule (Regel erstellen).
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben AWS-Region und auf demselben Event Bus haben.
5. Wählen Sie als Event bus (Event Bus) den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf übereinstimmende Ereignisse reagiert, die von Ihrem eigenen AWS-Konto stammen, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es stets an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus.

7. Wählen Sie Next (Weiter).
8. Wählen Sie für Event source (Ereignisquelle) AWS events or EventBridge partner events (-Ereignisse oder EventBridge-Partnerereignisse).
9. Wählen Sie im Abschnitt Ereignismuster die Option Ereignismusterformular aus.
10. Als Event source (Ereignisquelle) wählen Sie AWS-Services aus.
11. Wählen Sie für AWS service (-Service), die Option Systems Manager aus.
12. Führen Sie für Type (Typ) eine der folgenden Aktionen aus:

    - Wählen Sie Add Events (Ereignisse hinzufügen) aus.

      Wenn Sie All Events (Alle Ereignisse) auswählen, stimmen alle von diesem Systems Manager-Service ausgegebenen Ereignisse mit der Regel überein. Beachten Sie, dass diese Option zu vielen Ereigniszielaktionen führen kann.
    - Wählen Sie den Typ des Systems Manager-Ereignisses aus, der für diese Regel verwendet werden soll. EventBridge unterstützt Ereignisse aus den folgenden AWS Systems Manager-Funktionen:
      - -Automatisierung
      - Change Calendar
      - -Compliance
      - -Bestand
      - Maintenance Windows
      - Parameter Store
      - Run Command
      - State Manager

      > **Note**
      >
      > Für Systems Manager-Aktionen, die von EventBridge nicht unterstützt werden, können Sie einen AWS-API-Aufruf über CloudTrail auswählen, um eine Ereignisregel zu erstellen, die auf einem API-Aufruf basiert, der von CloudTrail aufgezeichnet wird. Ein Beispiel finden Sie unter Überwachung der Sitzungsaktivität mit Amazon EventBridge (Konsole).

13. (Optional) Fügen Sie Filterwerte hinzu, um die Regel spezifischer zu gestalten. Wenn Sie beispielsweise State Manager ausgewählt haben und die Regel auf den Zustand einer einzelnen

verwalteten Instanz beschränken möchten, die von einer Zuordnung anvisiert wird, wählen Sie für Specific type(s) (Spezifische(r) Typ(en)) EC2 State Manager Instance Association State Change aus.

Ausführliche Informationen zu unterstützten Detailtypen finden Sie unter Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager.

Einige Detailtypen verfügen über andere unterstützte Optionen wie den Status. Die verfügbaren Optionen hängen von der ausgewählten Funktion ab.

14. Wählen Sie Next (Weiter).
15. Bei Target types (Zieltypen) wählen Sie AWS-Service aus.
16. Wählen Sie für Ziel auswählen ein Ziel aus, z. B. ein Amazon-SNS-Thema oder eine AWS Lambda-Funktion. Das Ziel wird ausgelöst, wenn ein Ereignis empfangen wird, das dem in der Regel definierten Ereignismuster entspricht.
17. Für viele Zieltypen benötigt EventBridge Berechtigungen zum Senden von Ereignissen an das Ziel. In diesen Fällen kann EventBridge die AWS Identity and Access Management-(IAM)-Rolle erstellen, die zum Ausführen Ihrer Regel erforderlich ist:
    - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Create a new role for this specific resource (Eine neue Rolle für diese spezifische Ressource erstellen).
    - Wenn Sie eine zuvor erstellte IAM-Rolle verwenden möchten, wählen Sie Use existing role (Vorhandene Rolle verwenden)
18. (Optional) Wählen Sie Add another target (Weiteres Ziel hinzufügen) aus, um ein weiteres Ziel für diese Regel hinzuzufügen.
19. Wählen Sie Next (Weiter).
20. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Amazon-EventBridge-Tags im Benutzerhandbuch zu Amazon EventBridge.
21. Wählen Sie Next (Weiter).
22. Überprüfen Sie die Details der Regel und wählen Sie dann Create rule (Regel erstellen) aus.

Weitere Informationen

- Erstellen eines EventBridge Ereignisses, das ein Runbook verwendet (Konsole)
- Übergabe von Daten an Automation mithilfe von Eingangstransformatoren
- Beheben von Compliance-Problemen mithilfe von EventBridge

- Anzeigen von Löschaktionen für einen Bestand in EventBridge
- Konfigurieren von EventBridge-Regeln zum Erstellen von OpsItems
- Konfigurieren von EventBridge Regeln für Parameter und Parameterrichtlinien

EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager

Im Folgenden finden Sie Beispiele im JSON-Format für unterstützte EventBridge Ereignisse für AWS Systems Manager.

Systems Manager Ereignistypen

- AWS Systems Manager Automatisierungsereignisse
- AWS Systems Manager EreignisseChange Calendar
- AWS Systems Manager EreignisseChange Manager
- AWS Systems Manager Ereignisse zur Einhaltung von Vorschriften
- AWS Systems Manager EreignisseMaintenance Windows
- AWS Systems Manager EreignisseParameter Store
- AWS Systems Manager EreignisseOpsCenter
- AWS Systems Manager EreignisseRun Command
- AWS Systems Manager EreignisseState Manager

AWS Systems Manager Automatisierungsereignisse

Benachrichtigung über die Änderung des Automatisierungsschrittstatus

```
{
  "version": "0",
  "id": "eeca120b-a321-433e-9635-dab369006a6b",
  "detail-type": "EC2 Automation Step Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-11-29T19:43:35Z",
  "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, 2016 7:43:25 PM",
    "StartTime": "Nov 29, 2016 7:43:23 PM",
    "Time": 2630.0,
    "StepName": "runFixedCmds",
    "Action": "aws:runCommand"
  }
}
```

Benachrichtigung über die Änderung des Ausführungsstatus

```
{
  "version": "0",
  "id": "d290ece9-1088-4383-9df6-cd5b4ac42b99",
  "detail-type": "EC2 Automation Execution Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-11-29T19:43:35Z",
  "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, 2016 7:43:20 PM",
    "EndTime": "Nov 29, 2016 7:43:26 PM",
    "Time": 5753.0,
    "ExecutedBy": "arn:aws:iam::123456789012:user/userName"
  }
}
```

AWS Systems Manager EreignisseChange Calendar

Im Folgenden finden Sie Beispiele für Ereignisse für AWS Systems ManagerChange Calendar.

> **Note**
>
> Statusänderungen für Kalender, die von anderen gemeinsam genutzt wurden, AWS-Konten werden derzeit nicht unterstützt.

Kalender OFFEN

```
{
    "version": "0",
    "id": "47a3f03a-f30d-1011-ac9a-du3bdEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2020-09-19T18:00:07Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:123456789012:document/MyCalendar"
    ],
    "detail": {
        "state": "OPEN",
        "atTime": "2020-09-19T18:00:07Z",
        "nextTransitionTime": "2020-10-11T18:00:07Z"
    }
}
```

Kalender GESCHLOSSEN

```
{
    "version": "0",
    "id": "f30df03a-1011-ac9a-47a3-f761eEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2020-09-17T21:40:02Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:123456789012:document/MyCalendar"
    ],
    "detail": {
        "state": "CLOSED",
        "atTime": "2020-08-17T21:40:00Z",
```

```
        "nextTransitionTime": "2020-09-19T18:00:07Z"
    }
}
```

AWS Systems Manager EreignisseChange Manager

Benachrichtigung über Statusaktualisierung der Änderungsanfrage – Beispiel 1

```
{
  "version": "0",
  "id": "feab80c1-a8ff-c721-b8b1-96ce70939696",
  "detail-type": "Change Request Status Update",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2023-10-24T10:51:52Z",
  "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": "2023-10-24T10:50:33.180334Z",
    "ops-item-modified-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-modified-time": "2023-10-24T10:50:33.180340Z",
    "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"
    ]
  }
}
```

Benachrichtigung über Statusaktualisierung der Änderungsanfrage – Beispiel 2

```
{
  "version": "0",
```

```
  "id": "25ce6b03-2e4e-1a2b-2a8f-6c9de8d278d2",
  "detail-type": "Change Request Status Update",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2023-10-24T10:51:52Z",
  "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": "2023-10-24T10:50:33.180334Z",
    "ops-item-modified-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-modified-time": "2023-10-24T10:50:33.997163Z",
    "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"
    ]
  }
}
```

AWS Systems Manager Ereignisse zur Einhaltung von Vorschriften

Im Folgenden finden Sie Beispiele für Veranstaltungen zum Thema AWS Systems Manager Compliance.

Zuordnung regelkonform

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:03:26Z",
```

```
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2017-01-01T10:10:10Z",
    "compliance-status": "compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Zuordnung nicht regelkonform

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:02:31Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2017-01-01T10:10:10Z",
    "compliance-status": "non_compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Patch regelkonform

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.123456789012",
  "account": "123456789012",
```

```
  "time": "2017-07-17T19:03:26Z",
  "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"
  }
}
```

Patch nicht regelkonform

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:02:31Z",
  "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 Manager EreignisseMaintenance Windows

Es folgen Beispiele der Ereignisse für Systems Manager Maintenance Windows.

Registrieren eines Ziels

Der andere gültige Statuswert ist `DEREGISTERED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Target Registration Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2016-11-16T00:58:37Z",
   "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"
   }
}
```

Fensterausführungstyp

Die anderen gültigen Statuswerte sind`PENDING`,`IN_PROGRESS`,`SUCCESS`,`FAILED`,`TIMED_OUT`, und`SKIPPED_OVERLAPPING`aus.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Execution State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2016-11-16T01:00:57Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.427Z",
      "end-time":"2016-11-16T01:00:57.070Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
```

```
      "status":"TIMED_OUT"
   }
}
```

Typ der Aufgabenausführung

Die anderen gültigen Statuswerte sindIN_PROGRESS,SUCCESS,FAILED, undTIMED_OUTaus.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Execution State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2016-11-16T01:00:56Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.759Z",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "end-time":"2016-11-16T01:00:56.847Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "status":"TIMED_OUT"
   }
}
```

Aufgabenziel verarbeitet

Die anderen gültigen Statuswerte sindIN_PROGRESS,SUCCESS,FAILED, undTIMED_OUTaus.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Target Invocation State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2016-11-16T01:00:57Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-123456789012345678"
```

```
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.427Z",
      "end-time":"2016-11-16T01:00:57.070Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "window-target-id":"e7265f13-3cc5-4f2f-97a9-123456789012",
      "status":"TIMED_OUT",
      "owner-information":"Owner"
   }
}
```

Fensterstatusänderung

Die gültigen Werte sind `ENABLED` und `DISABLED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2016-11-16T00:58:37Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "window-id":"mw-123456789012",
      "status":"DISABLED"
   }
}
```

AWS Systems Manager EreignisseParameter Store

Es folgen Beispiele der Ereignisse für Systems Manager Parameter Store.

Create Parameter (Parameter erstellen)

```
{
```

```
  "version": "0",
  "id": "6a7e4feb-b491-4cf7-a9f1-bf3703497718",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:43:48Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:parameter/MyExampleParameter"
  ],
  "detail": {
    "operation": "Create",
    "name": "MyExampleParameter",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

Update Parameter (Parameter aktualisieren)

```
{
  "version": "0",
  "id": "9547ef2d-3b7e-4057-b6cb-5fdf09ee7c8f",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:44:48Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:parameter/MyExampleParameter"
  ],
  "detail": {
    "operation": "Update",
    "name": "MyExampleParameter",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

DeleteParameter (Parameter löschen)

```
{
  "version": "0",
```

```
  "id": "80e9b391-6a9b-413c-839a-453b528053af",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:45:48Z",
  "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 Manager EreignisseOpsCenter

OpsCenter OpsItem Benachrichtigung erstellen

```
{
  "version": "0",
  "id": "aae66adc-7aac-f0c0-7854-7691e8c079b8",
  "detail-type": "OpsItem Create",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2023-10-19T02:48:11Z",
  "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": "2023-10-19T02:46:53.629361Z",
    "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"
  }
}
```

OpsCenter OpsItem Benachrichtigung aktualisieren

```
{
  "version": "0",
  "id": "2fb5b168-b725-41dd-a890-29311200089c",
  "detail-type": "OpsItem Update",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2023-10-19T02:48:11Z",
  "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": "2023-10-19T02:46:54.049271Z",
    "modified-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "modified-time": "2023-10-19T02:46:54.337354Z",
    "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 Manager EreignisseRun Command

Run Command-Statusänderungsbenachrichtigung

```
{
    "version": "0",
    "id": "51c0891d-0e34-45b1-83d6-95db273d1602",
    "detail-type": "EC2 Command Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2016-07-10T21:51:32Z",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-abcd1111"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
```

```
        "expire-after": "2016-07-14T22:01:30.049Z",
        "parameters": {
            "executionTimeout": ["3600"],
            "commands": ["date"]
        },
        "requested-date-time": "2016-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

Run CommandInvocation Status-Änderungsbenachrichtigung

```
{
    "version": "0",
    "id": "4780e1b8-f56b-4de5-95f2-95db273d1602",
    "detail-type": "EC2 Command Invocation Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2016-07-10T21:51:32Z",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-abcd1111"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "instance-id": "i-9bb89e2b",
        "requested-date-time": "2016-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

AWS Systems Manager EreignisseState Manager

State Manager-Association-Statusänderung

```
{
   "version":"0",
   "id":"db839caf-6f6c-40af-9a48-25b2ae2b7774",
   "detail-type":"EC2 State Manager Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2017-05-16T23:01:10Z",
   "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":"2017-02-13T17:22:54.458Z",
      "last-successful-execution-date":"2017-05-16T23:00:01Z",
      "last-execution-date":"2017-05-16T23:00:01Z",
      "last-updated-date":"2017-02-13T17:22:54.458Z",
      "status":"Success",
      "association-status-aggregated-count":"{\"Success\":1}",
      "schedule-expression":"cron(0 */30 * * * ? *)",
      "association-cwe-version":"1.0"
   }
}
```

State Manager-Instance-Association-Statusänderung

```
{
   "version":"0",
   "id":"6a7e8feb-b491-4cf7-a9f1-bf3703467718",
   "detail-type":"EC2 State Manager Instance Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2017-02-23T15:23:48Z",
   "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-12345678",
      "document-name":"my-custom-document",
      "document-version":"1",
      "targets":"[{\"key\":\"instanceids\",\"values\":[\"i-12345678\"]}]",
      "creation-date":"2017-02-23T15:23:48Z",
      "last-successful-execution-date":"2017-02-23T16:23:48Z",
      "last-execution-date":"2017-02-23T16:23:48Z",
      "status":"Success",
```

```
      "detailed-status":"",
      "error-code":"testErrorCode",
      "execution-summary":"testExecutionSummary",
      "output-url":"sampleurl",
      "instance-association-cwe-version":"1"
   }
}
```

Beispielszenarien: Systems-Manager-Ziele in Amazon-EventBridge-Regeln

Wenn Sie das aufzurufende Ziel in einer Amazon EventBridge-Regel angeben, können Sie zwischen mehr als 20 Zieltypen auswählen und jeder Regel bis zu fünf Ziele hinzufügen.

Von den verschiedenen Zielen können Sie aus Automation, OpsCenter und Run Command als Zielaktionen wählen, welche Funktionen von AWS Systems Manager sind, wenn ein EventBridge Ereignis eintritt.

Im Folgenden finden Sie einige Beispiele, wie Sie diese Funktionen als Ziel einer EventBridge-Regel verwenden können.

Beispiele zur Automation

Sie können eine EventBridge-Regel so konfigurieren, dass Automation-Workflows gestartet werden, wenn Ereignisse wie die folgenden auftreten:

- Wenn ein Amazon-CloudWatch-Alarm meldet, dass ein verwalteter Knoten eine Statusprüfung nicht bestanden hat (`StatusCheckFailed_Instance=1`), führen Sie das `AWSSupport-ExecuteEC2Rescue`-Automatisierungs-Runbook auf dem Knoten aus.
- Wenn ein `EC2 Instance State-change Notification`-Ereignis auftritt, weil eine neue Amazon Elastic Compute Cloud (Amazon EC2)-Instance ausgeführt wird, führen Sie das aus `AWS-AttachEBSVolume`–Automation-Runbook auf der Instance aus.
- Wenn ein Amazon Elastic Block Store (Amazon EBS)-Volume erstellt wurde und verfügbar ist, führen Sie das aus `AWS-CreateSnapshot`–Automation-Runbook auf dem Volume aus.

Beispiele für OpsCenter

Sie können eine EventBridge Regel konfigurieren, um ein neues OpsItem zu erstellen, wenn Vorfälle wie die folgenden Bedingungen eintreten:

- Ein Drosselungsereignis für Amazon DynamoDB tritt auf, oder die Leistung des Amazon EBS-Volumes hat sich verschlechtert.
- Eine Amazon-EC2-Auto-Scaling-Gruppe kann einen Knoten nicht starten, oder ein Systems-Manager-Automatisierungs-Workflow schlägt fehl.
- Eine EC2-Instance ändert den Status von `Running` auf `Stopped`.

Beispiele für Run Command

Sie können eine EventBridge-Regel zum Ausführen eines Systems Manager Befehlsdokuments in Run Command konfigurieren, wenn Ereignisse wie die folgenden auftreten:

- Wenn eine Auto-Scaling-Gruppe kurz vor dem Ende steht, könnte ein Run Command-Skript die Protokolldateien des Knotens erfassen, bevor er beendet wird.
- Wenn ein neuer Knoten in einer Auto-Scaling-Gruppe erstellt wird, könnte eine Run Command-Zielaktion die Webserver-Rolle aktivieren oder Software auf dem Knoten installieren.
- Wenn ein verwalteter Knoten als nicht konform befunden wird, könnte eine Run Command-Zielaktion Patches auf dem Knoten aktualisieren, indem das `AWS-RunPatchBaseline`-Dokument ausgeführt wird.

Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen

> **Note**
>
> FIFO-Themen des Amazon Simple Notification Service werden nicht unterstützt.

Sie können Amazon Simple Notification Service (Amazon SNS) zum Senden von Benachrichtigungen über den Status der Befehle konfigurieren, die Sie mithilfe von Run Command oder Maintenance Windows, welche Funktionen von AWS Systems Manager sind, senden. Amazon SNS koordiniert und verwaltet das Senden und Zustellen von Benachrichtigungen an Clients oder Endpunkte, die Amazon SNS-Themen abonniert haben. Sie können eine Benachrichtigung erhalten, wenn ein Befehl in einen neuen Status oder in einen bestimmten Status wechselt, z. B. Ausgefallen oder Timeout. In Fällen, in denen Sie einen Befehl an mehrere Knoten senden, können Sie eine Benachrichtigung für

jede Kopie des Befehls abrufen, die an einen bestimmten Knoten gesendet wurde. Jede Kopie wird als Aufruf bezeichnet.

Amazon SNS kann Benachrichtigungen als HTTP oder HTTPS POST, E-Mail (SMTP, entweder im Klartext oder im JSON-Format) oder als Nachricht, die an eine Amazon Simple Queue Service (Amazon SQS)-Queue gesendet wird, verschicken. Weitere Informationen finden Sie unter Was ist Amazon SNS im Amazon Simple Notification Service-Entwicklerhandbuch. Beispiele für die Struktur der JSON-Daten in der Amazon SNS-Benachrichtigung, die von Run Command und Maintenance Windows bereitgestellt wird, finden Sie unter Beispiele für Amazon SNS-Benachrichtigungen für AWS Systems Manager.

Konfigurieren von Amazon SNS-Benachrichtigungen für AWS Systems Manager

Run Command und Maintenance Windows-Aufgaben, die für ein Wartungsfenster registriert sind, können Amazon SNS-Benachrichtigungen zu Befehlsaufgaben senden, die in folgende Status wechseln:

- In Bearbeitung
- Herzlichen Glückwunsch
- Fehlgeschlagen
- Timed Out (Zeitüberschreitung)
- Canceled

Weitere Informationen über die Bedingungen, die dazu führen, dass ein Befehl in einen dieser Status wechselt, finden Sie unter Grundlegendes zu Befehlsstatus.

> **Note**
>
> Befehle, die mit Run Command gesendet wurden, melden auch die Status „Cancelling (Abbrechen)“ und „Pending (Ausstehend)“. Diese Status werden nicht von Amazon SNS-Benachrichtigungen erfasst.

Amazon SNS-Benachrichtigungen

Wenn Sie Run Command oder eine Run Command-Aufgabe in Ihrem Wartungsfenster für Amazon SNS-Benachrichtigungen konfigurieren, sendet Amazon SNS zusammenfassende Nachrichten mit folgenden Informationen.

Feld	Typ	Beschreibung
eventTime	Zeichenfolge	Der Zeitpunkt, an dem das Ereignis initiiert wurde. Der Zeitstempel ist wichtig, weil Amazon SNS keine Garantie für die Reihenfol ge der Nachrichtenzustell ung übernimmt. Beispiel: 2016-04-26T13:15:30Z
documentName	Zeichenfolge	Der Name des SSM-Dokum ents, das zur Ausführung dieses Befehls verwendet wurde.
commandId	Zeichenfolge	Die von Run Command erstellte ID, nachdem der Befehl gesendet wurde.
expiresAfter	Datum	Wenn diese Zeit erreicht ist und der Befehl noch nicht gestartet wurde, wird er nicht ausgeführt.
outputS3BucketName	Zeichenfolge	Der Amazon Simple Storage Service (Amazon S3)-Bucket, in dem die Antworten auf die Befehlsausführung gespeiche rt werden sollten.

Feld	Typ	Beschreibung
outputS3KeyPrefix	Zeichenfolge	Der Amazon S3-Verzeichnispfad innerhalb des Buckets, in dem die Antworten auf die Befehlsausführung gespeichert werden sollten.
requestedDateTime	Zeichenfolge	Die Uhrzeit und das Datum, an dem die Anforderung an diesen spezifischen Knoten gesendet wurde.
instanceIds	StringList	Die Knoten, auf die der Befehl abzielt hat. **Note** Instance-IDs sind nur in der zusammenfassenden Nachricht enthalten, wenn die Run Command-Aufgabe direkt auf Instance-IDs abzielt. Instance-IDs sind nicht Bestandteil der zusammenfassenden Nachricht, wenn die Run Command-Aufgabe mithilfe von Tag-basiertem Targeting ausgestellt wurde.
status	Zeichenfolge	Befehlsstatus für den Befehl.

Aufrufbasierte Amazon SNS-Benachrichtigungen

Wenn Sie einen Befehl an mehrere Knoten senden, kann Amazon SNS Nachrichten über jede Kopie oder jeden Aufruf des Befehls senden. Die Nachrichten enthalten folgende Angaben.

Feld	Typ	Beschreibung
eventTime	Zeichenfolge	Der Zeitpunkt, an dem das Ereignis initiiert wurde. Der Zeitstempel ist wichtig, weil Amazon SNS keine Garantie für die Reihenfol ge der Nachrichtenzustell ung übernimmt. Beispiel: 2016-04-26T13:15:30Z
documentName	Zeichenfolge	Der Name des System Manager-Dokuments (SSM-Dokument), das zur Ausführun g dieses Befehls verwendet wurde.
requestedDateTime	Zeichenfolge	Die Uhrzeit und das Datum, an dem die Anforderung an diesen spezifischen Knoten gesendet wurde.
commandId	Zeichenfolge	Die von Run Command erstellte ID, nachdem der Befehl gesendet wurde.
instanceId	Zeichenfolge	Die Instance, auf die der Befehl abzielt.
status	Zeichenfolge	Befehlsstatus für diesen Aufruf.

Um Amazon SNS-Benachrichtigungen einzurichten, wenn ein Befehl seinen Status ändert, führen Sie die folgenden Aufgaben aus.

> **Note**
>
> Wenn Sie keine Amazon SNS-Benachrichtigungen für Ihr Wartungsfenster konfigurieren, können Sie Aufgabe 5 weiter unten in diesem Thema überspringen.

Themen

- Aufgabe 1: Erstellen und Abonnieren eines Amazon SNS-Themas
- Aufgabe 2: Erstellen einer IAM-Richtlinie für Amazon SNS-Benachrichtigungen
- 3Aufgabe 2: Erstellen einer IAM-Role für Amazon SNS-Benachrichtigungen
- Aufgabe 4: Konfigurieren des Benutzerzugriffs
- Aufgabe 5: Anfügen der iam:PassRole-Richtlinie an die Wartungsfenster-Rolle

Aufgabe 1: Erstellen und Abonnieren eines Amazon SNS-Themas

Ein Amazon SNS-Thema ist ein Kommunikationskanal, über den Run Command und Run Command -Aufgaben, die in einem Wartungsfenster registriert sind, Benachrichtigungen über den Status Ihrer Befehle senden. Amazon SNS unterstützt verschiedene Kommunikationsprotokolle, einschließlich HTTP/S, E-Mail und andere AWS-Services wie Amazon Simple Queue Service (Amazon SQS). Für den Anfang empfehlen wir, mit dem E-Mail-Protokoll anzufangen. Weitere Informationen zum Erstellen eines Themas finden Sie unter Erstellen eines Amazon-SNS-Themas im Entwicklerhandbuch zu Amazon Simple Notification Service.

> **Note**
>
> Nachdem Sie das Thema erstellt haben, kopieren oder notieren Sie sich die Thema-ARN. Sie legen diesen ARN fest, wenn Sie einen Befehl senden, der so konfiguriert wurde, dass er Statusbenachrichtigungen zurückgibt.

Nachdem Sie das Thema erstellt haben, abonnieren Sie es, indem Sie einen Endpunkt angeben. Wenn Sie das E-Mail-Protokoll gewählt haben, ist der Endpunkt die E-Mail-Adresse, an die Sie Benachrichtigungen erhalten möchten. Weitere Informationen zum Abonnieren eines Themas finden

Sie unter Abonnieren eines Amazon-SNS-Themas im Entwicklerhandbuch zu Amazon Simple Notification Service.

Amazon SNS sendet eine Bestätigungs-E-Mail von AWS Notifications an die von Ihnen angegebene E-Mail-Adresse. Öffnen Sie die E-Mail und wählen Sie den Link Abonnement bestätigen aus.

Sie erhalten eine Bestätigungsnachricht von AWS. Amazon SNS ist jetzt so konfiguriert, dass Benachrichtigungen empfangen und als E-Mail an die angegebene E-Mail-Adresse gesendet werden.

Aufgabe 2: Erstellen einer IAM-Richtlinie für Amazon SNS-Benachrichtigungen

Gehen Sie wie folgt vor, um eine benutzerdefinierte AWS Identity and Access Management (IAM)-Richtlinie zu erstellen, die Berechtigungen zum Auslösen von Amazon SNS-Benachrichtigungen bereitstellt.

So erstellen Sie eine benutzerdefinierte IAM-Richtlinie für Amazon SNS-Benachrichtigungen

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies und dann Create Policy. (Wenn die Schaltfläche Get Started (Erste Schritte) angezeigt wird, klicken Sie darauf und wählen Sie anschließend Create Policy (Richtlinie erstellen) aus.)
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie den Standardinhalt durch folgenden Inhalt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:region:account-id:sns-topic-name"
        }
    ]
}
```

region repräsentiert die Kennung für eine von AWS-Region unterstützte AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *Region*-

Werte finden Sie in der Spalte Region unter Systems-Manager-Service-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

account-id repräsentiert den 12-stelligen Bezeichner für Ihr AWS-Konto im Format 123456789012.

sns-topic-name repräsentiert den Namen des Amazon-SNS-Themas, das Sie zum Veröffentlichen von Benachrichtigungen verwenden möchten.

5. Wählen Sie Next: Markierungen (Weiter: Markierungen).
6. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern.
7. Wählen Sie Weiter: Prüfen aus.
8. Geben Sie auf der Seite Review Policy (Richtlinie prüfen) im Feld Name (Name) einen Namen für die Inline-Richtlinie ein. Beispiel: **my-sns-publish-permissions**.
9. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für die Richtlinie ein.
10. Wählen Sie Create Policy (Richtlinie erstellen) aus.

3Aufgabe 2: Erstellen einer IAM-Role für Amazon SNS-Benachrichtigungen

Verwenden Sie das folgende Verfahren zum Erstellen einer IAM-Rolle für Amazon SNS-Benachrichtigungen. Diese Service-Rolle wird von Systems Manager verwendet, um Amazon SNS-Benachrichtigungen zu initiieren. In allen nachfolgenden Verfahren wird diese Rolle als Amazon SNS IAM-Rolle bezeichnet.

So erstellen Sie eine IAM-Service-Rolle für Amazon SNS-Benachrichtigungen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Roles und wählen Sie dann Create role.
3. Wählen Sie den AWS-Service-Rollentyp und dann Systems Manager aus.
4. Wählen Sie den Systems-Manager-Anwendungsfall aus. Wählen Sie anschließend Weiter aus.
5. Markieren Sie auf der Seite Attach permissions policies (Richtlinien für Berechtigungen anfügen) das Kontrollkästchen links neben dem Namen der benutzerdefinierten Richtlinie aus, die Sie in Aufgabe 2 erstellt haben. Beispiel: **my-sns-publish-permissions**.

6. (Optional) Legen Sie eine Berechtigungsgrenze fest. Dies ist ein erweitertes Feature, die für Servicerollen verfügbar ist, aber nicht für servicegebundene Rollen.

   Öffnen Sie den Abschnitt Permissions boundary (Berechtigungsgrenze) und wählen Sie Use a permissions boundary to control the maximum role permissions (Eine Berechtigungsgrenze verwenden, um die maximalen Rollen-Berechtigungen zu steuern). IAM enthält eine Liste der von AWS verwalteten und vom Kunden verwaltete Richtlinien in Ihrem Konto. Wählen Sie die Richtlinie aus, die für die Berechtigungsgrenze verwendet werden soll, oder wählen Create policy (Richtlinie erstellen), um eine neue Registerkarte im Browser zu öffnen und eine vollständig neue Richtlinie zu erstellen. Weitere Informationen finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch. Nachdem Sie die Richtlinie erstellt haben, schließen Sie die Registerkarte und kehren zur ursprünglichen Registerkarte zurück, um die Richtlinie auszuwählen, die für die Berechtigungsgrenze verwendet werden soll.
7. Wählen Sie Next (Weiter).
8. Geben Sie möglichst einen Rollennamen oder ein Rollennamen-Suffix ein, mit dem der Zweck dieser Rolle einfach zu erkennen ist. Rollennamen müssen innerhalb Ihres AWS-Konto-Kontos eindeutig sein. Es wird hierbei nicht zwischen Groß- und Kleinschreibung unterschieden. Beispielsweise können Sie keine Rollen erstellen, die **`PRODROLE`** bzw. **`prodrole`** heißen. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung nicht bearbeitet werden.
9. (Optional) Geben Sie unter Role description (Rollenbeschreibung) eine Beschreibung für die neue Rolle ein.
10. Wählen Sie in den Abschnitten Step 1: Select trusted entities (Schritt 1: Vertrauenswürdige Entitäten auswählen) oder Step 2: Add permissions (Schritt 2: Berechtigungen hinzufügen) die Option Edit (Bearbeiten), um die Anwendungsfälle und Berechtigungen für die Rolle zu bearbeiten.
11. (Optional) Fügen Sie dem Benutzer Metadaten hinzu, indem Sie Markierungen als Schlüssel-Wert-Paare anfügen. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.
12. Prüfen Sie die Rolle und klicken Sie dann auf Create Role (Rolle erstellen).
13. Wählen Sie den Namen der Rolle und kopieren Sie dann oder notieren Sie sich den Wert der Role ARN (Rollen-ARN). Dieser Amazon-Ressourcenname (ARN) für die Rolle wird verwendet, wenn Sie einen Befehl senden, der so konfiguriert wurde, dass er Amazon-SNS-Benachrichtigungen zurückgibt.
14. Lassen Sie die Seite Summary (Übersicht) geöffnet.

Aufgabe 4: Konfigurieren des Benutzerzugriffs

Wenn einer IAM-Entität (Benutzer, Rolle oder Gruppe) Administratorberechtigungen zugewiesen wurden, hat der Benutzer oder die Rolle Zugriff auf Run Command und Maintenance Windows, Funktionen von AWS Systems Manager.

Für Entitäten ohne Administratorrechte muss ein Administrator der IAM-Entität die folgenden Berechtigungen gewähren:

- Die von `AmazonSSMFullAccess` verwaltete Richtlinie oder eine Richtlinie, die vergleichbare Berechtigungen bereitstellt.
- `iam:PassRole`-Berechtigungen für die Rolle, die in 3Aufgabe 2: Erstellen einer IAM-Role für Amazon SNS-Benachrichtigungen erstellt wurde. Beispiele:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::account-id:role/sns-role-name"
        }
    ]
}
```

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center-Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter Erstellen einer Rolle für einen externen Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter Erstellen einer Rolle für einen IAM-Benutzer im IAM-Benutzerhandbuch.

- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

So konfigurieren Sie Benutzerzugriff und fügen die Richtlinie **`iam:PassRole`** an ein Benutzerkonto an

1. Wählen Sie im IAM-Navigationsbereich die Option Users (Benutzer) und anschließend das Benutzerkonto aus, das Sie konfigurieren möchten.
2. Prüfen Sie auf der Registerkarte Permissions (Berechtigungen) in der Richtlinienliste, ob entweder die **`AmazonSSMFullAccess`**-Richtlinie aufgeführt ist, oder ob es eine vergleichbare Richtlinie gibt, die dem Konto Berechtigungen für den Zugriff auf Systems Manager erteilt.
3. Wählen Sie Add inline Policy (Inline-Richtlinie auswählen).
4. Wählen Sie auf der Seite Create policy die Registerkarte Visual editor aus.
5. Wählen Sie Choose a service und dann IAM aus.
6. Geben Sie unter Actions (Aktionen) in das Textfeld Filter actions (Aktionen filtern) **`PassRole`** ein und wählen Sie das Kontrollkästchen PassRole aus.
7. Vergewissern Sie sich bei Resources (Ressourcen), dass Specific (spezifisch) ausgewählt ist, und wählen Sie dann Add ARN (ARN hinzufügen).
8. Fügen Sie im Feld Specify ARN for role (ARN für Rolle angeben) den ARN der Amazon SNS IAM-Rolle ein, den Sie am Ende von Aufgabe 3 kopiert haben. Das System füllt die Felder Account (Konto) und Role name with path (Rollenname mit Pfad) automatisch aus.
9. Wählen Sie Add (Hinzufügen) aus.
10. Wählen Sie Review policy (Richtlinie prüfen).
11. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) einen Namen ein und wählen Sie anschließend Create Policy (Richtlinie erstellen) aus.

Aufgabe 5: Anfügen der iam:PassRole-Richtlinie an die Wartungsfenster-Rolle

Wenn Sie eine Run Command-Aufgabe mit einem Wartungsfenster registrieren, geben Sie den Amazon-Ressourcennamen (ARN) einer Servicerolle an. Diese Servicerolle wird von Systems Manager zur Durchführung von Aufgaben verwendet, die beim Wartungsfenster registriert sind. Hängen Sie eine `iam:PassRole`-Richtlinie an die angegebene Wartungsfenster-Servicerolle an, um Amazon SNS-Benachrichtigungen für eine registrierte Run Command-Aufgabe zu konfigurieren

Wenn Sie nicht beabsichtigen, die registrierte Aufgabe für Amazon SNS-Benachrichtigungen zu konfigurieren, können Sie diese Aufgabe überspringen.

Mithilfe der `iam:PassRole`-Richtlinie kann die Maintenance Windows-Servicerolle die in Aufgabe 3 erstellte IAM-Rolle an den Amazon SNS-Service übergeben. Das folgende Verfahren zeigt, wie die `iam:PassRole`-Richtlinie an die Maintenance Windows-Servicerolle angehängt wird.

> **Note**
>
> Verwenden Sie eine benutzerdefinierte Servicerolle für Wartungsfenster, um Benachrichtigungen mit Bezug zu den registrierten Run Command-Aufgaben zu senden. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.
> Wenn Sie eine benutzerdefinierte Servicerolle für Wartungsfenster-Aufgaben erstellen müssen, finden Sie weitere Informationen dazu unter Konfigurieren Sie mit der Konsole Berechtigungen für Wartungsfenster.

Anhängen der **`iam:PassRole`**-Richtlinie an Ihre Maintenance Windows-Rolle

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich erst Roles (Rollen) und dann die in Aufgabe 3 erstellte Amazon SNS IAM-Rolle aus.
3. Kopieren oder notieren Sie den Role ARN (Rollen-ARN) und kehren Sie zum Abschnitt Roles (Rollen) der IAM-Konsole zurück.
4. Wählen Sie die benutzerdefinierte Maintenance Windows-Servicerolle aus, die Sie aus der Liste Role name (Rollennamen) erstellt haben.
5. Stellen Sie in der Registerkarte Permissions (Berechtigungen) sicher, dass entweder die `AmazonSSMMaintenanceWindowRole`-Richtlinie oder eine vergleichbare Richtlinie angegeben ist, durch die Wartungsfenster Berechtigungen für die Systems-Manager-API erhalten. Wenn dies nicht der Fall ist, wählen Sie Berechtigungen hinzufügen, Richtlinien anfügen, um sie anzufügen.
6. Wählen Sie Add permissions, Create inline policy (Berechtigungen hinzufügen, eingebundene Richtlinie erstellen).
7. Wählen Sie die Registerkarte Visual Editor (Visueller Editor) aus.
8. Wählen Sie unter Service die Option IAM aus.

9. Geben Sie unter Actions (Aktionen) in das Textfeld Filter actions (Aktionen filtern) **`PassRole`** ein und wählen Sie das Kontrollkästchen PassRole aus.
10. Wählen Sie unter Resources (Ressourcen), die Option Specific (Spezifisch) und dann Add ARN (ARN hinzufügen)aus.
11. Fügen Sie im Feld Specify ARN for role (ARN für Rolle angeben) den ARN der in Aufgabe 3 erstellten Amazon SNS IAM-Rolle ein und wählen Sie dann Add (Hinzufügen) aus.
12. Wählen Sie Review policy (Richtlinie prüfen).
13. Geben Sie auf der Seite Richtlinie überprüfen einen Namen für die `PassRole`-Richtlinie an und wählen Sie dann Richtlinie erstellen aus.

Beispiele für Amazon SNS-Benachrichtigungen für AWS Systems Manager

Sie können Amazon Simple Notification Service (Amazon SNS) zum Senden von Benachrichtigungen über den Status der Befehle konfigurieren, die Sie mithilfe von Run Command oder Maintenance Windows, welche Funktionen von AWS Systems Manager sind, senden.

> **Note**
>
> Diese Anleitung enthält keine Beschreibungen darüber, wie Benachrichtigungen für Run Command oder Maintenance Windows konfiguriert werden. Weitere Informationen zum Konfigurieren von Run Command oder Maintenance Windows zum Senden von Amazon SNS-Benachrichtigungen über den Status von Befehlen finden Sie unter Konfigurieren von Amazon SNS-Benachrichtigungen für AWS Systems Manager.

Die folgenden Beispiele zeigen die Struktur der JSON-Ausgabe, die von Amazon SNS-Benachrichtigungen zurückgeben wird, wenn eine Konfiguration für Run Command oder Maintenance Windows durchgeführt wurde.

Beispiel einer JSON-Ausgabe für zusammenfassende Nachrichten zu Befehlen mithilfe von Instance-ID-Targeting

```
{
    "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": "DOC-EXAMPLE-BUCKET",
    "outputS3KeyPrefix": "runcommand",
    "status": "InProgress",
    "eventTime": "2019-04-25T17:57:09.236Z"
}
```

Beispiel einer JSON-Ausgabe für zusammenfassende Nachrichten zu Befehlen mithilfe von Tag-basiertem Targeting

```
{
    "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"
}
```

Beispiel einer JSON-Ausgabe für Nachrichten zu Aufrufen

```
{
    "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"
}
```

Verwenden von Run Command zum Senden eines Befehls, der Statusbenachrichtigungen zurückgibt

Die folgenden Verfahren zeigen, wie Sie mithilfe der AWS Command Line Interface (AWS CLI) oder AWS Systems Manager -Konsole einen Befehl über eine Funktion von AWS Systems Manager sendenRun Command, die für die Rückgabe von Statusbenachrichtigungen konfiguriert ist.

Senden eines Run Command, der Benachrichtigungen zurückgibt (Konsole)

Mit der folgenden Vorgehensweise können Sie mittels Run Command einen Befehl senden, der so konfiguriert ist, dass Statusbenachrichtigungen mithilfe der Systems Manager-Konsole zurückgeben werden.

So senden Sie einen Befehl, der Benachrichtigungen zurückgibt (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document ein Systems Manager-Dokument.
5. Geben Sie im Abschnitt Command parameters Werte für erforderliche Parameter an.
6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Other parameters (Weitere Parameter):

   - Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und

Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   Note

   Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

10. Wählen Sie im Abschnitt SNS Notifications (SNS-Benachrichtigungen)die Option Enable SNS notifications (SNS-Benachrichtigungen aktivieren) aus.
11. Wählen Sie für IAM role (IAM-Rolle) den ARN Amazon SNS IAM-Rolle aus, den Sie in Aufgabe 3 in Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen erstellt haben.
12. Geben Sie für SNS-Thema den ARN für das Amazon SNS-Thema an, welcher verwendet werden soll.
13. Wählen Sie für Event notifications (Ereignisbenachrichtigungen) die Ereignisse aus, für die Sie Benachrichtigungen erhalten möchten.

14. Wählen Sie für Change notifications (Änderungsbenachrichtigen), ob Sie Benachrichtigungen nur für die Befehlsübersicht erhalten möchten (Command status changes (Befehls-Statusänderungen)) oder für jede Kopie eines Befehls, der an mehrere Knoten (Command status on each instance changes (Befehls-Statusänderungen bei jeder Instance)) gesendet wurde.
15. Wählen Sie Ausführen aus.
16. Überprüfen Sie, ob Sie eine Nachricht von Amazon SNS erhalten haben, und öffnen Sie die E-Mail-Nachricht. Es kann einige Minuten dauern, bis Amazon SNS die E-Mail-Nachricht sendet.

Senden eines Run Command, der Benachrichtigungen zurückgibt (CLI)

Gehen Sie wie folgt vor, um mittels Run Command einen Befehl zu senden, der so konfiguriert ist, dass er mithilfe von AWS CLI Statusbenachrichtigungen zurückgibt.

So senden Sie einen Befehl, der Statusbenachrichtigungen zurückgibt (CLI)

1. Öffnen Sie das AWS CLI.
2. Geben Sie in folgendem Befehl Parameter an, um anhand von IDs von verwalteten Knoten Ziele anzuvisieren.

```
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"}'
```

Im Folgenden sehen Sie ein Beispiel.

```
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"}'
```

Alternative Befehle

Geben Sie im folgenden Befehl Parameter ab, um auf verwaltete Instances abzuzielen, die Tags verwenden.

```
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"}'
```

Im Folgenden sehen Sie ein Beispiel.

```
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. Drücken Sie die Eingabetaste.
4. Überprüfen Sie, ob Sie eine Nachricht von Amazon SNS erhalten haben, und öffnen Sie die E-Mail-Nachricht. Es kann einige Minuten dauern, bis Amazon SNS die E-Mail-Nachricht sendet.

Weitere Informationen finden Sie unter send-command in der Referenz zum AWS CLI -Befehl.

Verwenden eines Wartungsfensters zum Senden eines Befehls, der Statusbenachrichtigungen zurückgibt

Die folgenden Verfahren zeigen, wie Sie mithilfe der AWS Systems Manager Konsole oder der AWS Command Line Interface (AWS CLI) eine Run Command Aufgabe in Ihrem Wartungsfenster registrieren. Run Commandist eine Fähigkeit von AWS Systems Manager. Zudem wird erläutert, wie die Run Command-Aufgabe so konfiguriert wird, dass Statusbenachrichtigungen zurückgegeben werden.

Bevor Sie beginnen

Wenn Sie kein Wartungsfenster erstellt oder Ziele registriert haben, informieren Sie sich unter Arbeiten mit Wartungsfenstern (Konsole), wie ein Wartungsfenster erstellt wird und Ziele registriert werden.

Zum Abrufen von Benachrichtigungen über den Amazon Simple Notification Service (Amazon SNS)-Service müssen Sie eine `iam:PassRole`-Richtlinie an die Maintenance Windows-Servicerolle anhängen, die in der registrierten Aufgabe angegeben ist. Wenn Sie Ihrer Maintenance Windows-

Servicerolle keine `iam:PassRole`-Berechtigungen hinzugefügt haben, finden Sie weitere Informationen unter Aufgabe 5: Anfügen der iam:PassRole-Richtlinie an die Wartungsfenster-Rolle.

Registrieren einer Run Command-Aufgabe bei einem Wartungsfenster, die Benachrichtigungen zurückgibt (Konsole)

Gehen Sie wie folgt vor, um eine Run Command-Aufgabe zu registrieren, die so konfiguriert ist, dass mithilfe der Systems Manager-Konsole Statusbenachrichtigungen an Ihr Wartungsfenster zurückgegeben werden.

So registrieren Sie eine Run Command-Aufgabe mit Ihrem Wartungsfenster, die Benachrichtigungen zurückgibt (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie das Wartungsfenster aus, für das Sie eine Run Command-Aufgabe registrieren möchten, mit der Amazon Simple Notification Service (Amazon SNS)-Benachrichtigungen gesendet werden sollen.
4. Wählen Sie Actions (Aktionen) und anschließend Register Run command task (Run command-Aufgabe registrieren) aus.
5. (Optional) Geben Sie im Feld Name einen Namen für die Aufgabe ein.
6. (Optional) Geben Sie in das Feld Description eine Beschreibung ein.
7. Wählen Sie für Command document (Befehlsdokument) ein Befehlsdokument aus.
8. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in der Reihenfolge ihrer Priorität geplant. Aufgaben mit derselben Priorität werden parallel geplant.
9. Wählen Sie im Abschnitt Targets (Ziele) eine registrierte Zielgruppe aus, oder wählen Sie nicht registrierte Ziele aus.
10. Für Rate control (Ratenregelung):
    - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

11. Wählen Sie im Bereich IAM service role (Servicerolle) die Maintenance Windows-Servicerolle aus, die `iam:PassRole`-Berechtigungen für die SNS-Rolle hat.

> **Note**
>
> Fügen Sie der Maintenance Windows-Rolle `iam:PassRole`-Berechtigungen hinzu, damit Systems Manager die SNS-Rolle an Amazon SNS weitergeben kann. Wenn Sie keine `iam:PassRole`-Berechtigungen hinzugefügt haben, finden Sie weitere Informationen unter Aufgabe 5 im Thema Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

12. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-

Konto, dass das Instanzprofil oder die IAM-Dienstrolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

13. Führen Sie im Abschnitt SNS notifications (Benachrichtigungen) folgende Schritte aus:
    - Wählen Sie Enable SNS Notifications (Aktivieren von SNS-Benachrichtigungen).
    - Wählen Sie als IAM role (IAM-Rolle) den Amazon SNS IAM-Rolle Amazon-Ressourcennamen (ARN), den Sie in Aufgabe 3 in Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen erstellt haben, um Amazon SNS zu initiieren.
    - Geben Sie für SNS-Thema den ARN für das Amazon SNS-Thema an, welcher verwendet werden soll.
    - Wählen Sie unter Event type (Ereignistyp) die Ereignisse aus, für die Sie Benachrichtigungen erhalten möchten.
    - Wählen Sie für das Feld Notification type (Benachrichtigungstyp) aus, dass Sie Benachrichtigen für jede Kopie eines Befehls erhalten, der an mehrere Knoten (Aufrufe) gesendet wird, oder die Befehls-Zusammenfassung erhalten.
14. Geben Sie im Abschnitt Parameters (Parameter) die erforderlichen Parameter anhand des ausgewählten Befehlsdokuments ein.
15. Wählen Sie Register Run command task.
16. Sehen Sie nach der nächsten Ausführung Ihres Wartungsfensters in Ihrem Posteingang nach, ob Sie eine Nachricht von Amazon SNS erhalten haben und öffnen Sie die E-Mail-Nachricht. Es kann einige Minuten dauern, bis Amazon SNS die E-Mail-Nachricht sendet.

Registrieren einer Run Command-Aufgabe bei einem Wartungsfenster, die Benachrichtigungen zurückgibt (CLI)

Gehen Sie wie folgt vor, um eine Run Command-Aufgabe zu registrieren, die so konfiguriert ist, dass mithilfe der AWS CLI Statusbenachrichtigungen an Ihr Wartungsfenster zurückgegeben werden.

So registrieren Sie eine Run Command Aufgabe mit Ihrem Wartungsfenster, die Benachrichtigungen zurückgibt (CLI)

> **Note**
>
> Zur besseren Koordination der Aufgabenoptionen wird bei dieser Vorgehensweise die Befehlsoption `--cli-input-json` verwendet. Dabei sind die Optionswerte in einer JSON-Datei gespeichert.

1. Erstellen Sie auf dem lokalen Computer eine Datei mit dem Namen `RunCommandTask.json`.
2. Fügen Sie den folgenden Inhalt in die -Datei ein:

```
{
    "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. Ersetzen Sie die Beispielwerte mit Informationen über Ihre eigenen Ressourcen.

   Sie können auch Optionen wiederherstellen, die bei diesem Beispiel ausgelassen wurden, sofern Sie diese verwenden möchten. So kann die Befehlsausgabe in einem S3-Bucket gespeichert werden.

   Weitere Informationen finden Sie unter register-task-with-maintenance-window in der Referenz zum AWS CLI -Befehl.

4. Speichern Sie die Datei.

5. Führen Sie auf dem lokalen Computer in dem Verzeichnis, in dem die Datei gespeichert wurde, folgenden Befehl aus.

   ```
   aws ssm register-task-with-maintenance-window --cli-input-json file://
   RunCommandTask.json
   ```

   > **Important**
   >
   > Achten Sie darauf, dass `file://` vor dem Dateinamen steht. Dies ist bei diesem Befehl erforderlich.

   Wenn der Befehl erfolgreich ausgeführt wurde, sieht das Ergebnis in etwa wie folgt aus.

   ```
   {
       "WindowTaskId": "j2l8d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"
   }
   ```

6. Sehen Sie nach der nächsten Ausführung Ihres Wartungsfensters in Ihrem Posteingang nach, ob Sie eine Nachricht von Amazon SNS erhalten haben und öffnen Sie die E-Mail-Nachricht. Es kann einige Minuten dauern, bis Amazon SNS die E-Mail-Nachricht sendet.

Weitere Informationen zum Registrieren von Aufgaben für ein Wartungsfenster in der Befehlszeile finden Sie unter Register tasks with the maintenance window (Registrieren von Aufgaben im Wartungsfenster).

Produkt- und Service-Integrationen mit Systems Manager

AWS Systems Manager ist standardmäßig mit AWS-Services sowie andere Produkte und Services integriert. Die folgenden Informationen können Ihnen die Konfiguration von Systems Manager zum Integrieren in die von Ihnen verwendeten Produkte und Services erleichtern.

- Integration mit AWS-Services
- Integration in andere Produkte und Services

Integration mit AWS-Services

Durch die Verwendung von Systems Manager Manager-Befehlsdokumenten (SSM-Dokumenten) und Automation-Runbooks können Sie die Integration AWS Systems Manager mit verwenden. AWS-Services Weitere Informationen zu diesen Ressourcen finden Sie unter AWS Systems Manager-Documents.

Systems Manager ist in Folgendes integriert AWS-Services.

Datenverarbeitung

Amazon Elastic Compute Cloud (Amazon EC2)	Amazon EC2 stellt skalierbare Rechenkapazität in der AWS Cloud-Cloud bereit. Amazon EC2 beseitigt die Notwendigkeit, im Voraus in Hardware investieren zu müssen. Daher können Sie Anwendungen schneller entwickeln und bereitstellen. Mit Amazon EC2 können Sie so viele oder so wenige virtuelle Server starten, wie Sie benötigen, die Sicherheit und das Netzwerk konfigurieren und den Speicher verwalten. Systems Manager ermöglicht es Ihnen, mehrere Aufgaben auf EC2-Instances auszuführen. Sie können beispielsweise Ihre EC2-Instances starten, konfigurieren, verwalten, instandhalten, reparieren und eine sichere

	Verbindung zu Ihren EC2-Instances herstellen. Sie können Systems Manager auch verwenden , um Software bereitzustellen, den Compliance-Status zu ermitteln und Inventar aus Ihren EC2-Instances zu erfassen. Weitere Informationen • Mit verwalteten Knoten arbeiten • 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-Bestand
Amazon EC2 Auto Scaling	Auto Scaling hilft Ihnen sicherzustellen, dass Sie die richtige Anzahl von EC2-Instances zur Verfügung haben, um die Auslastung Ihrer Anwendung zu bewältigen. Sie erstellen Sammlungen von EC2 Instances, die als Auto Scaling-Gruppen bezeichnet werden. Systems Manager ermöglicht es Ihnen, gängige Prozeduren wie das Patchen des Amazon Machine Image (AMI) zu automatisieren, die in Ihrer Auto Scaling-Vorlage für Ihre Auto-Scaling-Gruppe verwendet wird. Weitere Informationen Aktualisieren von AMIs für Auto-Scaling-Gruppen

Amazon Elastic Container Service (Amazon ECS)	Amazon ECS ist ein hoch skalierbarer, schneller Container-Management-Service, der das Ausführen, Beenden und Verwalten von Docker-Containern in einem Cluster ermöglicht. Mit Systems Manager können Sie Container-Instance remote verwalten und sensible Daten in Ihre Container einspeisen, indem Sie Ihre sensiblen Daten in Parametern in Parameter Store speichern, eine Funktion von Systems Manager, und dann in Ihrer Container-Definition darauf verweisen. Weitere Informationen • Remote-Verwaltung von Container-Instances mit AWS Systems Manager • Angeben sensibler Daten mithilfe des Systems Manager-Parameter Parameter Store

AWS Lambda	Lambda ist ein Datenverarbeitungsservice, mit dem Sie Code ausführen können, ohne Server bereitstellen oder verwalten zu müssen. Lambda führt Ihren Code nur bei Bedarf aus und skaliert automatisch – von einigen Anforderungen pro Tag bis zu Tausenden pro Sekunde. Systems Manager ermöglicht es Ihnen, Lambda-Funktionen in Automation-Runbook-Inhalten mithilfe der `aws:invokeLambdaFunction` -Aktion zu verwenden. Um Parameter aus Parameter Store AWS Lambda Funktionen zu verwenden, können Sie die Lambda-Erweiterung Parameters and Secrets verwenden, um Parameterwerte abzurufen und sie für die future Verwendung AWS zwischenzuspeichern. Weitere Informationen Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store Verwenden von Parameter Store-Parametern in AWS Lambda -Funktionen

Internet of Things (IoT)

AWS IoT Greengrass Kerngeräte	AWS IoT Greengrass ist ein Open-Source IoT-Edge-Laufzeit- und Cloud-Service, mit dem Sie IoT-Anwendungen auf Ihren Geräten entwickeln, bereitstellen und verwalten können. Systems Manager bietet native Unterstützung für AWS IoT Greengrass Kerngeräte.

	Weitere Informationen Verwaltung von Edge-Geräten mit Systems Manager
AWS IoT Kerngeräte	AWS IoTstellt die Cloud-Dienste bereit, die Ihre IoT-Geräte mit anderen Geräten und AWS Cloud-Diensten verbinden. AWS IoT biete t Gerätesoftware, mit der Sie Ihre IoT-Gerät e in AWS IoT basierte Lösungen integrieren können. Wenn Ihre Geräte eine Verbindung herstellen können AWS IoT, AWS IoT können Sie sie mit den bereitgestellten Cloud-Diensten verbinden. AWS Systems Manager unterstützt AWS IoT Kerngeräte, sofern diese Geräte als verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung konfiguriert sind. Weitere Informationen Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen

Speicher

Amazon-Simple-Storage-Service (Amazon-S3)	Bei Amazon S3 handelt es sich um Speicher für das Internet. Der Service ist darauf ausgelegt, Cloud Computing für Entwickler zu erleichtern. Amazon S3 besitzt eine einfache Web-Service-Schnittstelle zum Speichern und Abrufen einer beliebigen Datenmenge zu jeder Zeit und von jedem Ort im Internet aus. Mit Systems Manager können Sie Remote-Skripts und SSM-Dokumente ausführen, die in Amazon S3 gespeichert sind. Distributor, eine Funktion von AWS Systems Manager,

	verwendet Amazon S3 zum Speichern von Paketen. Sie können die Ausgabe auch an Amazon S3 für Run Command undSessio n Manager, Funktionen von senden AWS Systems Manager. Weitere Informationen • Ausführen von Skripts von Amazon S3 • Ausführen von -Dokumenten von Remote-St andorten • AWS Systems Manager Distributor • Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)

Entwicklertools

AWS CodeBuild	CodeBuildist ein vollständig verwalteter Build-Service in der Cloud. CodeBuild kompiliert Ihren Quellcode, führt Komponententests durch und erzeugt Artefakte, die sofort einsatzbe reit sind. CodeBuild macht die Bereitstellung, Verwaltung und Skalierung Ihrer eigenen Build-Server überflüssig. Mit Parameter Store können Sie vertrauliche Informationen für Ihre Build-Spezifikationen und Projekte speichern. Weitere Informationen • Referenz zur Build-Spezifikation für CodeBuild • Erstellen Sie ein Build-Projekt in AWS CodeBuild

AWS CDK	Das AWS Cloud Development Kit (AWS CDK) ist ein Framework für die Definition der Cloud-Infrastruktur als Code mit Programmiersprachen und deren Bereitstellung AWS CloudFormation. Application Manager ermöglicht es Ihnen, Ihre CDK-Konstrukte als Anwendungen gruppiert zu betrachten, die Anwendungsstruktur einschließlich der zugrundeliegenden Ressourcen einzusehen, Warnungen anzuzeigen, Betriebsprobleme zu untersuchen und zu beheben und Kosten in der Application Manager-Konsole zu verfolgen. Weitere Informationen • Anzeigen von Übersichtsinformationen einer Anwendung • Anzeigen von Anwendungsressourcen

Sicherheit, Identität und Compliance

AWS Identity and Access Management (IAM)	IAM ist ein Webservice, mit dem Sie den Zugriff auf Ressourcen sicher kontrollieren können. AWS Sie verwenden IAM, um zu steuern, wer authentifiziert (angemeldet) und autorisiert (Berechtigungen besitzt) ist, Ressourcen zu nutzen. Systems Manager ermöglicht es Ihnen, den Zugriff auf Dienste mithilfe von IAM zu steuern. Weitere Informationen • Funktionsweise von AWS Systems Manager mit IAM

	• Aktionen, Ressourcen und Bedingungsschlüssel für AWS Systems Manager • Konfigurieren Sie die für Systems Manager erforderlichen Instanzberechtigungen
AWS Secrets Manager	Secrets Manager bietet eine einfachere Verwaltung von Secrets. Bei den Secrets kann es sich um Datenbank-Anmeldeinformationen, Passwörter, API-Schlüssel von Drittanbietern und sogar beliebigen Text handeln. Mit Parameter Store können Sie Secrets-Manager-Geheimnisse abrufen, wenn Sie andere AWS-Services verwenden, die bereits Referenzen zu Parameter Store-Parametern unterstützen. Weitere Informationen Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter

AWS Security Hub	Security Hub bietet einen umfassenden Überblick über Ihre Sicherheitswarnungen und den Compliance-Status von hoher Priorität über AWS-Konten hinweg. Security Hub aggregier t, organisiert und priorisiert Ihre Sicherhei tswarnungen oder Ergebnisse aus mehreren. AWS-Services Wenn Sie die Integration zwischen Security Hub und Patch Manager Security Hub aktiviere n AWS Systems Manager, überwacht Security Hub den Patch-Status Ihrer Flotten aus Sicherheitsgründen. Details zur Patch-Com pliance werden automatisch in den Security Hub exportiert. Auf diese Weise können Sie mit einer einzigen Ansicht den Patch-Com pliance-Status zentral überwachen und andere Sicherheitsergebnisse nachverfolgen. Sie können Warnungen erhalten, wenn Knoten in Ihrer Flotte die Patch-Compliance verletzen, und die Ergebnisse der Patch-Compliance in der Security-Hub-Konsole überprüfen. Sie können Security Hub auch mit Explorer undOpsCenter, Funktionen von integrier en AWS Systems Manager. Integration mit Security Hub ermöglicht Ihnen, Ergebnisse von Security Hub im Explorer zu OpsCenter erhalten. Die Ergebnisse des Security Hub stellen Sicherheitsinformationen bereit, die Sie in Explorer und OpsCenter verwenden können, um Ihre Sicherheits-, Leistungs- und Betriebsp robleme in Systems Manager zu aggregieren und Maßnahmen in AWS Systems Manager zu ergreifen.

	Für die Nutzung von Security Hub wird eine Gebühr erhoben. Weitere Informationen finden Sie unter Security Hub. Weitere Informationen • Empfangen von Ergebnissen von AWS Security Hub in Explorer • AWS Security Hub • Integrieren Patch Manager mit AWS Security Hub

Kryptografie und PKI

AWS Key Management Service (AWS KMS)	AWS KMS ist ein verwalteter Service, der es Ihnen ermöglicht, kundenverwaltete Schlüssel zu erstellen und zu kontrollieren, d. h. die Verschlüsselungsschlüssel, die zur Verschlüsselung Ihrer Daten verwendet werden. Mit Systems Manager können AWS KMS Sie `SecureString` Parameter erstellen und Session Manager Sitzungsdaten verschlüsseln. Weitere Informationen • Verwendung von AWS Systems ManagerParameter Store durch AWS KMS • So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole)

Verwaltung und Governance

AWS CloudFormation	AWS CloudFormation ist ein Service, der die Entwicklung und Einrichtung von Amazon

Web Services-Ressourcen erleichtert, sodass Sie weniger Zeit für die Verwaltung dieser Ressourcen aufwenden müssen und sich stattdessen mehr auf Ihre Anwendungen, die in AWS ausgeführt werden, konzentrieren können.

Parameter Store ist eine Quelle für dynamisch e Referenzen. Dynamische Verweise bieten eine kompakte und leistungsstarke Möglichke it, externe Werte anzugeben, die in anderen Diensten in Ihren AWS CloudFormation Stack-Vorlagen gespeichert und verwaltet werden.

Weitere Informationen

Verwenden von dynamischen Referenzen zum Angeben von Vorlagenwerten

AWS CloudTrail	CloudTrailist ein Programm AWS-Service , das Ihnen dabei hilft, Unternehmensführung, Compliance sowie Betriebs- und Risikoprüfungen Ihres AWS-Konto Unternehmens zu autorisieren. Aktionen, die von einem Benutzer, einer Rolle oder einem ausgeführt werden, AWS-Service werden als Ereignisse in CloudTrail aufgezeichnet. Zu den Ereignissen gehören Aktionen AWS Management Console, die in den AWS SDKs und APIs, AWS Command Line Interface (AWS CLI) ausgeführt wurden. Systems Manager ist integriert und erfasst CloudTrail die meisten Systems Manager Manager-API-Aufrufe als Ereignisse. Diese beinhalten API-Aufrufe, die von der Systems-Manager-Konsole initiiert werden, und Aufrufe der Systems-Manager-APIs. Weitere Informationen AWS Systems Manager API-Aufrufe protokollieren mit AWS CloudTrail

CloudWatch Amazon-Protokolle	Mit Amazon CloudWatch Logs können Sie die Protokolle all Ihrer Systeme und Anwendungen, AWS-Services die Sie verwenden, zentralisieren. Sie können sie dann anzeigen, nach bestimmten Fehlercodes oder Mustern suchen, sie anhand bestimmter Felder filtern oder sicher für zukünftige Analysen archivieren. Systems Manager unterstützt das Senden von Protokollen für die SSM AgentRun Command, und Session Manager an CloudWatch Logs. Weitere Informationen • Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent) • Konfiguration von Amazon CloudWatch Logs für Run Command • Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

Amazon EventBridge	EventBridgeliefert einen Stream von Systemereignissen nahezu in Echtzeit, der Änderungen an den Ressourcen von Amazon Web Services beschreibt. Mithilfe einfacher Regeln, die Sie schnell einrichten können, können Sie Ereignisse zuordnen und sie an eine oder mehrere Zielfunktionen oder Streams weiterleiten. EventBridge wird sich betrieblicher Änderungen bewusst, sobald sie auftreten. EventBridge reagiert auf diese betrieblichen Änderungen und ergreift gegebenenfalls Korrekturmaßnahmen. Dazu gehören das Senden von Nachrichten zur Reaktion auf die Umgebung, das Aktivieren von Funktionen und das Erfassen von Statusinformationen. Systems Manager verfügt über mehrere Ereignisse, die unterstützt werden, EventBridge sodass Sie auf der Grundlage des Inhalts dieser Ereignisse Maßnahmen ergreifen können. Weitere Informationen Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge **Note** Amazon EventBridge ist die bevorzugte Methode, um Ihre Veranstaltungen zu verwalten. CloudWatch Bei Events und EventBridge handelt es sich um denselben zugrunde liegenden Service und dieselbe API, EventBridge bieten aber mehr Funktionen. Änderungen, die Sie in einer CloudWatch oder mehreren Konsolen vornehmen, EventBridge

	spiegeln sich in jeder Konsole wider. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch.
AWS Config	AWS Configbietet einen detaillierten Überblick über die Konfiguration der AWS Ressourcen in Ihrem AWS-Konto. Dazu gehört auch, wie die Ressourcen jeweils zueinander in Beziehung stehen und wie sie konfiguriert wurden. Auf diese Weise können Sie sehen, wie sich die Konfigurationen und Beziehungen im Laufe der Zeit ändern. Systems Manager ist in integriert und bietet mehrere Regeln AWS Config, die Ihnen helfen, einen Überblick über Ihre EC2-Instances zu erhalten. Anhand dieser Regeln können Sie ermitteln, welche EC2-Instances von Systems Manager verwaltet werden, Betriebssystemkonfigurationen, Updates auf Systemebene, installierte Anwendungen, Netzwerkkonfigurationen und mehr. Weitere Informationen • AWS Config unterstützte Ressourcentypen • Aufzeichnen der Software-Konfiguration für verwaltete Instances • Anzeigen von Bestandsverlauf und Änderungsnachverfolgung

AWS Trusted Advisor	Trusted Advisor ist ein Online-Tool, das Sie in Echtzeit dabei unterstützt, Ihre Ressourcen gemäß den bewährten Methoden von AWS bereitzustellen. Systems Manager hostet Trusted Advisor und Sie können Trusted Advisor Daten in anzeigenExplorer. Weitere Informationen • AWS Systems Manager Explorer • Erste Schritte mit AWS Trusted Advisor

AWS Organizations	Organizations ist ein Kontoverwaltungsdienst, mit dem Sie mehrere Konten zu einer Organisation AWS-Konten zusammenfassen können, die Sie erstellen und zentral verwalten. Organisationen umfasst Kontoverwaltungs- und konsolidierte Fakturierung, mit denen Sie die Budget-, Sicherheits- und Compliance-Anforderungen Ihres Unternehmens besser erfüllen können. Die Integration zwischen Change Manager, eine Funktion von AWS Systems Manager, mit Organizations ermöglicht es, ein delegiertes Administratorkonto zu verwenden, um Änderungsanfragen, Änderungsvorlagen und Genehmigungen für Ihre gesamte Organisation über dieses einzige Konto zu verwalten. Die Integration von Organizations in Inventory, eine Funktion von und Explorerermöglicht es Ihnen AWS Systems Manager, Bestands- und Betriebsdaten (OpsData) aus mehreren AWS-Regionen und zu aggregieren AWS-Konten. Die Integration zwischenQuick Setup, eine Funktion von AWS Systems Manager und Organizations automatisiert allgemeine Aufgaben zur Einrichtung von Diensten und stellt Servicekonfigurationen auf der Grundlage von Best Practices in Ihren Organisationseinheiten (OUs) bereit.

Netzwerk und Bereitstellung von Inhalten

AWS PrivateLink	AWS PrivateLinkermöglicht es Ihnen, Ihre Virtual Private Cloud (VPC) privat mit unterstützten AWS-Services und VPC-Endpunktdienst

	en zu verbinden, ohne dass ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder AWS Direct Connect eine Verbindung erforderlich ist. Systems Manager unterstützt verwaltete Knoten, die mit Systems Manager APIs über AWS PrivateLink verbinden. Dies verbessert die Sicherheitslage Ihrer verwalteten Knoten, da der gesamte Netzwerkverkehr zwischen Ihren verwalteten Knoten, Systems Manager und Amazon EC2 auf das Amazon-Netzwerk AWS PrivateLink beschränkt wird. Dies bedeutet, dass verwaltete Knoten keinen Zugriff auf das Internet haben müssen. Weitere Informationen Verbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager

Analysen

Amazon Athena	Athena ist ein interaktiver Abfrageservice, der die direkte Analyse von Daten in Amazon Simple Storage Service (Amazon S3) mit Standard-SQL ermöglicht. Mit einigen Aktionen in der AWS Management Console können Sie Athena auf Ihre in Amazon S3 gespeicherten Daten verweisen und beginnen, Standard-SQL zu verwenden, um einmalige Abfragen auszuführen und innerhalb von Sekunden Ergebnisse zu erhalten. Systems Manager Inventory ist in Athena integriert, sodass Sie Inventardaten von

	mehreren AWS-Regionen und AWS-Konten abfragen können. Die Athena-Integration verwendet Ressourcen-Datensynchronisierung, sodass Sie Bestandsdaten aus allen verwalteten Knoten auf der Seite Detailed View (Detailansicht) in der Systems-Manager-Inventory-Konsole anzeigen können. Weitere Informationen • Abfragen von Bestandsdaten aus mehreren Regionen und Konten • Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten
AWS Glue	AWS Glue ist ein vollständig verwalteter ETL-Service (Extrahieren, Transformieren und Laden), mit dessen Hilfe einfach und wirtschaftlich Ihre Daten kategorisiert, bereinigt, erweitert und zwischen verschiedenen Datenspeichern und Datenströmen verschoben werden können. Systems Manager wird verwendet AWS Glue , um die Inventardaten in Ihrem S3-Bucket zu crawlen. Weitere Informationen Abfragen von Bestandsdaten aus mehreren Regionen und Konten

Amazon QuickSight	Amazon QuickSight ist ein Geschäftsanalysedienst, mit dem Sie Visualisierungen erstellen, einmalige Analysen durchführen und Geschäftserkenntnisse aus Ihren Daten gewinnen können. Es kann automatisch AWS -Datenquellen erkennen und arbeitet auch mit Ihren Datenquellen. Die Ressourcen-Datensynchronisierung von Systems Manager sendet die von all Ihren verwalteten Knoten erfassten Bestandsdaten an einen einzigen S3-Bucket. Sie können Amazon verwenden QuickSight , um die aggregierten Daten abzufragen und zu analysieren. Weitere Informationen • Konfigurieren von Resource Data Sync für Inventory • Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten

Anwendungsintegration

Amazon-Simple-Notification-Service (Amazon-SNS)	Amazon SNS ist ein Webservice, der die Zustellung oder das Senden von Nachrichten an abonnierende Endpunkte oder Clients koordiniert und verwaltet. Systems Manager generiert Status für mehrere Dienste, die von Amazon SNS-Benachrichtigungen erfasst werden können.

	Weitere Informationen • Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen • Einrichten von Benachrichtigungen oder Auslöseraktionen basierend auf Parameter Store-Ereignissen

AWS Management Console

AWS Resource Groups	Resource Groups organisieren Ihre AWS Ressourcen. Ressourcengruppen vereinfachen die gleichzeitige Verwaltung, Überwachung und Automatisierung von Aufgaben für eine große Zahl von Ressourcen. Systems-Manager-Ressourcentypen wie verwaltete Knoten, SSM-Dokumente, Wartungsfenster, Parameter Store-Parameter und Patch-Baselines können Ressourcengruppen hinzugefügt werden. Weitere Informationen Was sind AWS Resource Groups?

Themen

- Ausführen von Skripts von Amazon S3
- Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter
- Verwenden von Parameter Store-Parametern in AWS Lambda -Funktionen

Ausführen von Skripts von Amazon S3

In diesem Abschnitt wird beschrieben, wie Skripts von Amazon Simple Storage Service (Amazon S3) heruntergeladen und ausgeführt werden. Das folgende Thema enthält Informationen und Terminologie zu Amazon S3. Weitere Informationen zu Amazon S3 finden Sie unter Was ist Amazon S3? Sie können verschiedene Arten von Skripten ausführen, darunter Ansible Playbooks, Python, Ruby, Shell und. PowerShell

Sie können auch ein Verzeichnis mit mehreren Skripts herunterladen. Wenn Sie das primäre Skript im Verzeichnis ausführen, werden AWS Systems Manager auch alle referenzierten Skripten ausgeführt, die im Verzeichnis enthalten sind.

Beachten Sie die folgenden wichtigen Hinweise zum Ausführen von Skripts von Amazon S3:

- Systems Manager prüft nicht, ob Ihr Skript auf einem Knoten ausgeführt werden kann. Stellen Sie sicher, dass die erforderliche Software auf dem Knoten installiert ist, bevor Sie das Skript herunterladen und ausführen. Alternativ können Sie ein zusammengesetztes Dokument erstellen, das die Software über Run Command oder State Manager, eine Funktion von AWS Systems Manager, installiert und das Skript anschließend herunterlädt und ausführt.
- Stellen Sie sicher, dass Ihrem Benutzer, Ihrer Rolle oder Gruppe die AWS Identity and Access Management (IAM)-Berechtigungen gewährt wurden, die zum Lesen aus dem S3-Bucket erforderlich sind.
- Stellen Sie sicher, dass das Instance-Profil auf Ihren Amazon Elastic Compute Cloud (Amazon EC2)-Instances über `s3:ListBucket` und `s3:GetObject`-Berechtigungen verfügt. Wenn das Instance-Profil nicht über diese Berechtigungen verfügt, kann das System Ihr Skript nicht aus dem S3-Bucket herunterladen. Weitere Informationen finden Sie unter Verwenden von Instance-Profilen im IAM-Benutzerhandbuch.

Ausführen von Shell-Scripts von Amazon S3

Die folgenden Informationen enthalten Verfahren, die Ihnen helfen, Skripts von Amazon Simple Storage Service (Amazon S3) entweder über die AWS Systems Manager Konsole oder die AWS Command Line Interface (AWS CLI) auszuführen. Obwohl Shell-Skripte in den Beispielen verwendet werden, können andere Arten von Skripten ersetzt werden.

Ausführen eines Shell-Skripts von Amazon S3 (Konsole)

Ausführen eines Shell-Skripts von Amazon S3

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option **`AWS-RunRemoteScript`** aus.
5. Führen Sie unter Command parameters die folgenden Schritte aus:

   - Wählen Sie unter Source Type die Option S3 aus.
   - Geben Sie im Textfeld Source Info die für den Zugriff auf die Quelle erforderlichen Informationen im folgenden Format an. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

     > **Note**
     >
     > Ersetzen Sie https: //s3.*`aws-api-domain`* durch die URL für Ihren Bucket. Sie können Ihre Bucket-URL in Amazon S3 auf der Registerkarte Objects (Objekte) kopieren.

     ```
     {"path":"https://s3.aws-api-domain/path to script"}
     ```

     Im Folgenden wird ein Beispiel gezeigt.

     ```
     {"path":"https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/scripts/shell/
     helloWorld.sh"}
     ```

   - Geben Sie im Feld Command Line Parameter für die Skriptausführung ein. Ein Beispiel.

     ```
     helloWorld.sh argument-1 argument-2
     ```

   - (Optional) Geben Sie im Feld Working Directory (Arbeitsverzeichnis) den Namen eines Verzeichnisses auf dem Knoten ein, auf dem das Skript heruntergeladen und ausgeführt werden soll.

- (Optional) Geben Sie unter Execution Timeout die Dauer in Sekunden an, bis das System die Skriptbefehlausführung fehlschlagen lässt.

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Other parameters (Weitere Parameter):

   - Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

   > **Note**
   >
   > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

10. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Wählen Sie Ausführen aus.

Ausführen eines Shell-Skripts von Amazon S3 (Befehlszeile)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

> **Note**
>
> Ersetzen Sie https: //s3.*aws-api-domain* durch die URL für Ihren Bucket. Sie können Ihre Bucket-URL in Amazon S3 auf der Registerkarte Objects (Objekte) kopieren.

Linux & macOS

```
aws ssm send-command \
    --document-name "AWS-RunRemoteScript" \
    --output-s3-bucket-name "bucket-name" \
    --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 "bucket-name" ^
    --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 "bucket-name" `
    -OutputS3KeyPrefix "key-prefix" `
    -Target @{Key="InstanceIds";Values=@("instance-id")} `
    -Parameter @{ sourceType="S3";sourceInfo='{"path": "https://s3.aws-api-
domain/script path"}',; "commandLine"="script name and arguments"}
```

Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter

AWS Secrets Manager hilft Ihnen, wichtige Konfigurationsdaten wie z. B. Anmeldeinformationen, Passwörter und Lizenzschlüssel zu organisieren und zu verwalten. Parameter Store, eine Funktion von AWS Systems Manager, ist jetzt mit Secrets Manager integriert, sodass Sie Secrets-Manager-Geheimnisse abrufen können, wenn Sie andere AWS-Services verwenden, die bereits Verweise auf Parameter Store-Parameter unterstützen. Zu diesen Services gehören Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Container Service (Amazon ECS), AWS Lambda, AWS CloudFormation, AWS CodeBuild, AWS CodeDeploy und andere Systems Manager-Funktionen. Indem Sie Parameter Store zum Verweisen auf Secrets Manager-Geheimnisse verwenden, erstellen Sie einen konsistenten und sicheren Prozess zum Aufrufen und Verwenden von Geheimnissen und zum Referenzieren von Daten in Ihrem Code und den Konfigurationsskripts.

Weitere Informationen zu Secrets Manager finden Sie unter Was ist AWS Secrets Manager? im AWS Secrets Manager-Benutzerhandbuch.

Einschränkungen

Beachten Sie die folgenden Einschränkungen bei der Verwendung von Parameter Store zur Referenzierung von Secrets Manager-Geheimnissen:

- Sie können Secrets Manager-Geheimnisse nur mit den API-Aktionen GetParameter und GetParameters abrufen. Änderungsoperationen und frühzeitige Abfragen von API-Operationen, wie z. B. DescribeParameters oder GetParametersByPath werden für Secrets Manager nicht unterstützt.
- Sie können die AWS Command Line Interface (AWS CLI), AWS Tools for Windows PowerShell und die SDKs zum Abrufen eines Geheimnisses mit Hilfe von Parameter Store verwenden.
- Wenn Sie ein Secrets-Manager-Secret von Parameter Store abrufen, muss der Name mit dem folgenden reservierten Pfad beginnen: /aws/reference/secretsmanager/*secret-_ID*.

  Ein Beispiel: `/aws/reference/secretsmanager/CFCreds1`
- Parameter Store berücksichtigt AWS Identity and Access Management (IAM)-Richtlinien, die Secrets Manager-Geheimnissen zugeordnet sind. Beispiel: Wenn Benutzer 1 keinen Zugriff auf Geheimnis A hat, kann er das Geheimnis nicht mithilfe von Parameter Store abrufen.
- Parameter, die Secrets Manager-Geheimnisse referenzieren, können die Parameter Store-Versioning- oder Verlaufsfunktionen nicht verwenden.

- Parameter Store berücksichtigt Secrets Manager Versionsstufen. Wenn Sie eine Versionsstufe referenzieren, verwendet diese Buchstaben, Zahlen, einen Punkt (.), Bindestrich (-) oder Unterstrich (_). Alle anderen Symbole, die in der Versionsstufe angegeben sind, führen dazu, dass die Referenz fehlschlägt.

Referenzieren eines Secrets Manager-Geheimnisses mit Parameter Store

Im folgenden Verfahren wird beschrieben, wie Sie ein Secrets Manager-Geheimnis mithilfe von Parameter Store-APIs referenzieren. Das Verfahren referenziert weitere Verfahren im AWS Secrets Manager-Benutzerhandbuch.

> **Note**
>
> Bevor Sie beginnen, stellen Sie sicher, dass Sie über die Berechtigung zum Referenzieren von Secrets Manager-Geheimnissen in Parameter Store-Parametern verfügen. Wenn Sie über Administratorrechte in Secrets Manager und Systems Manager verfügen, können Sie Geheimnisse mithilfe von Parameter Store-APIs. Wenn Sie ein Secrets Manager-Geheimnis in einem Parameter Store-Parameter referenzieren und nicht über die Berechtigung für den Zugriff auf dieses Geheimnis verfügen, schlägt die Referenz fehl. Weitere Informationen finden Sie unter Authentifizierung und Zugriffskontrolle für AWS Secrets Manager im AWS Secrets Manager-Benutzerhandbuch.

> **Important**
>
> Parameter Store funktioniert als Pass-Through-Service für Verweise auf Secrets Manager-Geheimnisse. Parameter Store bewahrt keine Daten oder Metadaten über Geheimnisse auf. Die Referenz ist zustandslos.

Referenzieren eines Secrets Manager-Geheimnisses mit Parameter Store

1. Erstellen Sie ein Geheimnis in Secrets Manager. Weitere Informationen finden Sie unter Erstellen und Verwalten von Secrets mit AWS Secrets Manager.
2. Referenzieren Sie ein Geheimnis über die AWS CLI, mit AWS Tools for Windows PowerShell oder mit SDK. Wenn Sie ein Secrets Manager-Geheimnis referenzieren, muss der Name mit dem folgenden reservierten Pfad beginnen: `/aws/reference/secretsmanager/`. Durch die

Angabe dieses Pfads wird Systems Manager mitgeteilt, das Geheimnis von Secrets Manager anstelle von Parameter Store abzurufen. Hier sind einige Beispielnamen, die korrekt auf die Geheimnisse des Secrets Managers verweisen, `CFCreds1` und `DBPass` mithilfe von Parameter Store referenzieren.

- `/aws/reference/secretsmanager/CFCreds1`
- `/aws/reference/secretsmanager/DBPass`

Das folgende Java-Codebeispiel referenziert einen in Secrets Manager gespeicherten access-key und einen secret-key. In diesem Codebeispiel wird ein Amazon DynamoDB-Client eingerichtet. Der Code ruft die Konfigurationsdaten und Anmeldeinformationen von Parameter Store ab. Die Konfigurationsdaten werden als Zeichenfolgeparameter in Parameter Store und die Anmeldeinformationen in Secrets Manager gespeichert. Auch wenn die Konfigurationsdaten und Anmeldeinformationen in separaten Diensten gespeichert sind, kann auf beide Datensätze über Parameter Store mithilfe der `GetParameter`-API zugegriffen werden.

```
/**
* 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();
}
```

Hier sind einige AWS CLI-Beispiele. Verwenden des `aws secretsmanager list-secrets`-Befehls, um die Namen Ihrer Geheimnisse zu finden.

AWS CLI Beispiel 1: Referenz durch Verwenden des Geheimnisnamens

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "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",
      }
}
```

AWS CLI-Beispiel 2: Referenz mit der Versions-ID

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "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",
}
```

AWS CLI Beispiel 3: Referenz mit der Versionsstufe

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
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "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",
}
```

Verwenden von Parameter Store-Parametern in AWS Lambda -Funktionen

Parameter Store, eine Funktion von AWS Systems Manager, bietet sicheren, hierarchischen Speicher für die Verwaltung von Konfigurationsdaten und Geheimnissen. Sie können Daten wie Passwörter, Datenbankzeichenfolgen, Amazon Machine Image (AMI) IDs und Lizenzcodes als Parameterwerte speichern.

Um Parameter aus Parameter Store AWS Lambda Funktionen zu verwenden, ohne ein SDK zu verwenden, können Sie die Lambda-Erweiterung AWS Parameters and Secrets verwenden. Diese Erweiterung ruft Parameterwerte ab und speichert sie zur späteren Verwendung. Durch die Verwendung der Lambda-Erweiterung können Sie Ihre Kosten senken, indem Sie die Anzahl der API-Aufrufe auf Parameter Store reduzieren. Durch die Verwendung der Erweiterung kann auch die Latenzzeit verbessert werden, da der Abruf eines zwischengespeicherten Parameters schneller ist als der Abruf von Parameter Store.

Eine Lambda-Erweiterung ist ein begleitender Prozess, der die Fähigkeiten einer Lambda-Funktion erweitert. Eine Erweiterung ist wie ein Client, der parallel zu einem Lambda-Aufruf ausgeführt wird. Dieser parallele Client kann jederzeit während seines Lebenszyklus mit Ihrer Funktion verbunden werden. Weitere Informationen zu Lambda-Erweiterungen finden Sie unter Lambda-Erweiterungs-API im AWS Lambda -Entwicklerhandbuch.

Die Lambda-Erweiterung AWS Parameters and Secrets funktioniert Parameter Store sowohl AWS Secrets Manager für als auch. Informationen zur Verwendung der Lambda-Erweiterung mit Geheimnissen aus Secrets Manager finden Sie unter Verwenden von AWS Secrets Manager Geheimnissen in AWS Lambda Funktionen im AWS Secrets Manager Benutzerhandbuch.

Verwandte Informationen

Verwenden der Lambda-Erweiterung AWS Parameter and Secrets zum Zwischenspeichern von Parametern und Geheimnissen (AWS Compute Blog)

So funktioniert die Erweiterung

Um Parameter in einer Lambda-Funktion ohne die Lambda-Erweiterung zu verwenden, müssen Sie Ihre Lambda-Funktion so konfigurieren, dass sie Konfigurationsaktualisierungen erhält, indem Sie sie in die `GetParameter`-API-Aktion für Parameter Store integrieren.

Wenn Sie die Lambda-Erweiterung AWS Parameters and Secrets verwenden, ruft die Erweiterung den Parameterwert ab Parameter Store und speichert ihn im lokalen Cache. Dann wird der zwischengespeicherte Wert für weitere Aufrufe verwendet, bis er abläuft. Zwischengespeicherte Werte laufen ab, nachdem sie ihre time-to-live (TTL) überschritten haben. Sie können den TTL-Wert mithilfe der Umgebungsvariablen `SSM_PARAMETER_STORE_TTL` konfigurieren, wie weiter unten in diesem Thema erläutert.

Wenn die konfigurierte Cache-TTL nicht abgelaufen ist, wird der zwischengespeicherte Parameterwert verwendet. Wenn die Zeit abgelaufen ist, wird der zwischengespeicherte Wert ungültig und der Parameterwert wird von Parameter Store abgerufen.

Außerdem erkennt das System Parameterwerte, die häufig verwendet werden, und behält sie im Cache bei, während abgelaufene oder nicht verwendete Werte gelöscht werden.

Implementierungsinformationen

Verwenden Sie die folgenden Details, um Ihnen bei der Konfiguration der Lambda-Erweiterung AWS Parameters and Secrets zu helfen.

Authentifizierung

Um Parameter Store-Anfragen zu autorisieren und zu authentifizieren, verwendet die Erweiterung dieselben Anmeldeinformationen wie diejenigen, die zum Ausführen der Lambda-Funktion verwendet werden. Daher muss die AWS Identity and Access Management (IAM) -Rolle, mit der die Funktion ausgeführt wird, über die folgenden Berechtigungen für die Interaktion verfügen: Parameter Store

- `ssm:GetParameter` – Erforderlich, um Parameter von Parameter Store abzurufen
- `kms:Decrypt` – Erforderlich, wenn Sie `SecureString`-Parameter von Parameter Store abrufen

Weitere Informationen finden Sie unter AWS Lambda -Ausführungsrolle im AWS Lambda -Entwicklerhandbuch.

Instanziierung

Lambda instanziiert separate Instances, die der Gleichzeitigkeitsstufe entsprechen, die Ihre Funktion benötigt. Jede Instance ist isoliert und verwaltet ihren eigenen lokalen Cache Ihrer Konfigurationsdaten. Weitere Informationen über Lambda-Instances und Gleichzeitigkeit finden Sie unter Konfigurieren der reservierten Währung im AWS Lambda -Entwicklerhandbuch.

Keine SDK-Abhängigkeit

Die Lambda-Erweiterung AWS Parameters and Secrets funktioniert unabhängig von jeder AWS SDK-Sprachbibliothek. Ein AWS SDK ist nicht erforderlich, um GET-Anfragen an zu Parameter Store stellen.

Localhost-Port

Verwenden Sie `localhost` in Ihren GET-Anfragen. Die Erweiterung stellt Anfragen an den localhost-Port 2773. Sie müssen keinen externen oder internen Endpunkt angeben, um die Erweiterung zu verwenden. Sie können den Port konfigurieren, indem Sie die Umgebungsvariable auf `PARAMETERS_SECRETS_EXTENSION_HTTP_PORT` setzen.

In Python könnte Ihre GET-URL beispielsweise wie im folgenden Beispiel aussehen.

```
parameter_url = ('http://localhost:' + port + '/systemsmanager/parameters/get/?
name=' + ssm_parameter_path)
```

Änderungen an einem Parameterwert, bevor TTL abläuft

Die Erweiterung erkennt keine Änderungen am Parameterwert und führt keine automatische Aktualisierung durch, bevor die TTL abläuft. Wenn Sie einen Parameterwert ändern, schlagen Vorgänge, die den zwischengespeicherten Parameterwert verwenden, möglicherweise fehl, bis der Cache das nächste Mal aktualisiert wird. Wenn Sie häufige Änderungen an einem Parameterwert erwarten, empfehlen wir Ihnen, einen kürzeren TTL-Wert einzustellen.

Anfordern eines Headers

Um Parameter aus dem Erweiterungs-Cache abzurufen, muss der Header Ihrer GET-Anfrage eine `X-Aws-Parameters-Secrets-Token`-Referenz enthalten. Setzen Sie das Token auf `AWS_SESSION_TOKEN`, das von Lambda für alle laufenden Funktionen bereitgestellt wird. Die Verwendung dieses Headers zeigt an, dass sich der Anrufer in der Lambda-Umgebung befindet.

Beispiel

Das folgende Beispiel in Python demonstriert eine einfache Anfrage zum Abrufen des Wertes eines zwischengespeicherten Parameters.

```
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)
```

ARM-Unterstützung

Die Erweiterung unterstützt die ARM-Architektur trotzdem nicht, obwohl AWS-Regionen die x86 Architekturen x86_64 und unterstützt werden.

Eine vollständige Liste der Erweiterungs-ARNs finden Sie unter AWS Parameter und Geheimnisse Lambda Extension ARNs.

Protokollierung

Lambda protokolliert Ausführungsinformationen über die Erweiterung zusammen mit der Funktion mithilfe von Amazon CloudWatch Logs. Standardmäßig protokolliert die Erweiterung eine minimale Menge an CloudWatch Informationen unter. Um weitere Details zu protokollieren, setzen Sie die Umgebungsvariable `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` auf `DEBUG`.

Hinzufügen der Erweiterung zu einer Lambda-Funktion

Um die Lambda-Erweiterung AWS Parameters and Secrets zu verwenden, fügen Sie die Erweiterung als Ebene zu Ihrer Lambda-Funktion hinzu.

Verwenden Sie eine der folgenden Methoden, um die Erweiterung zu Ihrer Funktion hinzuzufügen.

AWS Management Console (Option „Ebene hinzufügen“)

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie Ihre Funktion. Wählen Sie im Bereich Layers (Ebenen) die Option Add a layer (Ebene hinzufügen) aus.
3. Wählen Sie im Bereich Eine Ebene auswählen die Option AWS -Ebenen aus.
4. Wählen Sie für AWS -Ebenen AWS-Parameter und Secrets-Lambda-Erweiterung aus, wählen Sie eine Version und wählen Sie anschließend Hinzufügen aus.

AWS Management Console (ARN-Option angeben)

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie Ihre Funktion. Wählen Sie im Bereich Layers (Ebenen) die Option Add a layer (Ebene hinzufügen) aus.
3. Wählen Sie im Bereich Choose a layer (Ebene auswählen) die Option Specify an ARN (ARN angeben) aus.
4. Geben Sie für Specify an ARN die Erweiterung ARN für Ihre AWS-Region und Architektur ein, und wählen Sie dann Hinzufügen aus.

AWS Command Line Interface

Führen Sie in der AWS CLI den folgenden aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws lambda update-function-configuration \
    --function-name function-name \
    --layers layer-ARN
```

Ähnliche Informationen

Verwenden von Ebenen mit Ihrer Lambda-Funktion

Konfigurieren von Erweiterungen (ZIP-Dateiarchiv)

AWS Parameter und Geheimnisse Umgebungsvariablen der Lambda-Erweiterung

Sie können die Erweiterung konfigurieren, indem Sie die folgenden Umgebungsvariablen ändern. Um die aktuellen Einstellungen zu sehen, setzen Sie `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` auf `DEBUG`. Weitere Informationen finden Sie unter Verwenden von AWS Lambda Umgebungsvariablen im AWS Lambda Entwicklerhandbuch.

> **Note**
>
> AWS Lambda zeichnet Betriebsdetails zur Lambda-Erweiterung und Lambda-Funktion in Amazon CloudWatch Logs auf.

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
`SSM_PARAMETER_STORE_TIMEOUT_MILLIS`	Timeout in Millisekunden für Anfragen an Parameter Store. Ein Wert von 0 (null) gibt an, dass kein Timeout vorliegt.	Nein	Alle ganzen Zahlen	0 (Null)
`SECRETS_MANAGER_TIMEOUT_MILLIS`	Timeout in Millisekunden für Anfragen an Secrets Manager. Ein Wert von 0 (null) gibt an, dass kein Timeout vorliegt.	Nein	Alle ganzen Zahlen	0 (Null)
`SSM_PARAMETER_STORE_TTL`	Maximal gültige Lebensdauer eines Parameters im Cache in Sekunden, bevor er ungültig wird.	Nein	0 (Null) bis 300 Sek. (Fünf Minuten)	300 Sek. (Fünf Minuten)

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
	Ein Wert von 0 (Null) gibt an, dass der Cache umgangen werden soll. Diese Variable wird ignoriert, wenn der Wert für `PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE` 0 (Null) ist.			
`SECRETS_MANAGER_TTL`	Maximal gültige Lebensdauer eines Secrets im Cache in Sekunden, bevor es ungültig wird. Ein Wert von 0 (Null) gibt an, dass der Cache umgangen wurde. Diese Variable wird ignoriert, wenn der Wert für `PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE` 0 (Null) ist.	Nein	0 (Null) bis 300 Sek. (Fünf Minuten)	300 Sek. (5 Minuten)

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
`PARAMETERS_SECRETS_EXTENSION_CACHE_ENABLED`	Bestimmt, ob der Cache für die Erweiterung aktiviert ist. Gültige Werte: `TRUE | FALSE`	Nein	TRUE | FALSE	TRUE
`PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE`	Die maximale Größe des Caches in Bezug auf die Anzahl der Elemente. Ein Wert von 0 (Null) gibt an, dass der Cache umgangen wurde. Diese Variable wird ignoriert, wenn beide Cache-TTL-Werte 0 (Null) sind.	Nein	0 (Null) bis 1 000	1000
`PARAMETERS_SECRETS_EXTENSION_HTTP_PORT`	Der Port für den lokalen HTTP-Server.	Nein	1 — 65535	2773

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
`PARAMETERS_SECRETS_EXTENSION_MAX_CONNECTIONS`	Maximale Anzahl von Verbindungen für die HTTP-Clients, die die Erweiterung verwendet, um Anfragen an Parameter Store oder Secrets Manager zu stellen. Dies ist eine Konfiguration pro Client für die Anzahl der Verbindungen, die sowohl der Secrets-Manager-Client als auch der Parameter Store-Client zu den Backend-Diensten herstellen.	Nein	Mindestens 1; Keine Höchstgrenze.	3

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
`PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL`	Der Detaillierungsgrad, der in Protokollen für die Erweiterung gemeldet wird. Wir empfehlen die Verwendung von `DEBUG` für die meisten Details zu Ihrer Cache-Konfiguration, während Sie die Erweiterung einrichten und testen. Protokolle für Lambda-Operationen werden automatisch an eine zugeordnete CloudWatch Logs-Protokollgruppe übertragen.	Nein	`DEBUG | WARN | ERROR | NONE | INFO`	`INFO`

Beispielbefehle für die Verwendung der AWS Systems Manager Parameter Store- und AWS Secrets Manager -Erweiterung

Die Beispiele in diesem Abschnitt zeigen API-Aktionen zur Verwendung mit der AWS Secrets Manager Erweiterung AWS Systems Manager Parameter Store und.

Beispielbefehle für Parameter Store

Die Lambda-Erweiterung verwendet schreibgeschützten Zugriff auf die GetParameterAPI-Aktion.

Führen Sie zum Aufrufen dieser Aktion einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get?name=parameter-
path&version=version&label=label&withDecryption={true|false}
```

In diesem Beispiel steht *Parameter-Path für den vollständigen Parameternamen*. *Version* und *Label* sind die Selektoren, die für die Verwendung mit der Aktion verfügbar sind. `GetParameter` Dieses Befehlsformat bietet Zugriff auf Parameter in der Standardparameterebene.

> **Note**
>
> Bei Verwendung von GET-Aufrufen müssen Parameterwerte für HTTP codiert werden, um Sonderzeichen zu erhalten. Anstatt beispielsweise einen hierarchischen Pfad wie `/a/b/c` zu formatieren, codieren Sie Zeichen, die als Teil der URL interpretiert werden könnten, wie z. B. `%2Fa%2Fb%2Fc`.

```
GET http://localhost:port/systemsmanager/parameters/get/?name=MyParameter&version=5
```

Um einen Parameter in einer Hierarchie aufzurufen, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get?name=%2Fa%2Fb%2F&label=release
```

Um einen öffentlichen (globalen) Parameter aufzurufen, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get/?name=%2Faws%2Fservice%20list
%2F…
```

Um einen HTTP-GET-Aufruf an ein Secrets-Manager-Secret mithilfe von Parameter Store-Referenzen durchzuführen, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get?name=%2Faws%2Freference
%2Fsecretsmanager%2F…
```

Um einen Aufruf unter Verwendung des Amazon-Ressourcennamens (ARN) für einen Parameter zu tätigen, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get?name=arn:aws:ssm:us-
east-1:123456789012:parameter/MyParameter
```

Um einen Aufruf zu tätigen, der auf einen `SecureString`-Parameter mit Entschlüsselung zugreift, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get?
name=MyParameter&withDecryption=true
```

Sie können angeben, dass Parameter nicht entschlüsselt werden, indem Sie `withDecryption` weglassen oder explizit auf `false` setzen. Sie können auch entweder eine Version oder ein Label angeben, aber nicht beides. Wenn Sie dies tun, wird nur der erste davon verwendet, der in der URL nach dem Fragezeichen (?) steht.

AWS Parameter und Geheimnisse Lambda Extension ARNs

Die folgenden Tabellen enthalten Erweiterungs-ARNs für unterstützte Architekturen und Regionen.

Themen

- Erweiterungs-ARNs für die x86_64- und x86-Architekturen
- Erweiterungs-ARNs für und Architekturen ARM64Mac with Apple silicon

Erweiterungs-ARNs für die x86_64- und x86-Architekturen

Region	ARN
US East (Ohio)	`arn:aws:lambda:us-east-2:590474943231:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:177933569100:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`

Region	ARN
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:997803712105:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
USA West (Oregon)	`arn:aws:lambda:us-west-2:345057560386:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:317013901791:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Asia Pacific (Hongkong)	`arn:aws:lambda:ap-east-1:768336418462:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Region Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:070087711984:layer:AWS-Parameters-and-Secrets-Lambda-Extension:8`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:490737872127:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Asien-Pazifik (Melbourne)	`arn:aws:lambda:ap-southeast-4:090732460067:layer:AWS-Parameters-and-Secrets-Lambda-Extension:1`

Region	ARN
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:176022468876:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:576959938190:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:738900069198:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:044395824272:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:665172237481:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:133490724326:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Canada (Central)	`arn:aws:lambda:ca-central-1:200266452380:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`

Region	ARN
Kanada West (Calgary)	`arn:aws:lambda:ca-west-1:243964427225:layer:AWS-Parameters-and-Secrets-Lambda-Extension:1`
China (Peking)	`arn:aws-cn:lambda:cn-north-1:287114880934:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
China (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:287310001119:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:187925254637:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Europa (Irland)	`arn:aws:lambda:eu-west-1:015030872274:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Europa (London)	`arn:aws:lambda:eu-west-2:133256977650:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Europa (Milan)	`arn:aws:lambda:eu-south-1:325218067255:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`

Region	ARN
Europa (Paris)	`arn:aws:lambda:eu-west-3:780235371811:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Region Europa (Spanien)	`arn:aws:lambda:eu-south-2:524103009944:layer:AWS-Parameters-and-Secrets-Lambda-Extension:8`
Europa (Stockholm)	`arn:aws:lambda:eu-north-1:427196147048:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:148806536434:layer:AWS-Parameters-and-Secrets-Lambda-Extension:1`
Region Europa (Zürich)	`arn:aws:lambda:eu-central-2:772501565639:layer:AWS-Parameters-and-Secrets-Lambda-Extension:8`
Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:832021897121:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:858974508948:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`

Region	ARN
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:933737806257:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
AWS GovCloud (US-Ost)	`arn:aws-us-gov:lambda:us-gov-east-1:129776340158:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`
AWS GovCloud (US-West)	`arn:aws-us-gov:lambda:us-gov-west-1:127562683043:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`

Erweiterungs-ARNs für und Architekturen ARM64Mac with Apple silicon

Region	ARN
US East (Ohio)	`arn:aws:lambda:us-east-2:590474943231:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:177933569100:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`
Region US West (N. California)	`arn:aws:lambda:us-west-1:997803712105:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`

Region	ARN
USA West (Oregon)	`arn:aws:lambda:us-west-2:345057560386:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`
Region Afrika (Kapstadt)	`arn:aws:lambda:af-south-1:317013901791:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`
Region Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:768336418462:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`
Region Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:490737872127:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:176022468876:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`
Asien-Pazifik (Osaka)	`arn:aws:lambda:ap-northeast-3:576959938190:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`
Region Asien-Pazifik (Seoul)	`arn:aws:lambda:ap-northeast-2:738900069198:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`

Region	ARN
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:044395824272:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:665172237481:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:133490724326:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`
Region Kanada (Zentral)	`arn:aws:lambda:ca-central-1:200266452380:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:187925254637:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`
Europa (Irland)	`arn:aws:lambda:eu-west-1:015030872274:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`
Europe (London)	`arn:aws:lambda:eu-west-2:133256977650:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:11`

Region	ARN
Region Europa (Mailand)	`arn:aws:lambda:eu-south-1:325218067255:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`
Region Europa (Paris)	`arn:aws:lambda:eu-west-3:780235371811:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`
Region Europa (Stockholm)	`arn:aws:lambda:eu-north-1:427196147048:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`
Region Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:832021897121:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`
Region Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:933737806257:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8`

Integration in andere Produkte und Services

AWS Systems Manager verfügt über eine integrierte Integration für die in der folgenden Tabelle gezeigten Produkte und Services.

Ansible	Ansible ist eine IT-Automatisierungsplattform, die die Bereitstellung Ihrer Anwendungen und Systeme erleichtert.

	Systems Manager stellt das Systems Manager-Dokument (SSM-Dokument) bereit`AWS-Apply AnsiblePlaybooks` , mit dem Sie State Manager Zuordnungen erstellen können, die Ansible Playbooks ausführen. Weitere Informationen Exemplarische Vorgehensweise: Erstellen von Verknüpfungen, die Playbooks ausführen Ansible
Chef	Chef ist ein IT-Automatisierungstool, das die Bereitstellung Ihrer Anwendungen und Systeme erleichtert. Systems Manager stellt das `AWS-Apply ChefRecipes` SSM-Dokument bereit, mit dem Sie Zuordnungen in State Manager, einer Funktion von AWS Systems Manager, erstellen können, die Chef Rezepte ausführen. Weitere Informationen Exemplarische Vorgehensweise: Erstellen von Verknüpfungen, die Rezepte ausführen Chef Systems Manager ist auch in -Chef InSpecProfile integriert, sodass Sie Compliance-Scans ausführen und konforme und nicht konforme Knoten anzeigen können. Weitere Informationen Verwenden von Chef InSpec Profilen mit Systems Manager Compliance

GitHub	GitHub bietet Hosting für die Versionskontrolle und Zusammenarbeit in der Softwareentwicklung. Systems Manager stellt das SSM-Dokument bereit`AWS-RunDocument`, mit dem Sie andere in gespeicherte SSM-Dokumente ausführen könnenGitHub, und das SSM-Dokument `AWS-RunRemoteScript`, mit dem Sie in gespeicherte Skripts ausführen könnenGitHub. Weitere Informationen • Ausführen von -Dokumenten von Remote-Standorten • Ausführen von Skripts von GitHub
Jenkins	Jenkins ist ein Open-Source-Automatisierungsserver, mit dem Entwickler ihre Software zuverlässig erstellen, testen und bereitstellen können. Automation, eine Funktion von Systems Manager, kann als Post-Build-Schritt verwendet werden, um Anwendungsversionen in Amazon Machine Images(AMIs) vorzuinstallieren. Weitere Informationen Aktualisierung AMIs mithilfe von Automation und Jenkins

ServiceNow	ServiceNow ist ein Enterprise-Service-Managementsystem, mit dem Sie Ihre IT-Services und -Operationen verwalten können. Automation, Change Manager, Incident Manager und OpsCenter, alle Funktionen von Systems Manager, lassen sich ServiceNow mithilfe des AWS Service Management Connector integrieren. Mit dieser Integration können Sie Korrespondenz anzeigen, erstellen, aktualisieren, hinzufügen und Fälle von lösen AWS Support ServiceNow. Weitere Informationen Integration mit ServiceNow

Themen

- Ausführen von Skripts von GitHub
- Verwenden von Chef InSpec Profilen mit Systems Manager Compliance
- Integration mit ServiceNow

Ausführen von Skripts von GitHub

In diesem Thema wird beschrieben, wie Sie das vordefinierte Systems Manager-Dokument (SSM-Dokument) verwenden, `AWS-RunRemoteScript` um Skripte von herunterzuladenGitHub, einschließlich Ansible Playbooks, Python, Ruby und PowerShell Skripts. Durch die Verwendung dieses SSM-Dokuments müssen Sie Skripts nicht mehr manuell in Amazon Elastic Compute Cloud (Amazon EC2) portieren oder in SSM-Dokumente einbinden. Die - AWS Systems Manager Integration mit GitHub fördert Infrastruktur als Code , wodurch die Zeit für die Verwaltung von Knoten reduziert und gleichzeitig Konfigurationen in Ihrer gesamten Flotte standardisiert werden.

Sie können auch benutzerdefinierte Systems Manager-Dokumente erstellen, mit denen Sie Skripts oder andere Systems Manager-Dokumente von Remote-Speicherorten herunterladen und ausführen können. Weitere Informationen finden Sie unter Erstellen von zusammengesetzten Dokumenten.

Sie können auch ein Verzeichnis mit mehreren Skripts herunterladen. Wenn Sie das primäre Skript im Verzeichnis ausführen, führt Systems Manager auch alle referenzierten Skripts aus, die im Verzeichnis enthalten sind.

Beachten Sie die folgenden wichtigen Hinweise zum Ausführen von Skripts von GitHub.

- Systems Manager prüft nicht, ob Ihr Skript auf einem Knoten ausgeführt werden kann. Stellen Sie sicher, dass die erforderliche Software auf dem Knoten installiert ist, bevor Sie das Skript herunterladen und ausführen. Alternativ können Sie ein zusammengesetztes Dokument erstellen, das die Software über Run Command oder State Manager, eine Funktion von AWS Systems Manager, installiert und das Skript anschließend herunterlädt und ausführt.
- Sie sind dafür verantwortlich, sicherzustellen, dass alle GitHub Anforderungen erfüllt werden. Dies umfasst die Aktualisierung Ihres Zugriffstokens, wenn erforderlich. Stellen Sie sicher, dass Sie die Anzahl an authentifizierten oder nicht authentifizierten Anfragen nicht überschreiten. Weitere Informationen finden Sie in der Dokumentation zu GitHub.
- GitHub Enterprise -Repositorys werden nicht unterstützt.

Themen

- AnsiblePlaybooks ausführen von GitHub
- Führen Sie Python-Skripte aus GitHub

AnsiblePlaybooks ausführen von GitHub

Dieser Abschnitt enthält Verfahren, mit denen Sie Ansible Playbooks entweder mit der Konsole oder mit der AWS Command Line Interface ()AWS CLI ausführen können. GitHub

Bevor Sie beginnen

Wenn Sie beabsichtigen, ein in einem privaten GitHub Repository gespeichertes Skript auszuführen, erstellen Sie einen AWS Systems Manager `SecureString` Parameter für Ihr GitHub Sicherheitszugriffstoken. Sie können nicht auf ein Skript in einem privaten GitHub Repository zugreifen, indem Sie Ihr Token manuell über SSH übergeben. Das Zugriffstoken muss als `SecureString`-Systems Manager-Parameter übertragen werden. Weitere Informationen zum Erstellen eines `SecureString`-Parameters finden Sie unter Erstellen von Systems Manager-Parametern.

Führen Sie ein Ansible Playbook von GitHub (Konsole) aus

Führen Sie ein Ansible Playbook aus GitHub

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option **`AWS-RunRemoteScript`** aus.
5. Führen Sie unter Command parameters die folgenden Schritte aus:

   - Wählen Sie unter Quelltyp die Option aus GitHub.
   - Geben Sie im Feld Source Info die für den Zugriff auf die Quelle erforderlichen Informationen im folgenden Format an.

   ```
   {
     "owner": "owner_name",
     "repository": "repository_name",
     "getOptions": "branch:branch_name",
     "path": "path_to_scripts_or_directory",
     "tokenInfo": "{{ssm-secure:SecureString_parameter_name}}"
   }
   ```

   Dieses Beispiel lädt eine Datei mit dem Namen `webserver.yml` herunter.

   ```
   {
       "owner": "TestUser1",
       "repository": "GitHubPrivateTest",
       "getOptions": "branch:myBranch",
       "path": "scripts/webserver.yml",
       "tokenInfo": "{{ssm-secure:mySecureStringParameter}}"
   }
   ```

   > **Note**
   >
   > `"branch"` ist nur erforderlich, wenn Ihr SSM-Dokument in einer anderen Verzweigung als `master` gespeichert ist.

Um die Version Ihrer Skripts zu verwenden, die sich in einem bestimmten Commit in Ihrem Repository befinden, verwenden Sie `commitID` mit `getOptions`statt `branch`. Zum Beispiel:
`"getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",`

- Geben Sie im Feld Command Line Parameter für die Skriptausführung ein. Ein Beispiel.

  `ansible-playbook -i “localhost,” --check -c local webserver.yml`

- (Optional) Geben Sie im Feld Working Directory (Arbeitsverzeichnis) den Namen eines Verzeichnisses auf dem Knoten ein, auf dem das Skript heruntergeladen und ausgeführt werden soll.
- (Optional) Geben Sie unter Execution Timeout die Dauer in Sekunden an, bis das System die Skriptbefehlausführung fehlschlagen lässt.

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   Tip

   Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Other parameters (Weitere Parameter):

   - Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Rate control (Ratenregelung):

   - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     Note

     Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und

Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

10. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Wählen Sie Ausführen aus.

Führen Sie ein Ansible Playbook aus, GitHub indem Sie AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um ein Skript von herunterzuladen und auszuführenGitHub.

```
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"]}'
```

   Hier ist ein Beispielbefehl für die Ausführung auf einem lokalen Linux-Computer.

```
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"]}'
```

Führen Sie Python-Skripte aus GitHub

Dieser Abschnitt enthält Verfahren, mit denen Sie Python-Skripte entweder über die AWS Systems Manager Konsole oder die AWS Command Line Interface (AWS CLI) ausführen können. GitHub

Führen Sie ein Python-Skript von GitHub (Konsole) aus

Führen Sie ein Python-Skript aus GitHub

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option **`AWS-RunRemoteScript`** aus.
5. Führen Sie unter Command parameters (Befehlsparameter) die folgenden Schritte aus:

   - Wählen Sie unter Quelltyp die Option aus GitHub.
   - Geben Sie im Feld Source Info die für den Zugriff auf die Quelle erforderlichen Informationen im folgenden Format an:

   ```
   {
       "owner": "owner_name",
       "repository": "repository_name",
       "getOptions": "branch:branch_name",
       "path": "path_to_document",
       "tokenInfo": "{{ssm-secure:SecureString_parameter_name}}"
   }
   ```

   Im Folgenden wird beispielsweise ein Verzeichnis von Skripts mit dem Namen complex-script heruntergeladen.

   ```
   {
       "owner": "TestUser1",
       "repository": "SSMTestDocsRepo",
       "getOptions": "branch:myBranch",
       "path": "scripts/python/complex-script",
       "tokenInfo": "{{ssm-secure:myAccessTokenParam}}"
   }
   ```

   > **Note**
   >
   > `"branch"` ist nur erforderlich, wenn Ihre Skripts in einer anderen Verzweigung als `master` gespeichert sind.
   > Um die Version Ihrer Skripts zu verwenden, die sich in einem bestimmten Commit in Ihrem Repository befinden, verwenden Sie `commitID` mit `getOptions`statt `branch`. Zum Beispiel:
   > `"getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",`

- Geben Sie für Command Line (Befehlszeile) Parameter für die Skriptausführung ein. Ein Beispiel.

```
mainFile.py argument-1 argument-2
```

  In diesem Beispiel wird `mainFile.py` ausgeführt. Diese Datei kann anschließend andere Skripts im Verzeichnis `complex-script` ausführen.

- (Optional) Geben Sie für Working Directory (Arbeitsverzeichnis) den Namen eines Verzeichnisses auf dem Knoten ein, auf dem das Skript heruntergeladen und ausgeführt werden soll.
- (Optional) Geben Sie für Execution Timeout (Ausführungstimeout) die Dauer in Sekunden an, bis das System die Skriptbefehlausführung fehlschlagen lässt.

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Other parameters (Weitere Parameter):

- Geben Sie im Feld Comment (Kommentar) Informationen zu diesem Befehl ein.
- Geben Sie für Timeout (seconds) (Timeout (Sekunden)) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Rate control (Ratenregelung):

- Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die

Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

10. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Wählen Sie Ausführen aus.

Führen Sie ein Python-Skript aus, GitHub indem Sie den AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um ein Skript von herunterzuladen und auszuführenGitHub.

```
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"]}'
```

   Ein Beispiel.

```
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 "]}'
```

   In diesem Beispiel wird ein Verzeichnis von Skripts mit dem Namen `complex-script` heruntergeladen. Der `commandLine`-Eintrag führt `mainFile.py` aus. Diese Datei kann anschließend andere Skripts im Verzeichnis `complex-script` ausführen.

Verwenden von Chef InSpec Profilen mit Systems Manager Compliance

AWS Systems Manager integriert mit Chef InSpec. Chef InSpecist ein Open-Source-Testframework, mit dem Sie menschenlesbare Profile erstellen können, um sie in GitHub oder im Amazon Simple Storage Service (Amazon S3) zu speichern. Anschließend können Sie Systems Manager verwenden, um Compliance-Scans auszuführen und konforme und nicht konforme Knoten anzuzeigen. Ein Profil ist eine Sicherheits-, Compliance- oder Richtlinienanforderung für Ihre Datenverarbeitungsumgebung. Sie können beispielsweise Profile erstellen, die folgende Überprüfungen durchführen, wenn Sie Ihre Knoten mit Compliance, eine Funktion von AWS Systems Manager, scannen:

- Überprüfen Sie, ob bestimmte Ports geöffnet oder geschlossen sind.

- Überprüfen Sie, ob bestimmte Anwendungen ausgeführt werden.
- Überprüfen Sie, ob bestimmte Pakete installiert sind.
- Prüfen Sie die Windows-Registry-Schlüssel auf spezifische Eigenschaften.

Sie können InSpec Profile für Amazon Elastic Compute Cloud (Amazon EC2) -Instances und lokale Server oder virtuelle Maschinen (VMs) erstellen, die Sie mit Systems Manager verwalten. Das folgende Chef InSpec Beispielprofil überprüft, ob Port 22 geöffnet ist.

```
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
```

InSpec enthält eine Sammlung von Ressourcen, mit denen Sie schnell Prüfungen und Überwachungskontrollen erstellen können. InSpec verwendet die InSpec domänenspezifische Sprache (DSL) zum Schreiben dieser Steuerelemente in Ruby. Sie können auch Profile verwenden, die von einer großen Benutzergemeinschaft erstellt wurden. InSpec Das Projekt DevSec chef-os-hardening GitHub umfasst beispielsweise Dutzende von Profilen, mit denen Sie Ihre Knoten schützen können. Sie können Profile in GitHub oder Amazon S3 erstellen und speichern.

Funktionsweise

So funktioniert die Verwendung von InSpec Profilen mit Compliance:

1. Identifizieren Sie entweder vordefinierte InSpec Profile, die Sie verwenden möchten, oder erstellen Sie Ihre eigenen. Sie können vordefinierte Profile verwendenGitHub, um loszulegen. Informationen zum Erstellen eigener InSpec Profile finden Sie unter Chef InSpecChef-Profile.
2. Speichern Sie Profile entweder in einem öffentlichen oder privaten GitHub Repository oder in einem S3-Bucket.
3. Führen Sie Compliance mit Ihren InSpec Profilen mithilfe des Systems Manager Manager-Dokuments (SSM-Dokument) `AWS-RunInspecChecks` durch. Sie können einen Konformitätsscan startenRun Command, indem Sie eine Funktion von AWS Systems Manager für On-Demand-

Scans verwenden, oder Sie können regelmäßige Konformitätsscans mit State Manager der AWS Systems Manager Funktion von planen.

4. Identifizieren Sie nicht konforme Knoten, indem Sie die Compliance-API oder Compliance-Konsole verwenden.

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - Chefverwendet einen Client auf Ihren Knoten, um das Profil zu verarbeiten. Sie müssen den Client nicht installieren. Wenn Systems Manager das SSM-Dokument `AWS-RunInspecChecks` ausführt, prüft das System, ob der Client installiert ist. Andernfalls installiert Systems Manager den Chef Client während des Scans und deinstalliert den Client nach Abschluss des Scans.
> - Ausführen des SSM-Dokuments `AWS-RunInspecChecks`, weist, wie in diesem Thema beschrieben, einen Compliance-Eintrag vom Typ `Custom:Inspec` zu jedem Ziel-Knoten zu. Um diesen Konformitätstyp zuzuweisen, ruft das Dokument den Vorgang PutComplianceItems API auf.

Einen InSpec Konformitätsscan ausführen

Dieser Abschnitt enthält Informationen zum Ausführen eines InSpec Konformitätsscans mithilfe der Systems Manager Manager-Konsole und der AWS Command Line Interface (AWS CLI). In der Konsolenprozedur wird angezeigt, wie Sie State Manager konfigurieren, um den Scan auszuführen. Das AWS CLI Verfahren zeigt, wie die Konfiguration für Run Command die Ausführung des Scans konfiguriert wird.

Ausführen eines InSpec Konformitätsscans mit State Manager (Konsole)

Um einen InSpec Konformitätsscan State Manager mit der AWS Systems Manager Konsole auszuführen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie Create association (Zuordnung erstellen) aus.

4. Geben Sie im Abschnitt Provide association details (Zuordungsdetails bereitstellen) einen Namen ein.
5. Wählen Sie in der Liste Dokument die Option **`AWS-RunInspecChecks`**. aus.
6. Wählen Sie in der Liste Document version (Dokumentversion) die Option Latest at runtime (Neueste zur Laufzeit) aus.
7. Wählen Sie im Abschnitt Parameter in der Liste Quelltyp entweder GitHuboder S3 aus.

   Wenn Sie möchten GitHub, geben Sie den Pfad zu einem InSpec Profil in einem öffentlichen oder privaten GitHub Repository in das Feld Quellinformationen ein. Hier ist ein Beispielpfad zu einem öffentlichen Profil, das vom Systems Manager Manager-Team vom folgenden Ort aus bereitgestellt wurde: https://github.com/awslabs/amazon-ssm/tree/master/Compliance/InSpec/PortCheck.

   ```
   {"owner":"awslabs","repository":"amazon-ssm","path":"Compliance/InSpec/
   PortCheck","getOptions":"branch:master"}
   ```

   Wenn Sie S3 wählen, geben Sie im Feld Quellinformationen eine gültige URL zu einem InSpec Profil in einem S3-Bucket ein.

   Weitere Informationen zur Integration von Systems Manager mit GitHub Amazon S3 finden Sie unterAusführen von Skripts von GitHub.
8. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.
9. Verwenden Sie im Abschnitt Specify schedule (Zeitplan festlegen) die Zeitplan-Builder-Optionen, um einen Zeitplan für das Ausführen des Compliance-Scans zu erstellen.
10. Für Rate control (Ratenregelung):

    - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags angeben, die auf verwaltete Knoten angewendet werden, oder indem Sie AWS -Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen kann, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

11. (Optional) Wenn Sie im Abschnitt Output options (Ausgabeoptionen) die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Enable writing to a S3 bucket (Schreiben in einen S3-Bucket aktivieren). Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind diejenigen des Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (hybrid-aktivierte Maschinen), die der Instance zugewiesen sind, und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen oder Erstellen einer IAM-Dienstrolle für eine Hybridumgebung. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das Instanzprofil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet sind, über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

12. Wählen Sie Create Association. Das System erstellt die Zuordnung und führt den Compliance-Scan automatisch aus.
13. Warten Sie einige Minuten, bis der Scan abgeschlossen ist, und wählen Sie dann Compliance im Navigationsbereich aus.

14. Suchen Sie unter Corresponding managed instances (Entsprechende verwaltete Instances) die Knoten, in denen die Spalte Compliance Type (Compliance-Typ) Custom:Inspec lautet.
15. Wählen Sie eine Knoten-ID aus, um die Details von nicht konformen Status anzuzeigen.

Einen InSpec Konformitätsscan mit Run Command ()AWS CLI ausführen

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie einen der folgenden Befehle aus, um ein InSpec Profil GitHub entweder von Amazon S3 aus auszuführen.

   Der -Befehl verwendet die folgenden Parameter:

   - sourceType: GitHub oder Amazon S3
   - sourceInfo: URL zum InSpec Profilordner entweder in GitHub oder einem S3-Bucket. Der Ordner muss die InSpec Basisdatei (*.yml) und alle zugehörigen Steuerelemente (*.rb) enthalten.

   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"}"]}'
   ```

   Ein Beispiel.

   ```
   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/DOC-EXAMPLE-
BUCKET/Inspec.yml_file\"}"]}'
```

Ein Beispiel.

```
aws ssm send-command --document-name "AWS-RunInspecChecks" --targets
 '[{"Key":"tag:testEnvironment","Values":["webServers"]}]' --
parameters'{"sourceType":["S3"],"sourceInfo":["{\"path\":\"https://s3.aws-api-
domain/DOC-EXAMPLE-BUCKET/InSpec/PortCheck.yml\"}"]}'
```

3. Führen Sie den folgenden Befehl aus, um eine Übersicht des Compliance-Scans anzuzeigen.

```
aws ssm list-resource-compliance-summaries --filters
 Key=ComplianceType,Values=Custom:Inspec
```

4. Führen Sie den folgenden Befehl aus, um Details eines Knotens anzuzeigen, der nicht konform ist.

```
aws ssm list-compliance-items --resource-ids node_ID --resource-type
 ManagedInstance --filters Key=DocumentName,Values=AWS-RunInspecChecks
```

Integration mit ServiceNow

ServiceNow bietet ein cloudbasiertes Service-Management-System zum Erstellen und Verwalten von Workflows auf Organisationsebene, z. B. für IT-Services, Ticketing-Systeme und Support. Der AWS Service Management Connector ist ServiceNow in Systems Manager integriert, um Ressourcen von bereitzustellen, zu verwalten und zu betreiben AWS ServiceNow. Sie können den AWS Service Management Connector verwenden, um alle OpsCenterFunktionen von ServiceNow in AutomationChange Manager, , Incident Manager und zu integrieren AWS Systems Manager.

Sie können die folgenden Aufgaben mit ausführenServiceNow:

- Führen Sie Automatisierungs-Playbooks aus Systems Manager aus.
- Zeigen Sie Incidents über Systems Manager OpsItems an, aktualisieren diese und beheben Sie sie.

- Zeigen Sie Betriebselemente, z. B. Vorfälle, über Systems Manager OpsCenter an und verwalten Sie diese.
- Zeigen Sie aus einer kuratierten Liste vorab genehmigter Änderungsvorlagen Systems-Manager-Änderungsanforderungen an und führen Sie diese aus.
- Verwalten und beheben Sie Vorfälle, an denen AWS gehostete Anwendungen beteiligt sind, durch Integration in Incident Manager.

> **Note**
>
> Informationen zur Integration von in ServiceNowfinden Sie unter Konfigurieren von AWS Service-Integrationen im AWS Service Management Connector Administratorhandbuch.

Markieren von Systems Manager-Ressourcen

Ein Tag (Markierung) ist eine Markierung, die Sie einer AWS-Ressource zuordnen. Jedes Tag besteht aus einem Schlüssel und einem Wert, die Sie beide selbst definieren.

Mithilfe von Tags können Sie Ihre AWS-Ressourcen auf verschiedene Arten kategorisieren, z. B. nach Zweck, Besitzer oder Umgebung. Wenn Sie beispielsweise Ihre Ressourcen danach organisieren und verwalten möchten, ob sie für die Entwicklung oder Produktion verwendet werden, können Sie einige von ihnen mit dem Schlüssel `Environment` und dem Wert `Production` markieren. Anschließend können Sie verschiedene Arten von Abfragen für Ressourcen ausführen, die mit `"Key=Environment,Values=Production"` markiert sind. Sie können beispielsweise einen Satz von Tags für die verwalteten Knoten Ihres Kontos definieren, die Ihnen die Nachverfolgung oder das Anvisieren von Instances nach Betriebssystem und Umgebung ermöglichen, beispielsweise SUSE Linux Enterprise Server gruppiert als `development`, `staging` und `production`. Sie können auch Operationen für Ressourcen ausführen, indem Sie dieses Schlüssel-Wert-Paar in Ihren Befehlen angeben, beispielsweise die Ausführung eines Aktualisierungsskripts für alle Knoten in der Gruppe oder die Überprüfung des Status dieser Knoten.

Sie können die auf Ihre AWS Systems Manager-Ressourcen angewendeten Tags in verschiedenen Operationen verwenden. Sie können Operationen beispielsweise ausschließlich auf verwaltete Knoten ausrichten, die mit einem angegebenen Tag-Schlüssel-Wert-Paar markiert sind, wenn Sie einen Befehl ausführen oder einem Wartungsfenster Ziele zuweisen. Sie können auch den Zugriff auf Ihre Ressourcen einschränken, indem Sie die auf diese angewendeten Tags als Grundlage nehmen.

Darüber hinaus können Sie Ressourcengruppen erstellen, indem Sie die gleichen Tags für AWS-Ressourcen verschiedener Typen angeben, nicht nur desselben Typs. Anschließend können Sie die Resource Groups verwenden, um Informationen dazu anzuzeigen, welche Ressourcen in einer Gruppe kompatibel sind und ordnungsgemäß funktionieren und welche Ressourcen eine Aktion erfordern. Die Ihnen angezeigten Informationen beziehen sich auf alle Typen von AWS-Ressourcen, die einer Ressourcengruppe hinzugefügt werden können, nicht nur auf unterstützte Systems Manager-Ressourcentypen. Weitere Informationen finden Sie unter Was sind AWS Resource Groups? im AWS Resource Groups-Benutzerhandbuch.

Im Rest dieses Kapitels wird beschrieben, wie Tags zu Systems Manager-Ressourcen hinzugefügt und aus diesen entfernt werden.

Themen

- Systems-Manager-Ressourcen, die Sie mit Tags versehen können

- Markieren von Systems-Manager-Zuordnungen
- Markieren von Automatisierungen
- Markierungen von Systems Manager-Dokumenten
- Markieren von Wartungsfenstern
- Markieren verwalteter Knoten
- Markieren von OpsItems
- Markieren von Systems Manager-Parametern
- Markieren von Patch-Baselines

Systems-Manager-Ressourcen, die Sie mit Tags versehen können

Sie können Tags auf die folgenden AWS Systems Manager-Ressourcen anwenden:

- Zuordnungen
- Automatisierungen
- -Documents
- Wartungsfenster
- Verwaltete Knoten
- OpsItems
- OpsMetadata
- Parameter
- Patch-Baselines

Sie können jeden dieser Typen, außer OpsItems und OpsMetadata, einer Ressourcengruppe hinzufügen.

Abhängig vom Ressourcentyp können Sie Tags verwenden, um zu ermitteln, welche Ressourcen in eine Operation eingefügt werden sollten. Beispielsweise können Sie eine Gruppe verwalteter Knoten markieren und anschließend eine Wartungsfensteraufgabe ausführen, die nur auf Knoten mit diesem Schlüssel-Wert-Paar ausgerichtet ist.

Sie können den Zugriff von Benutzern auf diese Ressourcentypen auch einschränken, indem Sie AWS Identity and Access Management (IAM-)Richtlinien erstellen, die die Tags festlegen, auf die

ein Benutzer zugreifen kann, und die Richtlinie an IAM-Entitäten (Benutzer, Rollen oder Gruppen) anfügen. Im Folgenden finden Sie einige Beispiele für die Einschränkung des Ressourcenzugriffs mithilfe von Tags.

- Sie können ein Tag auf eine Reihe von benutzerdefinierten Systems-Manager-Dokumenten (SSM-Dokumente) anwenden und dann eine IAM-Richtlinie erstellen und anwenden, die Zugriff auf Dokumente mit diesem Tag gewährt, aber nicht auf andere (oder die den Zugriff auf nur diese Dokumente verbietet).
- Sie können Tags zu OpsItems zuweisen und anschließend IAM-Richtlinien erstellen, die Einschränkungen hinsichtlich der Benutzer oder Gruppen enthalten, die diese Ressourcen anzeigen oder aktualisieren können. Organisationsdirektoren könnte beispielsweise ein vollständiger Zugriff auf alle OpsItems gewährt werden, während Softwareentwickler und Supportingenieure nur auf die Projekte oder Clientsegmente zugreifen können, für die sie verantwortlich sind.
- Sie können ein gemeinsames Tag auf Ressourcen aller sechs unterstützten Typen anwenden und eine IAM-Richtlinie erstellen, die ausschließlich Zugriff auf diese Ressourcen gewährt, z. B. `Key=Project,Value=ProjectA` oder `Key=Environment,Value=Development`. Sie können sogar ausschließlich Zugriff auf Ressourcen gewähren, denen beide Tag-Paare zugewiesen wurden. Dadurch ist es möglich, beispielsweise Benutzer auf die Arbeit ausschließlich mit Ressourcen für ProjectA in der Entwicklungsumgebung einzuschränken.

Sie können die Systems Manager Resource Groups-Konsole, die Konsole für die unterstützten Ressourcentypen (z. B. die Maintenance Windows-Konsole oder OpsCenter-Konsole), die AWS Command Line Interface (AWS CLI) und die AWS Tools for PowerShell nutzen. Sie können Tags hinzufügen, wenn Sie eine Ressource erstellen oder aktualisieren. Beispielsweise können Sie den AWS CLI-Befehl add-tags-to-resource verwenden, um Tags zu den unterstützten Systems-Manager-Ressourcentypen hinzuzufügen, nachdem sie erstellt wurden. Sie können den Befehl remove-tags-from-resource verwenden, um sie zu entfernen.

Markieren von Systems-Manager-Zuordnungen

In den Themen in diesem Abschnitt wird beschrieben, wie Sie mit Tags für State Manager-Zuordnungen arbeiten. State Manager ist eine Komponente von AWS Systems Manager.

Themen

- Erstellen von Zuordnungen mit Tags

- Hinzufügen von Tags zu einer vorhandenen Zuordnung
- Entfernen von Tags aus einer Zuordnung

Erstellen von Zuordnungen mit Tags

Sie können einer State Manager-Zuordnung Tags hinzufügen, wenn Sie sie mit der AWS CLI erstellen. Das Hinzufügen von Tags zu einer Zuordnung, wenn Sie sie mithilfe der Systems-Manager-Konsole erstellen, wird nicht unterstützt. Weitere Informationen finden Sie unter Erstellen einer Zuordnung (Befehlszeile).

Hinzufügen von Tags zu einer vorhandenen Zuordnung

Verwenden Sie die folgenden Verfahren, um einer vorhandenen State Manager-Zuordnung über die Befehlszeile Tags hinzuzufügen.

Themen

- Hinzufügen von Tags zu einer vorhandenen Zuordnung (AWS CLI)
- Hinzufügen von Tags zu einer vorhandenen Zuordnung (AWS Tools for PowerShell)

Hinzufügen von Tags zu einer vorhandenen Zuordnung (AWS CLI)

1. Führen Sie mit der AWS CLI den folgenden Befehl aus, um Zuordnungen aufzulisten, die Sie mit Tags versehen können.

   ```
   aws ssm list-associations
   ```

   Notieren Sie den Namen einer Zuordnung, die Sie markieren möchten.

2. Führen Sie den folgenden Befehl aus, um eine Zuordnung zu markieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   ```
   aws ssm add-tags-to-resource \
       --resource-type "Association" \
       --resource-id "association-ID" \
       --tags "Key=tag-key,Value=tag-value"
   ```

   Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Zuordnungs-Tags zu verifizieren.

```
aws ssm list-tags-for-resource --resource-type "Association" --resource-id
 "association-ID"
```

Hinzufügen von Tags zu einer vorhandenen Zuordnung (AWS Tools for PowerShell)

1. Führen Sie den folgenden Befehl aus, um Zuordnungen aufzulisten, die Sie mit Tags versehen können.

```
Get-SSMAssociationList
```

2. Führen Sie die folgenden Befehle aus, um einen Parameter zu taggen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
$tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
```

```
$tag.Key = "tag-key"
```

```
$tag.Value = "tag-value"
```

```
Add-SSMResourceTag `
    -ResourceType "Association" `
    -ResourceId "association-ID" `
    -Tag $tag `
    -Force
```

3. Führen Sie den folgenden Befehl aus, um die Zuordnungs-Tags zu verifizieren.

```
Get-SSMResourceTag `
    -ResourceType "Association" `
    -ResourceId "association-ID"
```

Entfernen von Tags aus einer Zuordnung

Sie können die Befehlszeile verwenden, um Tags aus einer State Manager-Zuordnung zu entfernen.

Entfernen von Tags aus einer Zuordnung (Befehlszeile)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die Zuordnungen in Ihrem Konto aufzulisten.

   Linux & macOS

   ```
   aws ssm list-associations
   ```

   Windows

   ```
   aws ssm list-associations
   ```

   PowerShell

   ```
   Get-SSMAssociationList
   ```

   Notieren Sie den Namen einer Zuordnungen, aus der Sie Tags entfernen möchten.

2. Führen Sie den folgenden Befehl aus, um Tags aus einer Zuordnung zu entfernen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm remove-tags-from-resource \
       --resource-type "Association" \
       --resource-id "association-ID" \
       --tag-key "tag-key"
   ```

   Windows

   ```
   aws ssm remove-tags-from-resource ^
       --resource-type "Association" ^
       --resource-id "association-ID" ^
       --tag-key "tag-key"
   ```

   PowerShell

   ```
   Remove-SSMResourceTag
   ```

```
    -ResourceId "association-ID"
    -ResourceType "Association"
    -TagKey "tag-key
```

Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Zuordnungs-Tags zu verifizieren.

   Linux & macOS

   ```
   aws ssm list-tags-for-resource \
       --resource-type "Association" \
       --resource-id "association-ID"
   ```

   Windows

   ```
   aws ssm list-tags-for-resource ^
       --resource-type "Association" ^
       --resource-id "association-ID"
   ```

   PowerShell

   ```
   Get-SSMResourceTag `
       -ResourceType "Association" `
       -ResourceId "association-ID"
   ```

Markieren von Automatisierungen

In den Themen in diesem Abschnitt wird beschrieben, wie Sie mit Tags für Automatisierungen arbeiten. Sie können AWS Systems Manager Automatisierungen maximal fünf Tags hinzufügen. Sie können Tags zu Automatisierungen hinzufügen, wenn Sie sie über die Konsole oder die Befehlszeile starten, oder nachdem sie über die Befehlszeile ausgeführt wurden.

Hinzufügen von Tags zu Automatisierungen (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.

3. Wählen Sie das Automation-Runbook aus, das Sie ausführen möchten.
4. Wählen Sie Execute automation (Automatisierung ausführen).
5. Wählen Sie im Abschnitt Tags (Tags) die Option Edit (Bearbeiten) aus. Fügen Sie anschließend ein oder mehrere Schlüssel-Wert-Tag-Paare hinzu.
6. Wählen Sie Speichern.

Hinzufügen von Tags zu Automatisierungen (Befehlszeile)

Führen Sie über Ihre bevorzugten Befehlszeilen-Tools den folgenden Befehl aus, um Tags zu einer Automatisierung hinzuzufügen, wenn sie gestartet wird. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name DocumentName \
    --parameters ParametersRequiredByDocument \
    --tags "Key=ExampleKey,Value=ExampleValue"
```

Windows

```
aws ssm start-automation-execution ^
    --document-name DocumentName ^
    --parameters ParametersRequiredByDocument ^
    --tags "Key=ExampleKey,Value=ExampleValue"
```

PowerShell

```
$exampleTag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
$exampleTag.Key = "ExampleKey"
$exampleTag.Value = "ExampleValue"

Start-SSMAutomationExecution `
    -DocumentName DocumentName `
    -Parameter ParametersRequiredByDocument
    -Tag $exampleTag
```

1. Sie können Automatisierungen nach der Ausführung auch markieren, indem Sie Ihr bevorzugtes Befehlszeilen-Tool verwenden. Führen Sie den folgenden Befehl aus, um einer Automatisierung Tags hinzuzufügen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm add-tags-to-resource \
       --resource-type "Automation" \
       --resource-id "automation-execution-id" \
       --tags "Key=ExampleKey,Value=ExampleValue"
   ```

   Windows

   ```
   aws ssm add-tags-to-resource ^
       --resource-type "Automation" ^
       --resource-id "automation-execution-id" ^
       --tags "Key=ExampleKey,Value=ExampleValue"
   ```

   PowerShell

   ```
   $exampleTag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
   $exampleTag.Key = "ExampleKey"
   $exampleTag.Value = "ExampleValue"

   Add-SSMResourceTag `
       -ResourceType "Automation" `
       -ResourceId "automation-execution-id" `
       -Tag $exampleTag `
       -Force
   ```

   Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

2. Führen Sie den folgenden Befehl aus, um die Tags der Automatisierung zu verifizieren.

   Linux & macOS

   ```
   aws ssm list-tags-for-resource \
       --resource-type "Automation" \
       --resource-id "automation-execution-id"
   ```

Windows

```
aws ssm list-tags-for-resource ^
    --resource-type "Automation" ^
    --resource-id "automation-execution-id"
```

PowerShell

```
Get-SSMResourceTag `
    -ResourceType "Automation" `
    -ResourceId "automation-execution-id"
```

Entfernen von Tags aus Automatisierungen

Sie können ein Befehlszeilen-Tool verwenden, um Tags aus einer Automatisierung zu entfernen.

Entfernen von Tags aus Automatisierungen (Befehlszeile)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um einen Tag aus einer Automatisierung zu entfernen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm remove-tags-from-resource \
       --resource-type "Automation" \
       --resource-id "automation-execution-id" \
       --tag-key "tag-key"
   ```

   Windows

   ```
   aws ssm remove-tags-from-resource ^
       --resource-type "Automation" ^
       --resource-id "automation-execution-id" ^
       --tag-key "tag-key"
   ```

   PowerShell

   ```
   Remove-SSMResourceTag `
   ```

```
    -ResourceId "automation-execution-id" `
    -ResourceType "Automation" `
    -TagKey "tag-key" `
    -Force
```

2. Führen Sie den folgenden Befehl aus, um die Tags der Automatisierung zu verifizieren.

   Linux & macOS

   ```
   aws ssm list-tags-for-resource \
       --resource-type "Automation" \
       --resource-id "automation-execution-id"
   ```

   Windows

   ```
   aws ssm list-tags-for-resource ^
       --resource-type "Automation" ^
       --resource-id "automation-execution-id"
   ```

   PowerShell

   ```
   Get-SSMResourceTag `
       -ResourceType "Automation" `
       -ResourceId "automation-execution-id"
   ```

Markierungen von Systems Manager-Dokumenten

In den Themen in diesem Abschnitt wird beschrieben, wie Sie mit Tags für Systems Manager-Dokumente (SSM-Dokumente) arbeiten.

Themen

- Erstellen von Dokumenten mit Tags
- Hinzufügen von Tags zu vorhandenen Dokumenten
- Entfernen von Tags aus SSM-Dokumenten

Erstellen von Dokumenten mit Tags

Sie können Tags zu benutzerdefinierten SSM-Dokumenten an dem Zeitpunkt hinzufügen, an dem Sie diese erstellen.

Weitere Informationen finden Sie unter den folgenden Themen:

- Erstellen eines SSM-Dokuments (Konsole)
- Erstellen eines SSM-Dokuments (Befehlszeile)

Hinzufügen von Tags zu vorhandenen Dokumenten

Sie können Tags zu benutzerdefinierten SSM-Dokumenten hinzufügen, die Sie besitzen. Hierzu verwenden Sie die Systems Manager-Konsole oder die Befehlszeile.

Themen

- Hinzufügen von Tags zu vorhandenen SSM-Dokumenten (Konsole)
- Hinzufügen von Tags zu vorhandenen SSM-Dokumenten (Befehlszeile)

Hinzufügen von Tags zu vorhandenen SSM-Dokumenten (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie die Registerkarte Owned by me (In meinem Besitz) aus.
4. Wählen Sie den Namen des Dokuments aus, dem Sie Tags hinzufügen möchten. Wählen Sie anschließend die Registerkarte Details (Details) aus.
5. Wählen Sie im Abschnitt Tags (Tags) die Option Edit (Bearbeiten) aus. Fügen Sie anschließend ein oder mehrere Schlüssel-Wert-Tag-Paare hinzu.
6. Wählen Sie Speichern.

Hinzufügen von Tags zu vorhandenen SSM-Dokumenten (Befehlszeile)

So fügen Sie Tags zu vorhandenen SSM-Dokumenten hinzu (Befehlszeile)

1. Führen Sie über das von Ihnen bevorzugte Befehlszeilen-Tool den folgenden Befehl aus, um die Liste der Dokumente anzuzeigen, die Sie markieren können.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

   Windows

   ```
   aws ssm list-documents
   ```

   PowerShell

   ```
   Get-SSMDocumentList
   ```

   Merken Sie sich die Namen der zu markierenden Dokumente.

2. Führen Sie den folgenden Befehl aus, um ein Dokument zu markieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm add-tags-to-resource \
       --resource-type "Document" \
       --resource-id "document-name" \
       --tags "Key=tag-key,Value=tag-value"
   ```

   Windows

   ```
   aws ssm add-tags-to-resource ^
       --resource-type "Document" ^
       --resource-id "document-name" ^
       --tags "Key=tag-key,Value=tag-value"
   ```

PowerShell

```
$tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
```

```
$tag.Key = "tag-key"
```

```
$tag.Value = "tag-value"
```

```
Add-SSMResourceTag `
    -ResourceType "Document" `
    -ResourceId "document-name" `
    -Tag $tag `
    -Force
```

Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Tags von Dokumenten überprüfen.

Linux & macOS

```
aws ssm list-tags-for-resource \
    --resource-type "Document" \
    --resource-id "document-name"
```

Windows

```
aws ssm list-tags-for-resource ^
    --resource-type "Document" ^
    --resource-id "document-name"
```

PowerShell

```
Get-SSMResourceTag `
    -ResourceType "Document" `
    -ResourceId "document-name"
```

Entfernen von Tags aus SSM-Dokumenten

Sie können die Systems Manager-Konsole oder die Befehlszeile verwenden, um Tags aus SSM-Dokumenten zu entfernen.

Themen

- Entfernen von Tags aus SSM-Dokumenten (Konsole)
- Entfernen von Tags aus SSM-Dokumenten (Befehlszeile)

Entfernen von Tags aus SSM-Dokumenten (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Documents (Dokumente) aus.
3. Wählen Sie die Registerkarte Owned by me (In meinem Besitz) aus.
4. Wählen Sie den Namen des Dokuments aus, aus dem Tags entfernt werden sollen. Wählen Sie anschließend die Registerkarte Details (Details) aus.
5. Wählen Sie im Abschnitt Tags (Tags) die Option Edit (Bearbeiten) aus. Wählen Sie anschließend neben dem von Ihnen nicht mehr benötigten Tag-Paar Remove (Entfernen) aus.
6. Wählen Sie Speichern.

Entfernen von Tags aus SSM-Dokumenten (Befehlszeile)

1. Führen Sie über das von Ihnen bevorzugte Befehlszeilen-Tool den folgenden Befehl aus, um die Dokumente in Ihrem Konto aufzulisten.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

   Windows

   ```
   aws ssm list-documents
   ```

PowerShell

```
Get-SSMDocumentList
```

Notieren Sie den Namen eines Dokuments, aus dem Sie Tags entfernen möchten.

2. Führen Sie den folgenden Befehl aus, um Tags aus einem Dokument zu entfernen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm remove-tags-from-resource \
       --resource-type "Document" \
       --resource-id "document-name" \
       --tag-key "tag-key"
   ```

   Windows

   ```
   aws ssm remove-tags-from-resource ^
       --resource-type "Document" ^
       --resource-id "document-name" ^
       --tag-key "tag-key"
   ```

   PowerShell

   ```
   Remove-SSMResourceTag `
       -ResourceId "document-name" `
       -ResourceType "Document" `
       -TagKey "tag-key" `
       -Force
   ```

   Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Tags von Dokumenten überprüfen.

   Linux & macOS

   ```
   aws ssm list-tags-for-resource \
       --resource-type "Document" \
   ```

```
    --resource-id "document-name"
```

Windows

```
aws ssm list-tags-for-resource ^
    --resource-type "Document" ^
    --resource-id "document-name"
```

PowerShell

```
Get-SSMResourceTag `
    -ResourceType "Document" `
    -ResourceId "document-name"
```

Markieren von Wartungsfenstern

In den Themen in diesem Abschnitt wird beschrieben, wie Sie mit Tags für Wartungsfenster arbeiten.

Themen

- Erstellen von Wartungsfenstern mit Tags
- Hinzufügen von Tags zu vorhandenen Wartungsfenstern
- Entfernen von Tags aus Wartungsfenstern

Erstellen von Wartungsfenstern mit Tags

Sie können Wartungsfenstern zum Zeitpunkt der Erstellung Tags hinzufügen.

Weitere Informationen finden Sie unter den folgenden Themen:

- Erstellen eines Wartungsfensters (Konsole)
- Tutorial: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI)

Hinzufügen von Tags zu vorhandenen Wartungsfenstern

Sie können über die AWS Systems Manager -Konsole oder die Befehlszeile Tags zu Wartungsfenstern hinzufügen, die Sie besitzen.

Themen

- Hinzufügen von Tags zu vorhandenen Wartungsfenstern (Konsole)
- Hinzufügen von Tags zu vorhandenen Wartungsfenstern (AWS CLI)
- Markieren Sie ein Wartungsfenster (AWS Tools for PowerShell).

Hinzufügen von Tags zu vorhandenen Wartungsfenstern (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie den Namen eines bereits von Ihnen erstellten Wartungsfensters aus. Wählen Sie anschließend die Registerkarte Tags (Tags) aus.
4. Wählen Sie Edit Tags (Tags bearbeiten) und anschließend Add Tag (Tag hinzufügen) aus.
5. Geben Sie in Key (Schlüssel) einen Schlüssel für das Tag ein, z. B. **`Environment`**.
6. Geben Sie in Value (Wert) einen Wert für das Tag ein, z. B. **`Test`**.
7. Wählen Sie Änderungen speichern aus.

Hinzufügen von Tags zu vorhandenen Wartungsfenstern (AWS CLI)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die Liste der Wartungsfenster anzuzeigen, die Sie markieren können.

   ```
   aws ssm describe-maintenance-windows
   ```

   Notieren Sie die ID eines Wartungsfensters, das Sie markieren möchten.

2. Führen Sie den folgenden Befehl aus, um ein Wartungsfenster zu markieren. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm add-tags-to-resource \
       --resource-type "MaintenanceWindow" \
       --resource-id "window-id" \
       --tags "Key=tag-key,Value=tag-value"
   ```

Windows

```
aws ssm add-tags-to-resource ^
    --resource-type "MaintenanceWindow" ^
    --resource-id "window-id" ^
    --tags "Key=tag-key,Value=tag-value"
```

Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Tags für das Wartungsfenster zu verifizieren.

Linux & macOS

```
aws ssm list-tags-for-resource \
    --resource-type "MaintenanceWindow" \
    --resource-id "window-id"
```

Windows

```
aws ssm list-tags-for-resource ^
    --resource-type "MaintenanceWindow" ^
    --resource-id "window-id"
```

Markieren Sie ein Wartungsfenster (AWS Tools for PowerShell).

1. Führen Sie den folgenden Befehl aus, um Wartungsfenster aufzulisten, die Sie markieren können.

```
Get-SSMMaintenanceWindow
```

2. Führen Sie die folgenden Befehle aus, um ein Wartungsfenster zu markieren.

```
$tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
```

```
$tag.Key = "tag-key"
```

```
$tag.Value = "tag-value"
```

```
Add-SSMResourceTag `
    -ResourceType "MaintenanceWindow" `
    -ResourceId "window-id" `
    -Tag $tag
```

window-id ist die ID des Wartungsfensters, das Sie markieren möchten.

tag-key ist der Name eines von Ihnen angegebenen benutzerdefinierten Schlüssels. Dies kann z. B. Umgebung oder Projekt sein.

tag-value ist der benutzerdefinierte Inhalt für den Wert, den Sie für diesen Schlüssel angeben möchten. Dies kann z. B. Produktion oder Q321 sein.

3. Führen Sie den folgenden Befehl aus, um die Tags für das Wartungsfenster zu verifizieren.

```
Get-SSMResourceTag `
    -ResourceType "MaintenanceWindow" `
    -ResourceId "window-id"
```

Entfernen von Tags aus Wartungsfenstern

Sie können die Systems Manager-Konsole oder die Befehlszeile verwenden, um Tags aus Wartungsfenstern zu entfernen.

Themen

- Entfernen von Tags aus Wartungsfenstern (Konsole)
- Entfernen von Tags aus Wartungsfenstern (Befehlszeile)

Entfernen von Tags aus Wartungsfenstern (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie den Namen des Wartungsfensters aus, aus dem Tags entfernt werden sollen. Wählen Sie anschließend die Registerkarte Tags (Tags) aus.

4. Wählen Sie Edit tags (Tags bearbeiten) und anschließend Remove tag (Tag entfernen) neben dem nicht mehr von Ihnen benötigten Tag-Paar aus.
5. Wählen Sie Änderungen speichern aus.

Entfernen von Tags aus Wartungsfenstern (Befehlszeile)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die Wartungsfenster in Ihrem Konto aufzulisten.

   Linux & macOS

   ```
   aws ssm describe-maintenance-windows
   ```

   Windows

   ```
   aws ssm describe-maintenance-windows
   ```

   PowerShell

   ```
   Get-SSMMaintenanceWindows
   ```

   Notieren Sie die ID eines Wartungsfensters, aus dem Sie Tags entfernen möchten.
2. Führen Sie den folgenden Befehl aus, um Tags aus einem Wartungsfenster zu entfernen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm remove-tags-from-resource \
       --resource-type "MaintenanceWindow" \
       --resource-id "window-id" \
       --tag-key "tag-key"
   ```

   Windows

   ```
   aws ssm remove-tags-from-resource ^
       --resource-type "MaintenanceWindow" ^
   ```

```
    --resource-id "window-id" ^
    --tag-key "tag-key"
```

PowerShell

```
Remove-SSMResourceTag `
    -ResourceType "MaintenanceWindow" `
    -ResourceId "window-id" `
    -TagKey "tag-key
```

Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Tags für das Wartungsfenster zu verifizieren.

Linux & macOS

```
aws ssm list-tags-for-resource \
    --resource-type "MaintenanceWindow" \
    --resource-id "window-id"
```

Windows

```
aws ssm list-tags-for-resource ^
    --resource-type "MaintenanceWindow" ^
    --resource-id "window-id"
```

PowerShell

```
Get-SSMResourceTag `
    -ResourceType "MaintenanceWindow" `
    -ResourceId "window-id"
```

Markieren verwalteter Knoten

In den Themen in diesem Abschnitt wird beschrieben, wie Sie mit Tags für verwaltete Knoten arbeiten.

Ein verwalteter Knoten ist jede Maschine, für die konfiguriert ist AWS Systems Manager. Dazu gehören Amazon Elastic Compute Cloud (Amazon EC2)-Instances sowie Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung, die für Systems Manager konfiguriert sind.

Die Anweisungen in diesem Thema gelten für alle Computer, die über Systems Manager verwaltet werden.

Themen

- Erstellen oder Aktivieren verwalteter Knoten mit Tags
- Hinzufügen von Tags zu vorhandenen verwalteten Knoten
- Entfernen von Tags aus verwalteten Knoten

Erstellen oder Aktivieren verwalteter Knoten mit Tags

Sie können EC2-Instances zum Zeitpunkt der Erstellung Tags hinzufügen. Sie können On-Premise-Servern und virtuellen Maschinen (VMs) zum Zeitpunkt ihrer Aktivierung Tags hinzufügen.

Weitere Informationen finden Sie unter den folgenden Themen:

- Informationen zu EC2-Instances finden Sie unter Taggen Ihrer Amazon EC2 EC2-Ressourcen im Amazon EC2 EC2-Benutzerhandbuch. (Inhalt gilt sowohl für EC2-Instances auf Linux als auch auf Windows)
- Informationen zu lokalen Servern und VMs finden Sie unter Erstellen einer Hybridaktivierung zur Registrierung von Knoten bei Systems Manager.

Hinzufügen von Tags zu vorhandenen verwalteten Knoten

Sie können verwalteten Knoten über die Systems-Manager-Konsole oder über die Befehlszeile Tags hinzufügen.

Themen

- Hinzufügen von Tags zu einem vorhandenen verwalteten Knoten (Konsole)
- Hinzufügen von Tags zu einem vorhandenen verwalteten Knoten (Befehlszeile)

Hinzufügen von Tags zu einem vorhandenen verwalteten Knoten (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die ID des verwalteten Knotens aus, dem Sie Tags hinzufügen möchten. Wählen Sie anschließend die Registerkarte Tags.

   > **Note**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

4. Wählen Sie im Abschnitt Tags (Tags) die Option Edit (Bearbeiten) aus. Fügen Sie anschließend ein oder mehrere Schlüssel-Wert-Tag-Paare hinzu.
5. Wählen Sie Speichern.

Hinzufügen von Tags zu einem vorhandenen verwalteten Knoten (Befehlszeile)

So fügen Sie einem vorhandenen verwalteten Knoten Tags hinzu (Befehlszeile)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die Liste der verwalteten Knoten anzuzeigen, die Sie markieren können.

   Linux & macOS

   ```
   aws ssm describe-instance-information
   ```

   Windows

   ```
   aws ssm describe-instance-information
   ```

   PowerShell

   ```
   Get-SSMInstanceInformation
   ```

Notieren Sie die ID des verwalteten Knotens, den Sie markieren möchten.

> **Note**
>
> Nicht-EC2-Maschinen, die für die Verwendung mit Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung registriert wurden, beginnen mit `mi-`, z. B. `mi-0471e04240EXAMPLE`. Die IDs von EC2-Instances beginnt mit `i-`, z. B. `i-02573cafcfEXAMPLE`.

2. Führen Sie den folgenden Befehl aus, um einen verwalteten Knoten zu markieren. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm add-tags-to-resource \
       --resource-type "ManagedInstance" \
       --resource-id "instance-id" \
       --tags Key=tag-key,Value=tag-value
   ```

   Windows

   ```
   aws ssm add-tags-to-resource ^
       --resource-type "ManagedInstance" ^
       --resource-id "instance-id" ^
       --tags "Key=tag-key,Value=tag-value"
   ```

   PowerShell

   ```
   $tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
   ```

   ```
   $tag.Key = "tag-key"
   ```

   ```
   $tag.Value = "tag-value"
   ```

   ```
   Add-SSMResourceTag `
       -ResourceType "ManagedInstance" `
       -ResourceId "instance-id" `
   ```

```
    -Tag $tag `
    -Force
```

Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Tags des verwalteten Knotens zu verifizieren.

Linux & macOS

```
aws ssm list-tags-for-resource \
    --resource-type "ManagedInstance" \
    --resource-id "instance-id"
```

Windows

```
aws ssm list-tags-for-resource ^
    --resource-type "ManagedInstance" ^
    --resource-id "instance-id"
```

PowerShell

```
Get-SSMResourceTag `
    -ResourceType "ManagedInstance" `
    -ResourceId "instance-id"
```

Entfernen von Tags aus verwalteten Knoten

Sie können die Systems-Manager-Konsole oder die Befehlszeile verwenden, um Tags aus verwalteten Knoten zu entfernen.

Themen

- Entfernen von Tags aus verwalteten Knoten (Konsole)
- Entfernen von Tags aus verwalteten Knoten (Befehlszeile)

Entfernen von Tags aus verwalteten Knoten (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie den Namen des verwalteten Knotens aus, aus dem Tags entfernt werden sollen. Wählen Sie anschließend die Registerkarte Tags.
4. Wählen Sie im Abschnitt Tags (Tags) die Option Edit (Bearbeiten) aus. Wählen Sie anschließend neben dem von Ihnen nicht mehr benötigten Tag-Paar Remove (Entfernen) aus.
5. Wählen Sie Speichern.

Entfernen von Tags aus verwalteten Knoten (Befehlszeile)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die verwalteten Knoten in Ihrem Konto aufzulisten.

   Linux & macOS

   ```
   aws ssm describe-instance-information
   ```

   Windows

   ```
   aws ssm describe-instance-information
   ```

   PowerShell

   ```
   Get-SSMInstanceInformation
   ```

   Notieren Sie den Namen des verwalteten Knotens, aus dem Sie Tags entfernen möchten.
2. Führen Sie den folgenden Befehl aus, um Tags aus einem verwalteten Knoten zu entfernen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm remove-tags-from-resource \
       --resource-type "ManagedInstance" \
       --resource-id "instance-id" \
       --tag-key "tag-key"
   ```

Windows

```
aws ssm remove-tags-from-resource ^
    --resource-type "ManagedInstance" ^
    --resource-id "instance-id" ^
    --tag-key "tag-key"
```

PowerShell

```
Remove-SSMResourceTag `
    -ResourceId "instance-id" `
    -ResourceType "ManagedInstance" `
    -TagKey "tag-key" `
    -Force
```

Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Tags des verwalteten Knotens zu verifizieren.

Linux & macOS

```
aws ssm list-tags-for-resource \
    --resource-type "ManagedInstance" \
    --resource-id "instance-id"
```

Windows

```
aws ssm list-tags-for-resource ^
    --resource-type "ManagedInstance" ^
    --resource-id "instance-id"
```

PowerShell

```
Get-SSMResourceTag `
    -ResourceType "ManagedInstance" `
    -ResourceId "instance-id"
```

Markieren von OpsItems

In den Themen in diesem Abschnitt wird beschrieben, wie Sie mit Tags für OpsItems arbeiten.

Themen

- Erstellen von OpsItems mit Tags
- Hinzufügen von Tags zu vorhandenen OpsItems
- Entfernen von Tags aus Systems Manager OpsItems

Erstellen von OpsItems mit Tags

Sie können benutzerdefinierten AWS Systems Manager OpsItems zum Zeitpunkt der Erstellung Tags hinzufügen, wenn Sie ein Befehlszeilen-Tool verwenden.

Informationen hierzu finden Sie im folgenden Thema:

Hinzufügen von Tags zu vorhandenen OpsItems

Sie können Tags zu OpsItems mithilfe eines Befehlszeilen-Tools hinzufügen.

Themen

- Hinzufügen von Tags zu einem vorhandenen OpsItem (Befehlszeile)

Hinzufügen von Tags zu einem vorhandenen OpsItem (Befehlszeile)

Hinzufügen von Tags zu einem vorhandenen OpsItem (Befehlszeile)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die Liste der OpsItemanzuzeigen, die Sie markieren können.

   Linux & macOS

   ```
   aws ssm describe-ops-items
   ```

   Windows

   ```
   aws ssm describe-ops-items
   ```

PowerShell

```
Get-SSMOpsItemSummary
```

Notieren Sie die ID eines OpsItem, das Sie markieren möchten.

2. Führen Sie den folgenden Befehl aus, um ein OpsItem zu markieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

Linux & macOS

```
aws ssm add-tags-to-resource \
    --resource-type "OpsItem" \
    --resource-id "ops-item-id" \
    --tags "Key=tag-key,Value=tag-value"
```

Windows

```
aws ssm add-tags-to-resource ^
    --resource-type "OpsItem" ^
    --resource-id "ops-item-id" ^
    --tags "Key=tag-key,Value=tag-value"
```

PowerShell

```
$tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
```

```
$tag.Key = "tag-key"
```

```
$tag.Value = "tag-value"
```

```
Add-SSMResourceTag `
    -ResourceType "OpsItem" `
    -ResourceId "ops-item-id" `
    -Tag $tag `
    -Force
```

Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die OpsItem-Tags zu verifizieren.

   Linux & macOS

   ```
   aws ssm list-tags-for-resource \
       --resource-type "OpsItem" \
       --resource-id "ops-item-id"
   ```

   Windows

   ```
   aws ssm list-tags-for-resource ^
       --resource-type "OpsItem" ^
       --resource-id "ops-item-id"
   ```

   PowerShell

   ```
   Get-SSMResourceTag `
       -ResourceType "OpsItem" `
       -ResourceId "ops-item-id"
   ```

Entfernen von Tags aus Systems Manager OpsItems

Sie können ein Befehlszeilen-Tool verwenden, um Tags aus Systems Manager-OpsItems zu entfernen.

Themen

- Entfernen von Tags von OpsItems (Befehlszeile)

Entfernen von Tags von OpsItems (Befehlszeile)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die OpsItems in Ihrem Konto aufzulisten.

Linux & macOS

```
aws ssm describe-ops-items
```

Windows

```
aws ssm describe-ops-items
```

PowerShell

```
Get-SSMOpsItemSummary
```

Notieren Sie den Namen eines OpsItem, aus dem Sie Tags entfernen möchten.

2. Führen Sie den folgenden Befehl aus, um Tags aus einem OpsItem zu entfernen. Ersetzen Sie jedes *Ressourcenplatzhalter-Beispiel* durch Ihre eigenen Informationen.

Linux & macOS

```
aws ssm remove-tags-from-resource \
    --resource-type "OpsItem" \
    --resource-id "ops-item-id" \
    --tag-key "tag-key"
```

Windows

```
aws ssm remove-tags-from-resource ^
    --resource-type "OpsItem" ^
    --resource-id "ops-item-id" ^
    --tag-key "tag-key"
```

PowerShell

```
Remove-SSMResourceTag `
    -ResourceId "ops-item-id" `
    -ResourceType "OpsItem" `
    -TagKey "tag-key" `
    -Force
```

Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die OpsItem-Tags zu verifizieren.

   Linux & macOS

   ```
   aws ssm list-tags-for-resource \
       --resource-type "OpsItem" \
       --resource-id "ops-item-id"
   ```

   Windows

   ```
   aws ssm list-tags-for-resource ^
       --resource-type "OpsItem" ^
       --resource-id "ops-item-id"
   ```

   PowerShell

   ```
   Get-SSMResourceTag `
       -ResourceType "OpsItem" `
       -ResourceId "ops-item-id"
   ```

Markieren von Systems Manager-Parametern

In den Themen dieses Abschnitts wird beschrieben, wie Sie mit Tags für AWS Systems Manager Parameter (SSM-Parameter) arbeiten.

Themen

- Erstellen von Parametern mit Tags
- Hinzufügen von Tags zu vorhandenen Parametern
- Entfernen von Tags aus SSM-Parametern

Erstellen von Parametern mit Tags

Sie können SSM-Parameter zum Zeitpunkt der Erstellung Tags hinzufügen.

Weitere Informationen finden Sie unter den folgenden Themen:

- Erstellen eines Systems Manager-Parameters (Konsole)
- Erstellen eines Systems Manager-Parameters (AWS CLI)
- Erstellen eines Systems Manager-Parameters (Tools for Windows PowerShell)

Hinzufügen von Tags zu vorhandenen Parametern

Sie können Tags zu benutzerdefinierten SSM-Parametern hinzufügen, die Sie besitzen, indem Sie die Systems Manager-Konsole oder die Befehlszeile verwenden.

Themen

- Hinzufügen von Tags zu vorhandenen Parametern (Konsole)
- Hinzufügen von Tags zu vorhandenen Parametern (AWS CLI)
- Hinzufügen von Tags zu vorhandenen Parametern (AWS Tools for PowerShell)

Hinzufügen von Tags zu vorhandenen Parametern (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, den Sie bereits erstellt haben, und öffnen Sie dann die Registerkarte Tags.
4. Geben Sie im ersten Feld einen Schlüssel für das Tag ein, z. B. **`Environment`**.
5. Geben Sie im zweiten Feld einen Wert für das Tag ein, z. B. **`Test`**.
6. Wählen Sie Speichern.

Hinzufügen von Tags zu vorhandenen Parametern (AWS CLI)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die Liste der Parameter anzuzeigen, die Sie markieren können.

   ```
   aws ssm describe-parameters
   ```

   Merken Sie sich den Namen des Parameters, den Sie taggen möchten.

2. Führen Sie den folgenden Befehl aus, um einen Parameter zu taggen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws ssm add-tags-to-resource \
    --resource-type "Parameter" \
    --resource-id "parameter-name" \
    --tags "Key=tag-key,Value=tag-value"
```

   Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Tags von Parametern zu überprüfen.

```
aws ssm list-tags-for-resource --resource-type "Parameter" --resource-id
 "parameter-name"
```

Hinzufügen von Tags zu vorhandenen Parametern (AWS Tools for PowerShell)

1. Führen Sie den folgenden Befehl aus, um die Parameter aufzulisten, die Sie taggen können.

```
Get-SSMParameterList
```

2. Führen Sie die folgenden Befehle aus, um einen Parameter zu taggen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
$tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
```

```
$tag.Key = "tag-key"
```

```
$tag.Value = "tag-value"
```

```
Add-SSMResourceTag `
    -ResourceType "Parameter" `
    -ResourceId "parameter-name" `
    -Tag $tag `
    -Force
```

3. Führen Sie den folgenden Befehl aus, um die Tags von Parametern zu überprüfen.

```
Get-SSMResourceTag `
    -ResourceType "Parameter" `
    -ResourceId "parameter-name"
```

Entfernen von Tags aus SSM-Parametern

Sie können die Systems Manager-Konsole oder die Befehlszeile verwenden, um Tags aus SSM-Parametern zu entfernen.

Themen

- Entfernen von Tags aus SSM-Parametern (Konsole)
- Entfernen von Tags aus SSM-Parametern (Befehlszeile)

Entfernen von Tags aus SSM-Parametern (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen des Parameters aus, aus dem Sie Tags entfernen möchten. Wählen Sie anschließend die Registerkarte Tags (Tags) aus.
4. Wählen Sie neben dem Tag-Paar, das Sie nicht mehr benötigen, Remove (Entfernen) aus.
5. Wählen Sie Speichern.

Entfernen von Tags aus SSM-Parametern (Befehlszeile)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die Parameter in Ihrem Konto aufzulisten.

   Linux & macOS

   ```
   aws ssm describe-parameters
   ```

Windows

```
aws ssm describe-parameters
```

PowerShell

```
Get-SSMParameterList
```

Notieren Sie den Namen eines Parameters, aus dem Sie Tags entfernen möchten.

2. Führen Sie den folgenden Befehl aus, um Tags aus einem Parameter zu entfernen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm remove-tags-from-resource \
       --resource-type "Parameter" \
       --resource-id "parameter-name" \
       --tag-key "tag-key"
   ```

   Windows

   ```
   aws ssm remove-tags-from-resource ^
       --resource-type "Parameter" ^
       --resource-id "parameter-name" ^
       --tag-key "tag-key"
   ```

   PowerShell

   ```
   Remove-SSMResourceTag
       -ResourceId "parameter-name"
       -ResourceType "Parameter"
       -TagKey "tag-key
   ```

   Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Tags von Dokumenten überprüfen.

Linux & macOS

```
aws ssm list-tags-for-resource \
    --resource-type "Parameter" \
    --resource-id "parameter-name"
```

Windows

```
aws ssm list-tags-for-resource ^
    --resource-type "Parameter" ^
    --resource-id "parameter-name"
```

PowerShell

```
Get-SSMResourceTag `
    -ResourceType "Parameter" `
    -ResourceId "parameter-name"
```

Markieren von Patch-Baselines

In den Themen in diesem Abschnitt wird beschrieben, wie Sie mit Tags für Patch-Baselines arbeiten.

Themen

- Erstellen von Patch-Baselines mit Tags
- Hinzufügen von Tags zu vorhandenen Patch-Baselines
- Entfernen von Tags aus Patch-Baselines

Erstellen von Patch-Baselines mit Tags

Sie können den AWS Systems Manager Patch-Baselines bei der Erstellung Tags hinzufügen.

Weitere Informationen finden Sie unter den folgenden Themen:

- Arbeiten mit benutzerdefinierten Patch-Baselines
- Erstellen einer Patch-Baseline

- Erstellen einer Patch-Baseline mit benutzerdefinierten Repositorys für verschiedene Betriebssystemversionen

Hinzufügen von Tags zu vorhandenen Patch-Baselines

Sie können Patch-Baselines, die Sie besitzen, Tags hinzufügen, indem Sie die Systems Manager-Konsole oder die Befehlszeile verwenden.

Themen

- Hinzufügen von Tags zu vorhandenen Patch-Baselines (Konsole)
- Hinzufügen von Tags zu vorhandenen Patch-Baselines (AWS CLI)
- Markieren einer Patch-Baseline (AWS Tools for PowerShell)

Hinzufügen von Tags zu vorhandenen Patch-Baselines (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie den Namen einer benutzerdefinierten Patch-Baseline aus, die Sie bereits erstellt haben, und scrollen Sie nach unten zum Abschnitt Tags table (Tag-Tabelle). Wählen Sie anschließend Edit tags (Tags bearbeiten) aus.
4. Wählen Sie Add tag.
5. Geben Sie in Key (Schlüssel) einen Schlüssel für das Tag ein, z. B. **`Environment`**.
6. Geben Sie in Value (Wert) einen Wert für das Tag ein, z. B. **`Test`**.
7. Wählen Sie Änderungen speichern aus.

Hinzufügen von Tags zu vorhandenen Patch-Baselines (AWS CLI)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die Liste der Patch-Baselines anzuzeigen, die Sie markieren können.

   ```
   aws ssm describe-patch-baselines --filters "Key=OWNER,Values=[Self]"
   ```

   Notieren Sie die ID einer Patch-Baseline, die Sie markieren möchten.

2. Führen Sie den folgenden Befehl aus, um eine Patch-Baseline zu markieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm add-tags-to-resource \
       --resource-type "PatchBaseline" \
       --resource-id "baseline-id" \
       --tags "Key=tag-key,Value=tag-value"
   ```

   Windows

   ```
   aws ssm add-tags-to-resource ^
       --resource-type "PatchBaseline" ^
       --resource-id "baseline-id" ^
       --tags "Key=tag-key,Value=tag-value"
   ```

   Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Patch-Baseline-Tags zu verifizieren.

   Linux & macOS

   ```
   aws ssm list-tags-for-resource \
       --resource-type "PatchBaseline" \
       --resource-id "baseline-id"
   ```

   Windows

   ```
   aws ssm list-tags-for-resource ^
       --resource-type "PatchBaseline" ^
       --resource-id "patchbaseline-id"
   ```

Markieren einer Patch-Baseline (AWS Tools for PowerShell)

1. Führen Sie den folgenden Befehl aus, um die Patch-Baseline aufzulisten, die Sie markieren können.

```
Get-SSMPatchBaseline
```

2. Führen Sie die folgenden Befehle aus, um eine Patch-Baseline zu markieren. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
$tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
```

```
$tag.Key = "tag-key"
```

```
$tag.Value = "tag-value"
```

```
Add-SSMResourceTag `
    -ResourceType "PatchBaseline" `
    -ResourceId "baseline-id" `
    -Tag $tag `
    -Force
```

3. Führen Sie den folgenden Befehl aus, um die Patch-Baseline-Tags zu verifizieren.

```
Get-SSMResourceTag `
    -ResourceType "PatchBaseline" `
    -ResourceId "baseline-id"
```

Entfernen von Tags aus Patch-Baselines

Sie können die Systems Manager-Konsole oder die Befehlszeile verwenden, um Tags aus einer Patch-Baseline zu entfernen.

Themen

- Entfernen von Tags aus einer Patch-Baseline (Konsole)
- Entfernen von Tags aus Patch-Baselines (Befehlszeile)

Entfernen von Tags aus einer Patch-Baseline (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie den Namen der Patch-Baseline aus, aus der Sie Tags entfernen möchten, und scrollen Sie nach unten zum Abschnitt Tags table (Tag-Tabelle). Wählen Sie anschließend die Registerkarte Edit tags (Tags bearbeiten) aus.
4. Wählen Sie neben dem Tag-Paar, das Sie nicht mehr benötigen, Remove tag (Tag entfernen) aus.
5. Wählen Sie Änderungen speichern aus.

Entfernen von Tags aus Patch-Baselines (Befehlszeile)

1. Führen Sie über Ihr bevorzugtes Befehlszeilen-Tool den folgenden Befehl aus, um die Patch-Baselines in Ihrem Konto aufzulisten.

   Linux & macOS

   ```
   aws ssm describe-patch-baselines
   ```

   Windows

   ```
   aws ssm describe-patch-baselines
   ```

   PowerShell

   ```
   Get-SSMPatchBaseline
   ```

   Notieren Sie die ID einer Patch-Baseline, aus der Sie Tags entfernen möchten.

2. Führen Sie den folgenden Befehl aus, um Tags aus einer Patch-Baseline zu entfernen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm remove-tags-from-resource \
       --resource-type "PatchBaseline" \
       --resource-id "baseline-id" \
       --tag-key "tag-key"
   ```

Windows

```
aws ssm remove-tags-from-resource ^
    --resource-type "PatchBaseline" ^
    --resource-id "baseline-id" ^
    --tag-key "tag-key"
```

PowerShell

```
Remove-SSMResourceTag `
    -ResourceType "PatchBaseline" `
    -ResourceId "baseline-id" `
    -TagKey "tag-key
```

Wenn der Befehl erfolgreich war, wird nichts ausgegeben.

3. Führen Sie den folgenden Befehl aus, um die Patch-Baseline-Tags zu verifizieren.

Linux & macOS

```
aws ssm list-tags-for-resource \
    --resource-type "PatchBaseline" \
    --resource-id "baseline-id"
```

Windows

```
aws ssm list-tags-for-resource ^
    --resource-type "PatchBaseline" ^
    --resource-id "baseline-id"
```

PowerShell

```
Get-SSMResourceTag `
    -ResourceType "PatchBaseline" `
    -ResourceId "baseline-id"
```

AWS Systems Manager Referenz

Die folgenden Informationen und Themen können Ihnen dabei helfen, AWS Systems Manager - Lösungen besser zu implementieren.

Auftraggeber

In AWS Identity and Access Management (IAM) können Sie einem Dienst mithilfe des Principal-Richtlinienelements Zugriff auf Ressourcen gewähren oder verweigern. Der Wert des Prinzipal-Richtlinienelements für Systems Manager lautet `ssm.amazonaws.com`.

Unterstützte Geräte AWS-Regionen und Endgeräte

Siehe Service-Endpunkte von Systems Manager im Allgemeine Amazon Web Services-Referenz.

Service Quotas

Weitere Informationen finden Sie unter Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz.

API Reference

Siehe AWS Systems Manager -API-Referenz.

AWS CLI Befehlsreferenz

Siehe AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

AWS Tools for PowerShell Cmdlet-Referenz

Weitere Informationen finden Sie AWS Systems Manager im Abschnitt der AWS Tools for PowerShell Cmdlet-Referenz.

SSM AgentRepository aktiviert GitHub

Siehe aws/ amazon-ssm-agent.

Stellen Sie ein Frage

Probleme mit Systems Manager in AWS re:Post

AWS Nachrichten-Blog

Verwaltungs-Tools

Weitere Referenzthemen

- Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager
- Referenz: Cron- und Rate-Ausdrücke für System Manager
- Referenz: ec2messages, ssmmessages und andere API-Operationen
- Referenz: Erstellen formatierter Datums- und Uhrzeitzeichenfolgen für Systems Manager

Referenz: Amazon EventBridge Ereignismuster und -typen für Systems Manager

> **Note**
>
> Amazon EventBridge ist die bevorzugte Methode zum Verwalten Ihrer Ereignisse. CloudWatch Events und EventBridge sind derselbe zugrunde liegende Service und dieselbe API, aber EventBridge bietet mehr Features. Änderungen, die Sie in CloudWatch oder EventBridge vornehmen, werden in allen Konsolen widergespiegelt. Weitere Informati onen finden Sie im Benutzerhandbuch für Amazon EventBridge.

Mit Amazon EventBridge können Sie Regeln erstellen, die eingehenden Ereignissen entsprechen und sie an Ziele zur Verarbeitung weiterleiten.

Ein Ereignis weist auf eine Änderung in einer Umgebung in Ihren eigenen Anwendungen, Software as a Service (SaaS)- Anwendungen oder einem AWS-Service hin. Ereignisse werden auf bestmögliche Weise ausgegeben. Nachdem ein in einer Regel spezifizierter Ereignistyp erkannt wurde, leitet EventBridge ihn zur Verarbeitung an ein angegebenes Ziel weiter. Zu den Zielen können Amazon Elastic Compute Cloud (Amazon EC2) Instances, AWS Lambda-Funktionen, Amazon Kinesis Streams, Amazon Elastic Container Service (Amazon ECS)-Aufgaben ,AWS Step Functions-Statuscomputer, Amazon Simple Notification Service (Amazon SNS)-Themen, Amazon Simple Queue Service (Amazon SQS)-Warteschlangen, integrierte Ziele und vieles mehr gehören.

Informationen zum Erstellen von EventBridge-Regeln finden Sie in den folgenden Themen:

- Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge
- EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager

- Getting Started with Amazon EventBridge (Erste Schritte mit Amazon EventBridge) im Benutzerhandbuch zu Amazon EventBridge

Im Rest dieses Themas werden die Typen von Systems Manager-Ereignissen beschrieben, die Sie in die EventBridge Regeln einbeziehen können.

Ereignistyp: Automation

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Benachrichtigung über die Änderung des Ausführungsstatus von EC2	Der Gesamtstatus eines Automation-Workflows ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Approved • Canceled • Fehlgeschlagen • PendingApproval • PendingChangeCalendarOverride • Rejected (Abgelehnt) • Scheduled (Geplant) • Herzlichen Glückwunsch • TimedOut
Benachrichtigung über die Änderung des Automatisierungsschrittstatus in EC2	Der Status eines bestimmten Schrittes in einem Automation-Workflows ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Canceled • Fehlgeschlagen • Herzlichen Glückwunsch

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• TimedOut

Ereignistyp: Change Calendar

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Änderung des Kalenderstatus	Der Status des Change Calendar ändert sich. Sie können einer Ereignisregel eine oder beide der folgenden Statusänderungen hinzufügen: • OPEN • CLOSED Statusänderungen für Kalender, die von anderen AWS-Konten freigegeben werden, werden nicht unterstützt.

Ereignistyp: Change Manager

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Aktualisierung des Status der Änderungsanforderung	Der Status der Change Manager-Änderungsanforderung. Sie können die folgenden Status in einer Ereignisregel verwenden: • Approved • Rejected (Abgelehnt) • InProgress

Ereignistyp: Configuration Compliance

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Configuration Compliance-Statusänderung	Der Zustand eines verwalteten Knotens ändert sich je nach Zuordnungs-Compliance oder Patch-Compliance. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • compliant • non_compliant

Ereignistyp: Inventory

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Inventory Resource-Statusänderung	Das Löschen eines benutzerdefinierten Inventars und eines PutInventory-Aufrufs, der eine alte Schemaversion verwendet. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Löschereignis für benutzerdefinierten Inventartyp auf einem bestimmten Knoten. EventBridge sendet ein Ereignis pro Knoten pro benutzerdefiniertem InventoryType. • Löschereignis für benutzerdefinierten Inventartyp auf allen Knoten. • PutInventory-Aufruf mit alten Schemaver sionsereignis EventBridge sendet dieses Ereignis, wenn die Schemaversion kleiner als das aktuelle Schema ist. Dieses Ereignis gilt für alle Inventararten.

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	Weitere Informationen finden Sie unter Informationen zur EventBridge-Überwachung von Inventory-Ereignissen.

Ereignistyp: Wartungsfenster

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Benachrichtigung über die Statusänderung des Wartungsfensters	Der Gesamtstatus eines oder mehrerer Wartungsfenster ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • DISABLED • ENABLED
Benachrichtigung zur Zielregistrierung des Wartungsfensters	Der Status eines oder mehrerer Wartungsf enster ändert sich. Sie können einer Ereignisr egel eine oder mehrere der folgenden Statusänderungen hinzufügen: • DEREGISTERED • REGISTERED • UPDATED
Benachrichtigung über Ausführungsstatusä nderung des Wartungsfensters	Der Gesamtstatus eines Wartungsfensters ändert sich während der Ausführung. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • CANCELLED • CANCELLING • FEHLGESCHLAGEN

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• IN_PROGRESS • PENDING • SKIPPED_OVERLAPPING • ERFOLG • TIMED_OUT
Benachrichtigung über Aufgaben-Ausführungsstatusänderung des Wartungsfensters	Der Status einer Aufgabe in einem Wartungsfenster ändert sich während der Ausführung. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • CANCELLED • CANCELLING • FEHLGESCHLAGEN • IN_PROGRESS • ERFOLG • TIMED_OUT

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Benachrichtigung über Zielaufrufungsstatusänderung der Wartungsfenstersaufgabe	Der Status einer Wartungsfensteraufgabe für ein bestimmtes Ziel ändert sich. Diese Benachrichtigung wird nur für Run Command-Aufgaben vollständig unterstützt. Sie können bei dieser Art von Aufgabe einer Ereignisregel eine oder mehrere der folgenden Zustandsänderungen hinzufügen: • CANCELLED • CANCELLING • FEHLGESCHLAGEN • IN_PROGRESS • ERFOLG • TIMED_OUT Für Automatisierungs-, AWS Lambda-, und AWS Step Functions-Aufgaben, meldet EventBridge nur die Zustände `IN_PROGRESS` und `COMPLETE`. `COMPLETE` wird berichtet, ob die Aufgabe erfolgreich war oder nicht.
Benachrichtigung über Aufgabenregistrierung des Wartungsfenster	Der Status einer oder mehrerer Wartungsfensteraufgaben ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • DEREGISTERED • REGISTERED • UPDATED

Ereignistyp: OpsCenter

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
OpsItem erstellen	Tritt auf, wenn OpsItem erstellt wird. Sie können Regeln für einen der folgenden OpsItem-Typen hinzufügen: • /aws/issue • /aws/task • /aws/insight • /aws/actionitem
OpsItem updaten	Tritt auf, wenn OpsItem aktualisiert wird. Sie können Regeln für einen der folgenden OpsItem-Typen hinzufügen: • /aws/issue • /aws/task • /aws/insight • /aws/actionitem

Ereignistyp: Parameter Store

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Änderung des Parameterspeichers	Der Status eines Paramaters ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Erstellen • Aktualisierung • Löschen

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• LabelParameterVersion Weitere Informationen finden Sie unter Konfigurieren von EventBridge Regeln für Parameter und Parameterrichtlinien.
Parameterstore-Richtlinienaktion	Eine Bedingung für eine erweiterte Parameter richtlinienänderung ist erfüllt. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Ablauf • ExpirationNotification • NoChangeNotification Weitere Informationen finden Sie unter Konfigurieren von EventBridge Regeln für Parameter und Parameterrichtlinien.

Ereignistyp: Run Command

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Benachrichtigung über die Änderung des Befehlsaufrufstatus in EC2	Der Status eines an eine einzelne verwaltete Instance gesendeten Befehls ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Herzlichen Glückwunsch • InProgress • TimedOut

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• Canceled • Fehlgeschlagen
Benachrichtigung über die Änderung des Befehlsstatus in EC2	Der Gesamtstatus eines Befehls ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Herzlichen Glückwunsch • InProgress • TimedOut • Canceled • Fehlgeschlagen

Ereignistyp: State Manager

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Änderung des EC2 State Manager-Zuordnungs status	Die gesamte Status einer Zuordnung ändert sich, wenn sie angewendet wird. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Fehlgeschlagen • Ausstehend • Herzlichen Glückwunsch
Änderung des Zuordnungsstatus der EC2 State Manager-Instance	Der Zustand einer einzeln verwalteten Instance, auf die eine Zuordnung abzielt, ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen:

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• Fehlgeschlagen • Ausstehend • Herzlichen Glückwunsch

Referenz: Cron- und Rate-Ausdrücke für System Manager

Wenn Sie eine State Manager-Zuordnung oder ein Wartungsfenster in AWS Systems Manager erstellen, geben Sie einen Zeitplan für die Ausführung des Fensters oder der Zuordnung an. Sie können einen Zeitplan als zeitbasierten Eintrag, einen sogenannten Cron-Ausdruck, oder als häufigkeitsbasierten Eintrag, einen sogenannten Rate-Ausdruck angeben.

Allgemeine Informationen zu Cron- und Rate-Ausdrücken

Die folgenden Informationen gelten für Cron- und Rate-Ausdrücke sowohl für Wartungsfenster als auch für Zuordnungen.

Zeitpläne für Einzelläufe

Wenn Sie ein eine Zuordnung oder ein Wartungsfenster erstellen, können Sie einen Zeitstempel in koordinierter Weltzeit (Coordinated Universal Time, UTC) angeben, damit es einmalig zum angegebenen Zeitpunkt ausgeführt wird. Zum Beispiel: `"at(2020-07-07T15:55:00)"`

Offsets planen

Assoziationen und Wartungsfenster unterstützen nur für Cron-Ausdrücke zudem auch Zeitplanversätze. Ein Zeitplanversatz ist die Anzahl der Tage, die nach dem über einen CRON-Ausdruck angegebenen Datum und der angegebenen Uhrzeit gewartet werden soll, bevor die Assoziation oder das Wartungsfenster ausgeführt wird.

Maintenance window example

Im folgenden Beispiel wird mit dem CRON-Ausdruck die Ausführung eines Wartungsfensters um 23.30 Uhr am dritten Dienstag jedes Monats geplant. Wenn der Zeitplanversatz jedoch 2 lautet, wird das Wartungsfenster erst zwei Tage später um 23:30 Uhr ausgeführt.

```
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

Im folgenden Befehl plant der Cron-Ausdruck, dass eine Zuordnung am zweiten Donnerstag eines jeden Monats ausgeführt wird. Da der Zeitplanversatz jedoch 3 ist, wird die Zuordnung erst am nächsten Sonntag, also drei Tage später, ausgeführt.

```
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**
>
> Um einen Offset mit einer Zuordnung zu verwenden, müssen Sie die `--apply-only-at-cron-interval`-Option angeben. Diese Option sagt dem System eine Assoziation nicht unmittelbar nach der Erstellung auszuführen.

Wenn Sie eine Zuordnung oder ein Wartungsfenster mit einem Cron-Ausdruck erstellen, das sich auf einen im aktuellen Zeitraum bereits vergangenen Tag bezieht, jedoch ein Zeitplanversatzdatum hinzufügen, das in der Zukunft liegt, wird die Assoziation oder das Wartungsfenster in dem betreffenden Zeitraum nicht ausgeführt. Es wird im folgenden Zeitraum in Kraft treten. Wenn Sie beispielsweise einen Cron-Ausdruck angeben, der gestern ein Wartungsfenster ausgeführt hätte, und einen Zeitplanversatz von zwei Tagen hinzufügen, wird das Wartungsfenster morgen nicht ausgeführt.

Pflichtfelder

Cron-Ausdrücke für Wartungsfenster haben sechs erforderliche Felder. Cron-Ausdrücke für Zuordnungen haben fünf. (State Manager unterstützt derzeit nicht die Angabe von Monaten in Cron-Ausdrücken für Zuordnungen.) Ein zusätzliches Feld, das Feld `Seconds` (das erste in einem Cron-Ausdruck) ist optional. Die Felder werden durch Leerzeichen voneinander getrennt.

Beispiele für Cron-Ausdrücke

Minuten	Stunden	Tag des Monats	Monat	Wochentag	Jahr	Bedeutung
0	10	*	*	?	*	Ausführung jeden Tag um 10:00 Uhr (UTC)
15	12	*	*	?	*	Ausführung jeden Tag um 12:15 Uhr (UTC)
0	18	?	*	MO-FR	*	Ausführung jeden Montag bis Freitag um 18:00 Uhr (UTC)
0	8	1	*	?	*	Ausführung jeden 1. Tag des Monats um 08:00 Uhr (UTC)

Unterstützte Werte

Die folgende Tabelle zeigt die Werte, die für erforderliche Cron-Einträge unterstützt werden.

Unterstützte Werte für Cron-Ausdrücke

Feld	Werte	Platzhalter
Minuten	0-59	, - * /
Stunden	0-23	, - * /
Day-of-month	1-31	, - * ? / L W
Monat (nur Wartungsfenster)	1–12 oder JAN–DEC	, - * /
Day-of-week	1–7 oder SUN–SAT	, - * ? / L #
Jahr	1970-2199	, - * /

> **Note**
>
> Sie können keinen Wert in der day-of-month und in den day-of-week Feldern im selben Cron-Ausdruck angeben. Wenn Sie einen Wert in einem der Felder angeben, verwenden Sie ? (Fragezeichen) im anderen Feld.

Platzhalter für Cron-Ausdrücke

Die folgende Tabelle zeigt die Platzhalterwerte, die von Cron-Ausdrücken unterstützt werden.

> **Note**
>
> Cron-Ausdrücke, die zu schnelleren Häufigkeiten als fünf (5) führen, werden nicht unterstützt. Die Unterstützung für die Angabe von day-of-week sowohl einem - als auch einem day-of-month -Wert ist nicht abgeschlossen. Verwenden Sie das Fragezeichen (?) in einem dieser Felder.

Unterstützte Platzhalter für Cron-Ausdrücke

Platzhalter	Beschreibung
,	Das Platzhalterzeichen , (Komma) umfasst zusätzliche Werte. Im Feld "Monat" steht JAN, FEB, MAR für Januar, Februar und März.
-	Das Platzhalterzeichen - (Bindestrich) gibt einen Bereich an. Im Feld "Tag" steht 1-15 für die Tage 1 bis 15 des angegebenen Monats.
*	Das Platzhalterzeichen * (Sternchen) steht für alle Werte im Feld. Im Feld für die Stundenangaben steht * für alle Stunden.
/	Das Platzhalterzeichen / (Schrägstrich) steht für schrittweise Steigerungen. Im Feld „Minuten“ könnten Sie 1/10 eingeben, um jede 10. Minute beginnend mit der ersten Minute der Stunde anzugeben. 1/10 gibt daher die erste, 11., 21., 31. usw. Minute an.
?	Das Platzhalterzeichen ? (Fragezeichen) steht für einen bestimmten Wert. In das ay-of-month Feld D könnten Sie 7 eingeben und wenn es Ihnen egal war, welcher Wochentag der 7. war, könnten Sie ? in das ay-of-week Feld D eingeben.
L	Der L Platzhalter in den ay-of-week Feldern D ay-of-month oder D gibt den letzten Tag des Monats oder der Woche an.
W	Der W Platzhalter im ay-of-month Feld D gibt einen Wochentag an. Im ay-of-month Feld D gibt 3W den Tag an, der dem dritten Wochentag des Monats am nächsten ist.

Platzhalter	Beschreibung
#	Der # Platzhalter im day-of-week Feld gefolgt von einer Zahl zwischen eins und fünf gibt einen bestimmten Tag des Monats an. 5#3 gibt den 3. Donnerstag des Monats an.

Rate-Ausdrücke

Rate-Ausdrücke bestehen aus den folgenden zwei Pflichtfeldern. Felder werden durch Leerzeichen voneinander getrennt.

Pflichtfelder für Rate-Ausdrücke

Feld	Werte
Wert	positive Zahl, z. B. 1 oder 15
Einheit	`minute` `minutes` `hour` `hours` `day` `days`

Wenn der Wert gleich 1 ist, muss die Einheit im Singular stehen. Wenn die Werte größer als 1 sind, muss die Einheit im Plural stehen. Beispielsweise sind `rate(1 hours)` und `rate(5 hour)` ungültige, `rate(1 hour)` und `rate(5 hours)` jedoch gültige Werte.

Themen

- Cron- und Rate-Ausdrücke für Zuordnungen
- Cron- und Rate-Ausdrücke für Wartungsfenster

Cron- und Rate-Ausdrücke für Zuordnungen

Dieser Abschnitt enthält Beispiele für Cron- und Rate-Ausdrücke für State Manager-Zuordnungen. Bevor Sie einen dieser Ausdrücke erstellen, beachten Sie die folgenden Informationen:

- Zuordnungen unterstützen die folgenden Cron-Ausdrücke: Alle 1/2, 1, 2, 4, 8 oder 12 Stunden; jeden Tag, jede Woche oder jeden angegebenen Tag und jede bestimmte Uhrzeit der Woche; ein bestimmter Tag in einer bestimmten Woche des Monats oder der letzte x-Tag des Monats zu einer bestimmten Zeit.
- Zuordnungen unterstützen die folgenden Rate-Ausdrücke: Intervalle von mindestens 30 Minuten und weniger als 31 Tagen.
- Wenn Sie das optionale Feld `Seconds` angeben, kann dessen Wert 0 (null) sein. Zum Beispiel: `cron(0 */30 * * * ? *)`
- Für eine Zuordnung, die Metadaten für Inventory sammelt, eine Funktion von AWS Systems Manager, wird empfohlen, einen Rate-Ausdruck zu verwenden.
- State Manager unterstützt derzeit nicht die Angabe von Monaten in Cron-Ausdrücken für Assoziationen.

Assoziationen unterstützen Cron-Ausdrücke, die einen Wochentag und das Zahlenzeichen (#) enthalten, um den x-ten Tag eines Monats für die Ausführung einer Assoziation anzugeben. Hier ist ein Beispiel, das am dritten Dienstag jeden Monats um 23:30 Uhr UTC einen Cron-Zeitplan ausführt:

```
cron(30 23 ? * TUE#3 *)
```

Hier ist ein Beispiel, das am zweiten Donnerstag jeden Monats um Mitternacht UTC läuft:

```
cron(0 0 ? * THU#2 *)
```

Assoziationen unterstützen auch das (L)-Zeichen, um dens letzten XTag des Monats anzugeben. Hier ist ein Beispiel, das am letzten Dienstag jeden Monats um Mitternacht UTC einen Cron-Zeitplan ausführt:

```
cron(0 0 ? * 3L *)
```

Um weiter zu steuern, wann eine Assoziation ausgeführt wird, z. B. wenn Sie zwei Tage nach dem Patch-Dienstag eine Assoziation ausführen möchten, können Sie einen Offset angeben. Ein Offset definiert, wie viele Tage nach dem geplanten Tag gewartet werden müssen, um eine

Assoziation auszuführen. Wenn Sie beispielsweise einen Cron-Zeitplan mit `cron(0 0 ? * THU#2 *)` angegeben haben, können Sie die Nummer 3 im Schedule offset (Planversatz)-Feld angeben, um die Assoziation jeden Sonntag nach dem zweiten Donnerstag im Monat auszuführen.

Um Offsets zu verwenden, müssen Sie entweder Apply association only at the next specified Cron interval (Übernehmen der Assoziation erst für das nächste angegebene Cron-Intervall)-Option in der Konsole auswählen oder Sie müssen den Nutze`--apply-only-at-cron-interval`-Parameter über die Befehlszeile angeben. Diese Option sagt State Manager eine Assoziation nicht unmittelbar nach der Erstellung auszuführen.

Die folgende Tabelle zeigt die Cron-Beispiele für Zuordnungen.

Cron-Beispiele für Zuordnungen

Beispiel	Details
cron(0/30 * * * ? *)	Alle 30 Minuten
cron(0 0/1 * * ? *)	Stündlich
cron(0 0/2 * * ? *)	Alle 2 Stunden
cron(0 0/4 * * ? *)	Alle 4 Stunden
cron(0 0/8 * * ? *)	Alle 8 Stunden
cron(0 0/12 * * ? *)	Alle 12 Stunden
cron(15 13 ? * * *)	Täglich um 13:15 Uhr
cron(15 13 ? * MON *)	Jeden Montag um 13:15 Uhr
cron(30 23 ? * TUE#3 *)	Jeden dritten Dienstag im Monat um 23:30 Uhr

Hier sind einige Rate-Beispiele für Zuordnungen.

Rate-Beispiele für Zuordnungen

Beispiel	Details
rate(30 minutes)	Alle 30 Minuten

Beispiel	Details
rate(1 hour)	Stündlich
rate(5 hours)	Alle 5 Stunden
rate(15 days)	Alle 15 Tage

AWS CLI-Beispiele für Zuordnungen

Um State Manager-Zuordnungen über die AWS CLI zu erstellen, fügen Sie den Parameter `--schedule-expression` mit einem Cron- oder Rate-Ausdruck hinzu. Die folgenden Beispiele verwenden die AWS CLI auf einem lokalen Linux-Computer.

> **Note**
>
> Wenn Sie eine neue Zuordnung erstellen, führt das System diese standardmäßig sofort nach der Erstellung und dann nach dem angegebenen Zeitplan aus. Geben Sie `--apply-only-at-cron-interval` an, damit die Zuordnung nicht unmittelbar nach der Erstellung ausgeführt wird. Dieser Parameter wird nicht für Rate-Ausdrücke unterstützt.

```
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
```

Cron- und Rate-Ausdrücke für Wartungsfenster

Dieser Abschnitt enthält Beispiele für Cron- und Rate-Ausdrücke für Wartungsfenster.

Anders als State Manager-Zuordnungen unterstützen Wartungsfenster alle Cron- und Rate-Ausdrücke. Dies umfasst die Unterstützung für Werte im Sekundenfeld.

Beispielsweise führt der folgende Cron-Ausdruck mit 6 Feldern jeden Tag um 9:30 Uhr ein Wartungsfenster aus.

```
cron(30 09 ? * * *)
```

Durch Hinzufügen eines Werts zum Feld `Seconds` führt der folgende Cron-Ausdruck mit 7 Feldern jeden Tag um 9:30:24 Uhr ein Wartungsfenster aus.

```
cron(24 30 09 ? * * *)
```

Die folgende Tabelle enthält zusätzliche Beispiele für Cron-Ausdrücke mit 6 Feldern für Wartungsfenster.

Cron-Beispiele für Wartungsfenster

Beispiel	Details
cron(0 2 ? * THU#3 *)	02:00 Uhr jeden dritten Donnerstag im Monat
cron(15 10 ? * * *)	10:15 Uhr jeden Tag
cron(15 10 ? * MON-FRI *)	10:15 Uhr jeden Montag, Dienstag, Mittwoch, Donnerstag und Freitag
cron(0 2 L * ? *)	02:00 Uhr jeden letzten Tag im Monat
cron(15 10 ? * 6L *)	10:15 Uhr jeden letzten Freitag im Monat

Die folgende Tabelle enthält Beispiele für Raten von Wartungsfenstern.

Rate-Beispiele für Wartungsfenster

Beispiel	Details
rate(30 minutes)	Alle 30 Minuten
rate(1 hour)	Stündlich
rate(5 hours)	Alle 5 Stunden
rate(25 days)	Alle 25 Tage

AWS CLI-Beispiele für Wartungsfenster

Um Wartungsfenster mithilfe der AWS CLI zu erstellen, fügen Sie den Parameter `--schedule` mit einem Cron- oder Rate-Ausdruck oder einen Zeitstempel ein. Die folgenden Beispiele verwenden die AWS CLI auf einem lokalen Linux-Computer.

```
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
```

Weitere Informationen

CRON-Ausdruck bei der Wikipedia-Webseite

Referenz: ec2messages, ssmmessages und andere API-Operationen

Wenn Sie API-Operationen überwachen, werden möglicherweise Aufrufe der folgenden Operationen angezeigt:

- `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`

Dies sind spezielle Operationen, die von verwendet werden AWS Systems Manager, wie im Rest dieses Themas beschrieben.

API-Operationen (**`ssmmessages`**und **`ec2messages`** Endpunkte) im Zusammenhang mit Agenten

ssmmessages-API-Operationen

Systems Manager verwendet den `ssmmessages` Endpunkt für die folgenden zwei Arten von API-Vorgängen:

- Operationen von SSM Agent bisSession Manager, eine Fähigkeit von AWS Systems Manager, in der Cloud. Dieser Endpunkt ist zum Erstellen und Löschen von Sitzungskanälen mit dem Session Manager-Service in der Cloud erforderlich. Wenn Konnektivität zulässig ist, SSM Agent empfängt er darüber hinaus `Command` DokumenteAmazon Message Gateway Service. Wenn Konnektivität nicht zulässig ist, SSM Agent empfängt `Command` Dokumente über dieAmazon Message Delivery Service. Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Session Manager Message Gateway Service.
- Operationen vom Systems Manager Agent (SSM Agent) zum Systems Manager Manager-Dienst in der Cloud.

ec2messages-API-Operationen

`ec2messages:*`-API-Operationen werden an den Amazon Message Delivery Service-Endpunkt gemacht. Systems Manager verwendet diesen Endpunkt für API-Operationen vom Systems-Manager-Agent (SSM Agent) an den Systems-Manager-Service in der Cloud.

> **Important**
>
> `ec2messages:*`API-Operationen werden nur für Operationen unterstützt AWS-Regionen , die vor 2024 gestartet wurden. In Regionen, die 2024 und später eingeführt wurden, werden nur `ssmmessages:*` API-Operationen unterstützt.

Rangfolge der Endpunktverbindungen

Ab Version 3.3.40.0 von begann Systems ManagerSSM Agent, den Endpunkt (), wann immer verfügbar, anstelle des `ssmmessages:*` `ec2messages:*` Endpunkts (Amazon Message Gateway Service) zu verwenden. Amazon Message Delivery Service

Wenn Sie `ssmmessages:*` in Ihren AWS Identity and Access Management (IAM-) Berechtigungsrichtlinien Zugriff auf gewähren, wird SSM Agent eine Verbindung zum `ssmmessages:*` Endpunkt hergestellt, auch wenn Ihr IAM-Instanzprofil so konfiguriert ist, dass beide Endpunkte zugelassen sind. Dazu gehören Richtlinien für IAM-Instanzprofile und IAM-Servicerollen, die Sie selbst erstellt haben, sowie für IAM-Instanzprofile, die mit der Host-Management-Konfiguration und der Quick SetupStandard-Host-Management-Konfiguration erstellt wurden.

Wenn Sie Berechtigungen für beide Endgeräte bereitgestellt und API-Operationen beispielsweise mithilfe von CloudWatch Metrics überwacht haben, werden Ihnen keine Aufrufe von angezeigt. `ec2messages:*`

Für AWS-Regionen Produkte, die vor 2024 veröffentlicht wurden: Sie können zu diesem Zeitpunkt problemlos `ec2messages:*` Berechtigungen aus Ihren Richtlinien entfernen.

Failover der Endpunktverbindung

Nur für vor 2024 AWS-Regionen gestartete Versionen: Wenn Ihr IAM-Instanzprofil zum `ssmmessages:*` Zeitpunkt des Starts des Agenten keine Berechtigungen bereitstellt, sondern lediglich die SSM Agent `ec2messages:*` Verbindung zum `ec2messages:*` Endpunkt herstellt. Wenn Sie beide haben `ssmmessages:*` und `ec2messages:*` zum Zeitpunkt des SSM Agent Starts, aber entfernen, `ssmmessages:*` nachdem der Agent gestartet wurde, wechselt SSM Agent bald die Verbindung zum `ec2messages:*` Endpunkt. Für Regionen, die 2024 und später eingeführt wurden, wird nur der `ssmmessages:*` Endpunkt unterstützt.

Weitere Informationen zu den `ec2messages:*` Endpunkten `ssmmessages` und finden Sie in den folgenden Themen in der AWS Service Authorization Reference.

- Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Message Gateway Service (`ssmmessages`).
- Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Message Delivery Service (`ec2messages:*`)

ssm:*API-Operationen im Zusammenhang mit Namespace-Instanzen

`DescribeDocumentParameters`

Systems Manager führt diesen API-Vorgang aus, um bestimmte Knoten in der Amazon EC2 EC2-Konsole zu rendern. Die Ergebnisse des `DescribeDocumentParameters` Vorgangs werden im Knoten Dokumente angezeigt.

`DescribeInstanceProperties`

Systems Manager führt diese API-Operationen aus, um bestimmte Knoten in der Amazon EC2 EC2-Konsole zu rendern. Ergebnisse der `DescribeInstanceProperties`-Operation werden im Knoten Fleet Manager angezeigt.

`GetCalendar`

Systems Manager führt diesen API-Vorgang aus, um Change Calendar Typdokumente in der Change Calendar Konsole zu rendern.

`GetManifest`

SSM Agentführt diesen API-Vorgang aus, um die Systemanforderungen für die Installation oder Aktualisierung einer bestimmten Version eines AWS Systems Manager Distributor Pakets zu ermitteln. Dies ist ein älterer API-Vorgang, der nicht verfügbar ist, wenn er nach 2017 AWS-Regionen gestartet wurde.

`ListInstanceAssociations`

SSM Agentführt diesen API-Vorgang aus, um festzustellen, ob eine neue State Manager Verknüpfung verfügbar ist. Dieser API-Vorgang ist für die Funktion von State Manager erforderlich.

`PutCalendar`

Systems Manager führt diesen API-Vorgang aus, um Change Calendar Typdokumente in der Change Calendar Konsole zu aktualisieren.

`PutConfigurePackageResult`

SSM Agentführt diesen API-Vorgang aus, um Messdaten zu Installationsfehlern und Latenz für öffentliche Distributor-Pakete auf dem Konto des Paketbesitzers zu veröffentlichen.

`RegisterManagedInstance`

SSM Agentführt diesen API-Vorgang für die folgenden Szenarien aus:

- Um einen lokalen Server oder eine virtuelle Maschine (VM) mit einem Aktivierungscode und einer ID bei Systems Manager als verwaltete Instanz zu registrieren.
- Um AWS IoT Greengrass Version 2 Anmeldeinformationen zu registrieren.

Dieser Vorgang wird auch von Amazon-EC2-Instances aufgerufen, auf denen SSM Agent-Version 3.1.x oder höher ausgeführt wird.

`RequestManagedInstanceRoleToken`

SSM Agentführt diesen API-Vorgang aus, um temporäre Anmeldeinformationen für den Zugriff auf den verwalteten Knoten abzurufen.

`UpdateInstanceAssociationStatus`

SSM Agentführt diesen API-Vorgang aus, um eine Zuordnung zu aktualisieren. Dieser API-Vorgang ist erforderlich State Manager AWS Systems Manager, damit er funktioniert.

`UpdateInstanceInformation`

SSM Agentruft alle 5 Minuten den Systems Manager Manager-Dienst in der Cloud auf, um Heartbeat-Informationen bereitzustellen. Dieser Aufruf ist erforderlich, um einen Heartbeat mit dem Agent beizubehalten, damit der Service erkennt, dass der Agent erwartungsgemäß funktioniert.

`UpdateManagedInstancePublicKey`

SSM Agentführt diesen API-Vorgang aus, um den öffentlichen Schlüssel bereitzustellen, nachdem das key pair auf dem verwalteten Knoten rotiert wurde. Der öffentliche Schlüssel wird zur Authentifizierung der mit dem privaten Schlüssel signierten Anfragen verwendet, um temporäre Anmeldeinformationen von Systems Manager zu erhalten.

Referenz: Erstellen formatierter Datums- und Uhrzeitzeichenfolgen für Systems Manager

AWS Systems Manager-API-Vorgänge akzeptieren Filter, um die Anzahl der von einer Anforderung zurückgegebenen Ergebnisse zu begrenzen. Einige dieser API-Vorgänge akzeptieren Filter, die zur Darstellung eines bestimmten Datums und einer bestimmten Uhrzeit eine formatierte Zeichenfolge erfordern. Beispielsweise akzeptiert der API-Vorgang `DescribeSessions` die Schlüssel `InvokedAfter` und `InvokedBefore` als gültige Werte für ein `SessionFilter`-Objekt. Ein weiteres Beispiel ist der API-Vorgang `DescribeAutomationExecutions`. Dieser akzeptiert die Schlüssel `StartTimeBefore` und `StartTimeAfter` als gültige Werte für ein

`AutomationExecutionFilter`-Objekt. Die Werte, die Sie für diese Schlüssel beim Filtern Ihrer Anforderungen angeben, müssen dem ISO 8601-Standard entsprechen. Weitere Informationen zu ISO 8601 finden Sie unter ISO 8601.

Diese formatierten Datums- und Uhrzeitzeichenfolgen sind nicht auf Filter beschränkt. Es gibt auch API-Vorgänge, die eine im ISO 8601-Format formatierte Zeichenfolge erfordern, um ein bestimmtes Datum und eine bestimmte Uhrzeit darzustellen, wenn ein Wert für einen Anforderungsparameter angegeben wird. Ein Beispiel ist der Anforderungsparameter `AtTime` für den Vorgang `GetCalendarState`. Das Erstellen dieser Zeichenfolgen ist schwierig. Die Beispiele in diesem Thema helfen Ihnen, formatierte Datums- und Uhrzeitzeichenfolgen für die Verwendung mit Systems Manager-API-Vorgängen zu erstellen.

Formatieren von Datums- und Uhrzeitzeichenfolgen für Systems Manager

Im Folgenden finden Sie ein Beispiel für eine im ISO 8601-Format formatierte Datums- und Uhrzeitzeichenfolge.

```
2020-05-08T15:16:43Z
```

Dies entspricht dem 8. Mai 2020 um 15:16 Uhr koordinierter Weltzeit (UTC). Der Kalenderdatumsbereich der Zeichenfolge wird durch ein vierstelliges Jahr, einen zweistelligen Monat und einen zweistelligen Tag dargestellt, getrennt durch Bindestriche. Dies kann im folgenden Format dargestellt werden.

```
YYYY-MM-DD
```

Der Zeitbereich der Zeichenfolge beginnt mit dem Buchstaben „T“ als Trennzeichen. Er wird durch eine zweistellige Stunde, eine zweistellige Minute und eine zweistellige Sekunde dargestellt, getrennt durch Doppelpunkte. Dies kann im folgenden Format dargestellt werden.

```
hh:mm:ss
```

Der Zeitbereich der Zeichenfolge endet mit dem Buchstaben „Z“, der den UTC-Standard angibt.

Erstellen benutzerdefinierter Datums- und Uhrzeitzeichenfolgen für Systems Manager

Sie können benutzerdefinierte Datums- und Uhrzeitzeichenfolgen auf Ihrem lokalen Computer mit Ihrem bevorzugten Befehlszeilen-Tool erstellen. Die Syntax, die Sie zum Erstellen einer im ISO 8601-

Format formatierten Datums- und Uhrzeitzeichenfolge verwenden, ist vom Betriebssystem Ihres lokalen Computers abhängig. Im Folgenden finden Sie Beispiele dafür, wie Sie `date` aus coreutils von GNU unter Linux oder PowerShell unter Windows verwenden, um eine im ISO 8601-Format formatierte Datums- und Uhrzeitzeichenfolge zu erstellen.

coreutils

```
date '+%Y-%m-%dT%H:%M:%SZ'
```

PowerShell

```
(Get-Date).ToString("yyyy-MM-ddTH:mm:ssZ")
```

Wenn Sie mit Systems Manager-API-Vorgängen arbeiten, müssen Sie möglicherweise zu Berichts- oder Fehlerbehebungszwecken historische Datums- und Uhrzeitzeichenfolgen erstellen. Im Folgenden finden Sie Beispiele dafür, wie Sie benutzerdefinierte, im ISO 8601-Format formatierte historische Datums- und Uhrzeitzeichenfolgen für den AWS Tools for PowerShell und die AWS Command Line Interface (AWS CLI) erstellen und verwenden.

AWS CLI

- Rufen Sie die letzte Woche des Befehlsverlaufs für ein SSM-Dokument ab.

  ```
  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
  ```

- Rufen Sie die letzte Woche des Automatisierungsausführungsverlaufs ab.

  ```
  lastWeekStamp=$(date '+%Y-%m-%dT%H:%M:%SZ' -d '7 days ago')

  aws ssm describe-automation-executions \
      --filters Key=StartTimeAfter,Values=$lastWeekStamp
  ```

- Rufen Sie den letzten Monat des Sitzungsverlaufs ab.

```
lastWeekStamp=$(date '+%Y-%m-%dT%H:%M:%SZ' -d '30 days ago')

aws ssm describe-sessions \
    --state History \
    --filters key=InvokedAfter,value=$lastWeekStamp
```

AWS Tools for PowerShell

- Rufen Sie die letzte Woche des Befehlsverlaufs für ein SSM-Dokument ab.

```
$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
```

- Rufen Sie die letzte Woche des Automatisierungsausführungsverlaufs ab.

```
$lastWeekStamp = (Get-Date).AddDays(-7).ToString("yyyy-MM-ddTH:mm:ssZ")

Get-SSMAutomationExecutionList `
    -Filters @{Key="StartTimeAfter";Values=$lastWeekStamp}
```

- Rufen Sie den letzten Monat des Sitzungsverlaufs ab.

```
$lastWeekStamp = (Get-Date).AddDays(-30).ToString("yyyy-MM-ddTH:mm:ssZ")

Get-SSMSession `
    -State History `
```

```
    -Filters @{Key="InvokedAfter";Value=$lastWeekStamp}
```

Anwendungsfälle und bewährte Methoden

In diesem Thema werden allgemeine Anwendungsfälle und bewährte Methoden für AWS Systems Manager Funktionen aufgeführt. Wo verfügbar, sind in diesem Thema auch Links zu relevanten Blogbeiträgen und technischer Dokumentation enthalten.

> **Note**
>
> Die Abschnittstitel sind aktive Links zum entsprechenden Abschnitt in der technischen Dokumentation.

Automation

- Erstellen Sie Self-Service-Automation-Runbooks für Infrastruktur.
- Verwenden Sie Automation, eine Funktion von AWS Systems Manager, um das Erstellen Amazon Machine Images (AMIs) anhand von AWS Marketplace oder benutzerdefinierten Dokumenten zu vereinfachenAMIs, indem Sie öffentliche Systems Manager Manager-Dokumente (SSM-Dokumente) verwenden oder Ihre eigenen Workflows erstellen.
- Erstellen und verwalten Sie AMIs mithilfe der `AWS-UpdateLinuxAmi` und `AWS-UpdateWindowsAmi` Automation-Runbooks oder mithilfe benutzerdefinierter Automation-Runbooks, die Sie erstellen.

Inventory

- Verwenden Sie Inventory, eine Funktion von AWS Systems Manager, mit, AWS Config um Ihre Anwendungskonfigurationen im Laufe der Zeit zu überprüfen.

Maintenance Windows

- Definieren Sie einen Zeitplan zur Ausführung potenziell störender Aktionen auf Ihren Knoten, wie z. B. Betriebssystem-Patches, Treiber-Updates oder Software-Installationen.
- Informationen zu den Unterschieden zwischen State Manager und und Maintenance Windows zu den Funktionen von AWS Systems Manager finden Sie unterAuswahl zwischen State Manager und Maintenance Windows.

Parameter Store

- Verwenden SieParameter Store, eine Funktion von AWS Systems Manager, um globale Konfigurationseinstellungen zentral zu verwalten.
- Wie AWS Systems ManagerParameter Store verwendet AWS KMS.
- Verweisen Sie auf AWS Secrets Manager Geheimnisse aus Parameter Store Parametern.

Patch Manager

- Verwenden Sie Patch Manager eine Funktion von AWS Systems Manager, um Patches in großem Umfang bereitzustellen und die Transparenz der Flottenkonformität auf Ihren Knoten zu erhöhen.
- Integrieren Sie Patch Manager in AWS Security Hub, um Warnungen zu erhalten, wenn Knoten in Ihrer Flotte nicht konform sind, und überwachen Sie den Patching-Status Ihrer Flotten hinsichtlich der Sicherheit. Für die Nutzung von Security Hub wird eine Gebühr erhoben. Weitere Informationen finden Sie unter -Preisgestaltung.
- Verwenden Sie jeweils nur eine Methode zum Scannen verwalteter Knoten auf Patch-Compliance, um unbeabsichtigtes Überschreiben von Compliance-Daten zu vermeiden.

Run Command

- Verwalten Sie Instances in großem Umfang, ohne SSH-Zugriff, mit EC2 Run Command.
- Prüfen Sie alle API-AufrufeRun Command, die von oder im Namen einer Funktion von AWS Systems Manager, durchgeführt AWS CloudTrail werden.
- Wenn Sie einen Befehl mit Run Command senden, schließen Sie keine vertraulichen Informationen ein, die als Klartext formatiert sind, z. B. Passwörter, Konfigurationsdaten oder andere Geheimnisse. Alle Systems Manager Manager-API-Aktivitäten in Ihrem Konto werden in einem S3-Bucket für AWS CloudTrail Protokolle protokolliert. Dies bedeutet, dass jeder Benutzer mit Zugriff auf den S3-Bucket die Klartextwerte dieser Geheimnisse anzeigen kann. Aus diesem Grund empfehlen wir, `SecureString`-Parameter zu erstellen und zu verwenden, um die sensiblen Daten zu verschlüsseln, die Sie in Ihren Systems-Manager-Operationen verwenden.

  Weitere Informationen finden Sie unter Einschränken des Zugriffs auf Systems Manager-Parameter mithilfe von IAM-Richtlinien.

> **Note**
>
> Standardmäßig werden die von an Ihren Bucket übermittelten Protokolldateien durch CloudTrail serverseitige Amazon-Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) verschlüsselt. Um eine Sicherheitsebene bereitzustellen, die direkt verwaltet werden kann, können Sie stattdessen serverseitige Verschlüsselung mit AWS KMS verwalteten Schlüsseln (SSE-KMS) für Ihre Protokolldateien verwenden. CloudTrail
> Weitere Informationen finden Sie im Benutzerhandbuch unter Verschlüsseln von CloudTrail Protokolldateien mit AWS KMS verwalteten Schlüsseln (SSE-KMS).AWS CloudTrail

- Verwenden Sie die Ziel- und Tempo-Steuerungsfunktionen in Run Command zum Ausführen zwischengespeicherter Befehle.
- Verwenden Sie mithilfe von (IAM-) Richtlinien detaillierte Zugriffsberechtigungen für Run Command (und alle Systems Manager Manager-Funktionen). AWS Identity and Access Management

Session Manager

- Prüfen Sie die Sitzungsaktivität in Ihrer Nutzung. AWS-Konto AWS CloudTrail
- Protokollieren Sie Sitzungsdaten in Ihrer AWS-Konto Nutzung von Amazon CloudWatch Logs oder Amazon S3.
- Steuerung des Benutzer-Sitzungszugriffs auf Instances.
- Beschränken des Zugriffs auf Befehle in einer Sitzung.
- Deaktivieren oder Aktivieren der Administratorberechtigungen für das SSM-Benutzerkonto.

State Manager

- Aktualisieren Sie den SSM Agent mindestens einmal pro Monat mit dem vorkonfigurierten `AWS-UpdateSSMAgent`-Dokument.
- (Windows) Laden Sie das PowerShell oder DSC-Modul auf Amazon Simple Storage Service (Amazon S3) hoch und verwenden Sie `AWS-InstallPowerShellModule` es.
- Erstellen Sie Anwendungsgruppen für Ihre Knoten mit Tags. Wählen Sie anschließend Knoten mit dem `Targets`-Parameter aus, anstatt individuelle Knoten-IDs anzugeben.

- Beseitigen Sie die von Amazon Inspector erzeugten Ergebnisse mit Systems Manager automatisch.
- Verwenden Sie ein zentrales Konfigurations-Repository für Ihre SSM-Dokumente und geben Sie Dokumente in Ihrer Organisation frei.
- Informationen zu den Unterschieden zwischen State Manager und Maintenance Windows finden Sie unterAuswahl zwischen State Manager und Maintenance Windows.

Verwaltete Knoten

- Systems Manager erfordert genaue Zeitreferenzen, um seine Operationen auszuführen. Wenn in Ihrem Knoten das Datum und die Uhrzeit nicht korrekt festgelegt wurden, stimmen sie möglicherweise nicht mit dem Signaturdatum Ihrer API-Anforderungen überein. Dies kann zu Fehlern oder unvollständiger Funktionalität führen. Beispiel: Knoten mit falschen Zeiteinstellungen werden nicht in die Liste der verwalteten Knoten aufgenommen.

  Informationen zum Einstellen der Uhrzeit auf Ihren Knoten finden Sie unter Zeit für Ihre Amazon EC2 EC2-Instance festlegen.
- Überprüfen Sie auf verwalteten Linux-Knoten die Signatur von SSM Agent.

Weitere Informationen

- Bewährte Methoden für die Sicherheit für Systems Manager

Löschen von Systems Manager Ressourcen und Artefakten

Als bewährte Methode wird empfohlen, Systems Manager Ressourcen und Artefakte zu löschen, wenn Sie keine Daten zu diesen Ressourcen mehr anzeigen oder die Artefakte in irgendeiner Weise verwenden müssen. In der folgenden Tabelle werden die einzelnen Systems Manager-Funktionen oder Artefakte sowie ein Link zu weiteren Informationen zum Löschen der von Systems Manager erstellten Ressourcen oder Artefakte aufgeführt.

Funktion oder Artefakt	Details
Application Manager	Sie können keine Anwendung im Applicati on Manager löschen, Sie können aber eine Anwendung aus dem Dienst entfernen, indem

Funktion oder Artefakt	Details
	Sie die zugrunde liegenden Tags, -Ressourc e Groups, oder AWS CloudFormation Stacks löschen.
Automatisierung	Wenn Sie AWS Ressourcen mithilfe von Systems Manager Automation erstellen , müssen Sie diese Ressourcen manuell löschen, indem Sie die entsprechenden Ressourcen verwenden AWS Managemen t Console. Wenn Sie ein benutzerdefinierte s Runbook erstellt haben, können Sie das zugrunde liegende SSM-Dokument löschen. Weitere Informationen finden Sie unter Löschen benutzerdefinierter SSM-Dokumente.
Change Calendar	Sie können einen Änderungskalender und ein Änderungskalenderereignis löschen. Weitere Informationen finden Sie unter Einen Änderungskalender löschen und Löschen eines Change Calendar-Ereignisses.
Change Manager	Sie können eine Änderungsvorlage löschen. Weitere Informationen finden Sie unter Löschen von Änderungsvorlagen.
-Compliance	Systems Manager Compliance zeigt automatis ch Konformitätsdaten zu Patch Manager-P atching und State Manager-Verknüpfungen an. Sie können diese Daten nicht löschen. Wenn Sie eine Ressourcendaten-Synchronisierung konfiguriert haben, um Compliance-Daten in einem S3-Bucket zu zentralisieren, können Sie die Synchronisierung löschen. Weitere Informationen finden Sie unter Löschen einer Ressource Data Sync für Compliance.

Funktion oder Artefakt	Details
Distributor	Sie können Pakete in Distributor löschen. Weitere Informationen finden Sie unter Löschen eines Pakets.
Explorer	Sie können die Verbindung zu den Quellen trennen, aus denen die Explorer Daten stammen OpsData. Weitere Informationen finden Sie unter Bearbeiten von Systems-Manager-Explorer-Datenquellen. Sie können auch eine Ressourcendatensynchronisierung löschen, die von Explorer zum Aggregieren OpsData und OpsItems von mehreren AWS-Regionen AND-Konten zu einem einzigen Amazon Simple Storage Service (Amazon S3) -Bucket verwendet wird. Weitere Informationen finden Sie unter Löschen einer Systems-Manager-Explorer-Ressourcendatensynchronisierung. Informationen zum Löschen eines S3-Buckets finden Sie unter Löschen eines Buckets im Entwicklerhandbuch für Amazon Simple Email Service.
Fleet Manager	Sie können einen verwalteten Knoten nicht mit Fleet Manager löschen. Sie müssen Amazon Elastic Compute Cloud (Amazon EC2) verwenden. Weitere Informationen finden Sie unter Beenden Ihrer Instance (Linux) und Beenden Ihrer Instance (Windows).

Funktion oder Artefakt	Details
-Bestand	Sie können die Inventory-Datensammlung stoppen, indem Sie die State Manager-V erknüpfungen löschen, die den Zeitplan und die Ressourcen definieren, aus denen Metadaten gesammelt werden sollen. Weitere Informati onen finden Sie unter Anhalten der Datenerfa ssung und Löschen von Bestandsdaten. Wenn Sie AWS Systems Manager Inventar nicht mehr verwenden möchten, um Metadaten zu Ihren AWS Ressourcen anzuzeigen, empfehlen wir außerdem, die für die Erfassung von Inventardaten verwendeten Ressource ndatensynchronisationen zu löschen. Weitere Informationen finden Sie unter Löschen einer Inventory Resource Data Sync.
Maintenance Windows	Sie können ein Wartungsfenster, ein Wartungsf ensterziel und eine Aufgabe im Wartungsf enster löschen. Weitere Informationen finden Sie unter Aktualisieren oder Löschen von Wartungsfenster-Ressourcen (Konsole).
OpsCenter	Sie können eine Person löschen, OpsItem indem Sie den Vorgang „OpsItemAPI löschen“ mit dem AWS Command Line Interface oder dem AWS SDK aufrufen. Sie können ein OpsItem nicht in der AWS Management Console löschen. Weitere Informationen finden Sie unter Löschen Sie OpsItems.
Parameter Store	Sie können einen Parameter löschen, den Sie erstellt haben. Weitere Informationen finden Sie unter Löschen von Systems-Manager-Pa rametern.

Funktion oder Artefakt	Details
Patch Manager	Sie können eine benutzerdefinierte Patch-Baseline löschen. Weitere Informationen finden Sie unter Aktualisieren oder Löschen einer benutzerdefinierten Patch-Baseline.
Quick Setup	Sie können mit Quick Setup erstellte Zuordnungen löschen. Die Zuordnungen werden gespeichert und von State Manager verarbeitet. Weitere Informationen finden Sie unter Löschen von Zuordnungen.
Run Command	Nachdem die Verarbeitung eines Befehls abgeschlossen ist, werden Informationen darüber in Befehls-Verlauf-Registerkarte gespeichert. Sie können keine Informationen aus der Befehls-Verlauf-Registerkarte löschen.
Servicegebundene Rolle	Systems Manager erstellt automatisch dienstverknüpfte Rollen für einige Funktionen. Sie können diese Rollen löschen. Weitere Informationen finden Sie unter Löschen einer `AWSServiceRoleForAmazonSSM` -serviceverknüpften Rolle für Systems Manager.
Session Manager	Session Manager speichert keine Daten zu Ihren Ressourcen, nachdem Sie eine Sitzung beendet haben. Weitere Informationen zum Beenden einer Sitzung finden Sie unter Beenden einer Sitzung.

Funktion oder Artefakt	Details
SSM Agent	Sie können SSM Agent manuell von Ihren Knoten deinstallieren. Weitere Informationen finden Sie unter den folgenden Themen. • Linux: Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für Linux • macOS: Manuelles Installieren und Deinstallieren SSM Agent auf EC2-Instances für macOS • Windows Server: Öffnen Sie Bedienfeld und wählen Sie dann Programme hinzufügen/entfernen.
State Manager	Sie können eine Verknüpfung löschen. Weitere Informationen finden Sie unter Löschen von Zuordnungen.
Systems Manager-Dokumentenservice	Sie können keine von AWS oder bereitgestellten Runbooks löschen AWS Support, aber Sie können benutzerdefinierte Runbooks löschen. Weitere Informationen finden Sie unter Löschen benutzerdefinierter SSM-Dokumente.

Auswahl zwischen State Manager und Maintenance Windows

State Manager und Maintenance Windows, beide Funktionen von AWS Systems Manager, können ähnliche Arten von Updates auf Ihren verwalteten Knoten durchführen. Welche Option Sie wählen, hängt davon ab, ob Sie die System-Compliance automatisieren oder zeitkritische Aufgaben mit hoher Priorität während der von Ihnen angegebenen Zeiträume ausführen müssen.

State Manager und Maintenance Windows: Hauptanwendungsfälle

State Manager, eine Funktion von AWS Systems Manager, legt die Zielstatuskonfiguration für verwaltete Knoten und AWS Ressourcen in Ihrem fest und verwaltet sie AWS-Konto. Sie können Kombinationen von Konfigurationen und Zielen als Zuordnungen definieren. State Manager ist

die empfohlene Funktion, wenn Sie alle verwalteten Knoten in Ihrem Konto in einem konsistenten Zustand erhalten möchten, Amazon EC2 Auto Scaling zum Generieren neuer Knoten verwenden oder strenge Anforderungen an die Compliance-Berichterstattung für die verwalteten Knoten in Ihrem Konto haben möchten.

Die wichtigsten Anwendungsfälle für State Manager lauten wie folgt:

- Auto-Scaling-Szenarien: State Manager kann alle neuen Knoten, die innerhalb eines Kontos gestartet werden, entweder manuell oder über Auto-Scaling-Gruppen überwachen. Wenn im Konto Verknüpfungen vorhanden sind, die auf diesen neuen Knoten ausgerichtet sind (über Tags oder alle Knoten), wird diese bestimmte Zuordnung automatisch auf den neuen Knoten angewendet.
- Compliance-Berichtswesen: State Manager kann das Compliance-Berichtswesen der gewünschten Status für Ressourcen in Ihrem Konto steuern.
- Unterstützt alle Knoten: State Manager kann alle Knoten innerhalb eines bestimmten Kontos anvisieren.

Ein Wartungsfenster führt eine oder mehrere Aktionen auf AWS -Ressourcen innerhalb eines bestimmten Zeitfensters aus. Sie können ein einziges Wartungsfenster mit Start- und Endzeiten definieren. Sie können mehrere Aufgaben angeben, die in diesem Wartungsfenster ausgeführt werden sollen. Verwenden Sie Maintenance Windows, eine Funktion von AWS Systems Manager, wenn Ihre Vorgänge mit hoher Priorität das Patchen Ihrer verwalteten Knoten, das Ausführen mehrerer Aufgabentypen auf Ihren Knoten während eines Aktualisierungszeitraums oder das Steuern, wann Aktualisierungsvorgänge auf Ihren Knoten ausgeführt werden können, beinhalten.

Die wichtigsten Anwendungsfälle für Maintenance Windows lauten wie folgt:

- Ausführen mehrerer Dokumente: Wartungsfenster können mehrere Aufgaben ausführen. Jede Aufgabe kann einen anderen Dokumenttyp verwenden. Dadurch können Sie komplexe Workflows mit unterschiedlichen Aufgaben innerhalb eines einzigen Wartungsfensters erstellen.
- Patching: Ein Wartungsfenster kann Patching-Unterstützung für alle verwalteten Knoten in einer einzelnen Region bieten, die mit einem bestimmten Tag oder einer bestimmten Ressourcengruppe versehen sind. Da das Patchen normalerweise das Herunterfahren von Knoten (z. B. das Entfernen von Knoten aus einem Load Balancer), das Patchen und die Nachbearbeitung (das Zurücksetzen von Knoten in die Produktion) umfasst, kann das Patchen als eine Reihe von Aufgaben innerhalb eines bestimmten Patch-Zeitfensters durchgeführt werden.

> **Note**
>
> Wenn Sie ein Wartungsfenster verwenden, ist Ihr Patching-Vorgang auf eine einzige Region in einem einzigen Konto beschränkt. Mit einer in Quick Setup erstellten Patch-Richtlinie, einer Funktion von Systems Manager, können Sie stattdessen das Patching für einige oder alle Konten und Regionen in einer in AWS Organizations erstellten Organisation konfigurieren. Weitere Informationen finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien.

- Fensteraktionen: Wartungsfenster können einen oder mehrere Aktionssätze innerhalb eines bestimmten Zeitfensters starten. Wartungsfenster werden nicht außerhalb dieses Fensters gestartet. Bereits gestartete Aktionen werden bis zum Abschluss fortgesetzt, auch wenn sie außerhalb des Zeitfensters abgeschlossen werden.

In der folgenden Tabelle werden die wichtigsten Features von State Manager und Maintenance Windows verglichen.

Funktion	State Manager	Maintenance Windows
AWS CloudFormation Integration	AWS CloudFormation -Vorlagen unterstützen -State ManagerZuordnungen.	AWS CloudFormation -Vorlagen unterstützen Wartungsfenster, Fensterziele und Fensteraufgaben.
Compliance	Jede State Manager-Zuordnung meldet die Compliance in Bezug auf den erforderlichen Status der Zielressource. Sie können das Compliance-Dashboard verwenden, um die gemeldete Compliance zu aggregieren und anzuzeigen.	Nicht zutreffend.
Integration der Konfigurationsverwaltung	State Manager unterstützt externe Lösungen für den	Nicht zutreffend.

Funktion	State Manager	Maintenance Windows
	Zielstatus wie Microsoft PowerShell Desired State Configuration (DSC), Ansible Playbooks und Chef Rezepte. Sie können State Manager-Zuordnungen verwenden, um zu testen, ob die Konfigurationsverwaltungs-Lösungen funktionieren, und um deren Konfigurationsänderungen auf Ihre Knoten anzuwenden, wenn Sie bereit sind.	
Dokumente	State Manager-Konfigurationen können als Richtliniendokumente (zum Sammeln von Bestandsinformationen), Automation-Runbooks, für AWS -Ressourcen wie Amazon Simple Storage Service (Amazon S3)-Buckets oder Systems-Manager-Befehlsdokumente (SSM-Dokumente) für verwaltete Knoten definiert werden.	Maintenance Windows-Konfigurationen können als Automatisierungsdokumente (mehrstufige Aktionen mit optionalen Genehmigungs-Workflows) oder SSM-Dokumente (gewünschter Status für verwaltete Knoten) definiert werden.

Funktion	State Manager	Maintenance Windows
Überwachung	State Manager überwacht Änderungen in der Konfiguration, der Zuordnung oder dem Zustand eines Knotens (z. B. neue Knoten, die online geschaltet werden). Wenn State Manager diese Änderungen erkennt, wird die gegebene Zuordnung erneut auf die Knoten angewendet, auf die diese Zuordnung ursprünglich abzielte.	Nicht zutreffend.
Prioritäten innerhalb von Aufgaben	Nicht zutreffend.	Aufgaben innerhalb eines Wartungsfensters können mit einer Priorität versehen werden. Alle Aufgaben mit derselben Priorität werden parallel ausgeführt. Aufgaben mit niedrigeren Prioritäten werden ausgeführt, nachdem Aufgaben mit höheren Prioritäten einen endgültigen Status erreicht haben. Es gibt keine Möglichkeit, Aufgaben bedingt auszuführen. Nachdem eine Aufgabe mit höherer Priorität den endgültigen Status erreicht hat, wird die nächste Prioritätsaufgabe unabhängig vom Status der vorherigen Aufgabe ausgeführt.

Funktion	State Manager	Maintenance Windows
Sicherheitskontrollen	State Manager unterstützt zwei Sicherheitskontrollen bei der Bereitstellung von Konfigurationen in einer großen Flotte. Sie können die maximale Nebenläufigkeit verwenden, um festzulegen, auf wie viele nebenläufige Knoten oder Ressourcen die Konfiguration angewendet werden soll. Sie können eine maximale Fehlerrate festlegen, die verwendet werden kann, um die State Manager-Verknüpfung zu pausieren, wenn eine bestimmte Anzahl oder ein bestimmter Prozentsatz von Fehlern in der gesamten Flotte auftritt.	Wartungsfenster unterstützen zwei Sicherheitskontrollen bei der Bereitstellung von Konfigurationen in einer großen Flotte. Sie können die maximale Nebenläufigkeit verwenden, um festzulegen, auf wie viele nebenläufige Knoten oder Ressourcen die Konfiguration angewendet werden soll. Sie können eine maximale Fehlerrate festlegen, die verwendet werden kann, um die Aktionen in einem Wartungsfenster zu pausieren, wenn eine bestimmte Anzahl oder ein Prozentsatz von Fehlern in der gesamten Flotte auftritt.

Funktion	State Manager	Maintenance Windows
Planung	Sie können State Manager-Verknüpfungen bei Bedarf, in einem bestimmten Cron-Intervall, mit einer bestimmten Rate oder einmal nach der Erstellung ausführen. Dies ist nützlich, wenn Sie den gewünschten Status Ihrer Ressourcen konsistent und zeitnah aufrechterhalten möchten. **Important** Cron-Ausdrücke für State Manager-Zuordnungen unterstützen das Monatsfeld nicht, wie `03` oder `MAR` für den Monat März. Wenn Sie monatliche oder vierteljährliche Konfigurationsupdates benötigen, kann ein Wartungsfenster Ihre Anforderungen am besten erfüllen. Weitere Informationen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.	Wartungsfenster unterstützen mehrere Planungsoptionen, einschließlich `at`-Ausdrücken (z. B. `"at(2021-07-07T13:15:30)"`), Cron- und Rate-Ausdrücke, Cron mit Offsets und Start- und Endzeiten für die Ausführung von Wartungsfenstern sowie Grenzzeiten, um anzugeben, wann die Planung innerhalb eines bestimmten Zeitfensters beendet werden soll.

Funktion	State Manager	Maintenance Windows
Targeting	State Manager-Zuordnungen können auf einen oder mehrere Knoten mithilfe der Knoten-ID, des Tags oder der Ressourcengruppe abzielen. State Manager kann auf alle verwalteten Knoten innerhalb eines bestimmten Kontos abzielen.	Wartungsfenster können auf einen oder mehrere Knoten mithilfe von Knoten-IDs, Tags oder Ressourcengruppen abzielen.

Funktion	State Manager	Maintenance Windows
Aufgaben innerhalb von Wartungsfenstern	Nicht zutreffend.	Wartungsfenster können eine oder mehrere Aufgaben unterstützen, bei denen jede Aufgabe auf ein bestimmte s Automation-Runbook oder eine Command-Dokumentat kion abzielt. Alle Aufgaben innerhalb eines Wartungsf ensters werden parallel ausgeführt, es sei denn, für unterschiedliche Aufgaben sind unterschiedliche Priorität en festgelegt. Insgesamt unterstützen die Wartungsfenster vier Aufgabentypen: • AWS Systems Manager Run Command-Befehle • AWS Systems Manager Automation-Workflows • AWS Lambda -Funktionen • AWS Step Functions - Aufgaben

Ähnliche Informationen

Die folgenden verwandten Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit diesem Service.

Preisgestaltung

Einige Systems Manager Funktionen sind gebührenpflichtig. Weitere Informationen finden Sie unter AWS Systems Manager Preise.

AWS Systems Manager-Dokumentationsbibliothek

AWS Systems Manager-Dokumentation – Zugriff auf die gesamte Benutzerdokumentation für Systems Manager, einschließlich AWS AppConfig, Vorfallmanager und AWS Systems Manager für SAP.

AWS re:Post

AWS re:Post – AWS-verwalteter Frage-und-Antwort-Dienst (F & A), der von Experten geprüfte Crowdsourcing-Antworten auf Ihre technischen Fragen bietet.

AWS-Blog und Podcast

Lesen Sie Blogbeiträge über Systems Manager in der Kategorie AWS-Management-Tools und andere Beiträge, die mit `#Systems Manager` getaggt sind.

Service Quotas

Überprüfen Sie Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz. Wenn nicht anders angegeben, gilt jedes Kontingent für eine Region in einem AWS-Konto.

Referenz zur Serviceautorisierung für Systems Manager

In der AWS Referenz zur Serviceautorisierung finden Sie Informationen zu den Aktionen, Ressourcen und Bedingungskontextschlüsseln, die Sie in AWS Identity and Access Management (IAM)-Richtlinien für Systems Manager verwenden können.

AWS Systems Manager Service Level Agreement

Das AWS Systems Manager Service Level Agreement (SLA) ist eine Richtlinie, die die Verwendung von Systems Manager regelt und für jedes AWS-Konto, das Systems Manager verwendet, separat gilt.

Allgemeine AWS-Ressourcen

Die folgenden allgemeinen Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit AWS.

- Kurse und Workshops – Links zu rollenbasierten und speziellen Kursen sowie Übungen im Selbststudium zur Verbesserung Ihrer AWS-Kompetenzen und Erweiterung Ihrer praktischen Erfahrung.
- AWS-Entwicklerzentrum – Entdecken Sie Tutorials, laden Sie Tools herunter und erfahren Sie mehr über Veranstaltungen für AWS-Entwickler.
- AWS-Entwickler-Tools – Links zu Entwickler-Tools, SDKs, IDE-Toolkits und Befehlszeilen-Tools für die Entwicklung und Verwaltung von AWS-Anwendungen.
- Ressourcenzentrum für die ersten Schritte – Erfahren Sie, wie Sie Ihr AWS-Konto einrichten, der AWS-Community beitreten und Ihre erste Anwendung starten.
- Praktische Tutorials – Folgen Sie den step-by-step Tutorials, um Ihre erste Anwendung auf zu startenAWS.
- AWS Whitepaper – Links zu einer umfangreichen Liste technischer AWS-Whitepaper zu Themen wie Architektur, Sicherheit und Wirtschaftlichkeit. Diese Whitepaper wurden von AWS-Lösungsarchitekten und anderen technischen Experten verfasst.
- AWS Support-Center – Hub für die Erstellung und Verwaltung Ihrer AWS Support-Fälle. Stellt darüber hinaus Links zu weiteren nützlichen Ressourcen bereit, beispielsweise Foren, häufig gestellten technischen Fragen, Status der Service-Integrität und AWS Trusted Advisor.
- AWS Support – Die primäre Webseite für Informationen zu AWS Support, einem one-on-one, reaktionsschnellen Support-Kanal, der Sie beim Erstellen und Ausführen von Anwendungen in der Cloud unterstützt.
- Kontakt – Zentraler Kontaktpunkt für Fragen zu AWS-Abrechnung, Konten, Ereignissen Missbrauch und anderen Problemen.
- Nutzungsbedingungen für die AWS-Website – Detaillierte Informationen zu unseren Copyright- und Markenbestimmungen, Ihrem Konto, den Lizenzen und anderen Themen.

Dokumentverlauf

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation seit der letzten Version von AWS Systems Manager beschrieben. Für Benachrichtigungen über Aktualisierungen dieser Dokumentation können Sie einen RSS-Feed abonnieren.

- API-Version: 2014-11-06

Änderung	Beschreibung	Datum
Update: Regionale Verfügbarkeit des `/aws/service/global-infrastructure` Parameterpfads	Wir haben geklärt, aus welchen kommerziellen Regionen der `/aws/service/global-infrastructure` öffentliche Parameterpfad abgefragt werden kann und wie eine Abfrage für den Pfad ausgeführt wird, wenn Sie in einem anderen kommerziellen Bereich arbeiten. AWS-Region Weitere Informationen finden Sie unter Aufrufen öffentlicher Parameter für AWS Dienste, Regionen, Endpunkte, Availability Zones, Local Zones und Wavelength Zones.	12. Juni 2024
Neu: Kapitel mit Codebeispielen	Ein neues Kapitel, Codebeispiele für Systems Manager mit AWS SDKs, enthält Beispiele in verschiedenen SDK-Sprachen für die Arbeit mit dem Systems Manager Manager-Dienst.	8. Mai 2024

Änderungen an der `ec2messages:*` Endpunktunterstützung	Für den AWS-Regionen Start im Jahr 2024 oder später werden die `ec2messages:*` Endpunkte nicht unterstützt, um SSM Agent Status- und Ausführungsinformationen zurück an den Systems Manager Manager-Dienst zu senden. Konten in diesen Regionen müssen verwenden `ssmmessages:*` . In Regionen, die vor 2024 gestartet wurden, `ec2messages:*` werden beide `ssmmessages:*` weiterhin unterstützt. Wir empfehlen jedoch, jetzt nur den `ssmmessages:*` Endpunkt (Amazon Message Gateway Service) zu verwenden. Sie können derzeit problemlos `ec2messages:*` Berechtigungen aus Ihren Richtlinien entfernen. Weitere Informationen finden Sie unter Arbeiten mit SSM Agent und agentenbezogene API-Operationen (ssmmessages- und ec2messages-Endpunkte).	3. Mai 2024

Zusätzliche Laufzeiten sind für die Ausführung von Skripten in Automation-Runbooks verfügbar	Die `aws:executeScript` Aktion unterstützt jetzt die Python-Laufzeiten 3.9, 3.10 und 3.11. Weitere Informationen zur Verwendung dieser Aktion finden Sie unter. `aws:executeScript`	23. April 2024
Support für die Versionen 8.8 und 8.9: AlmaLinuxOracle Linux, und Rocky Linux	Systems Manager unterstützt jetzt zusätzlich zu früheren 8.x-Versionen die Versionen 8.8 und 8.9 von AlmaLinux Rocky Linux, und. Oracle Linux Eine vollständige Liste der unterstützten Betriebssysteme und Versionen finden Sie unter Unterstützte Betriebssysteme für Systems Manager.	22. April 2024

Patch Manager: Wechseln Sie zum Patch-Status 'INSTALLED_PENDING_REBOOT'	Bisher konnten nur Patches, die von installiert wurden, als markiert werden. Patch Manager `INSTALLED_PENDING_REBOOT` Patches, die außerhalb von installiert Patch Manager wurden, erhielten diesen Status nie. `INSTALLED_PENDING_REBOOT` Kann jetzt auf jeden Patch angewendet werden, der seit dem letzten Neustart auf einen verwalteten Knoten angewendet wurde. Dazu gehören Patches, die Patch Manager mit der ausgewählten `NoReboot` Option installiert wurden, und Patches, die außerhalb oder Patch Manager nach dem letzten Neustart des Knotens installiert wurden. Eine Beschreibung aller Werte für den Patch Manager Patchstatus finden Sie unter Grundlegendes zu den Werten für den Status der Patch-Konformität.	16. April 2024
Support für RHEL 8.9 und 9.3	Systems Manager, einschließlichPatch Manager, unterstützt jetzt die Red Hat Enterprise Linux (RHEL) Versionen 8.9 und 9.3 zusätzlich zu den früheren Versionen 8.x und 9.x.	26. März 2024

Themen-Update: AWS verwaltete Richtlinien für AWS Systems Manager	Das Thema AWS Verwaltete Richtlinien für AWS Systems Manager enthält Informationen zu den vier verwalteten Richtlinien für Systems Manager, die seit dem 12. März 2021 eingeführt oder aktualisiert wurden. Wir haben diesem Thema einen Abschnitt mit Informationen zu den 12 anderen verwalteten Richtlinien zur Verwendung mit Systems Manager hinzugefügt, die vor diesem Datum erstellt oder zuletzt aktualisiert wurden. Einzelheiten finden Sie unter Zusätzliche verwaltete Richtlinien für Systems Manager.	1. März 2024

Parameter Storeunterstützt jetzt kontoübergreifendes Teilen	Sie können jetzt erweiterte Parameter sicher und effizient innerhalb Ihrer Organisation AWS-Konten oder innerhalb Ihrer AWS Organisation teilen, indem Sie die gemeinsame Nutzung von Ressourcen einrichten. Die gemeinsame Nutzung von Ressourcen ermöglicht es Ihnen, das Anwendungskonfigurationsmanagement zu zentralisieren und den betrieblichen Aufwand zu reduzieren, der durch die gemeinsame Nutzung der Parameter mit jedem einzelnen Konto, das Sie besitzen, entsteht. Parameter können über die Parameter Store Konsole, die Konsole oder die von mehreren AWS RAM Konten gemeinsam genutzt werden. AWS CLI Weitere Informationen finden Sie unter Arbeiten mit gemeinsam genutzten Parametern.	21. Februar 2024

Verbesserung der Automatisierungsaktionen	Sie können jetzt die `isCritical` Eigenschaften `onFailure` und mit der `aws:approve` Aktion verwenden. Weitere Informationen zur `aws:approve` Aktion finden Sie unter aws:approve — Eine Automatisierung für die manuelle Genehmigung pausieren.	12. Februar 2024
Zusätzliche Unterstützung für Betriebsversionen für Patch Manager	Wir haben die Liste der unterstützten Betriebssystemversionen für erweitert Patch Manager. Support wurde für Folgendes hinzugefügt: • Debian Server11.x und 12.x • macOS14,0 (Sonoma) • SUSE Linux Enterprise Server() 15,5 SLES • Ubuntu Server23,04	04. Januar 2024
Automatische SSM Agent-Updates mithilfe der Application Manager-Konsole konfigurieren	Sie können jetzt die Application Manager-Konsole verwenden, um SSM Agent-Updates für Ihre Anwendungs-Instances zu automatisieren. Weitere Informationen finden Sie unter Arbeiten mit Ihren Anwendungs-Instances.	21. Dezember 2023

Aktualisierter Prozess für die Registrierung von Maschinen, die nicht zu Amazon EC2 gehören, in Hybrid- und Multi-Cloud-Umgebungen	Systems Manager bietet jetzt die `ssm-setup-cli` als Unterstützung bei der Registrierung von Maschinen, die nicht zu Amazon Elastic Compute Cloud (Amazon EC2) gehören, in Hybrid- und Multi-Cloud-Umgebungen. Weitere Informationen finden Sie unter So installieren Sie den SSM Agent auf Hybrid-Linux-Knoten und So installieren Sie den SSM Agent auf Hybrid-Windows-Knoten.	20. Dezember 2023
Verwalten von Amazon-EBS-Volumes mit Fleet Manager	Sie können jetztFleet Manager, eine Funktion von, verwenden AWS Systems Manager, um Amazon Elastic Block Store-Volumes auf Ihren verwalteten Instances zu verwalten. Sie können beispielsweise ein EBS-Volume initialisieren, eine Partition formatieren und das Volume mounten, um es für die Nutzung verfügbar zu machen. Weitere Informationen finden Sie unter EBS-Volumeverwaltung.	14. Dezember 2023

Erweiterung des Session Manager-Plug-ins	Unterstützung für die Übergabe einer StartSession API-Antwort als Umgebungsvariable an hinzugefügt session-manager-plugin.	4. Dezember 2023
Neue visuelle Designerfahrung für Automation-Runbooks	Sie können Runbooks jetzt mithilfe einer visuellen Designerfahrung erstellen und bearbeiten, die von Systems Manager Automation entwickelt wurde. Das visuelle Designerlebnis bietet eine drag-and-drop Low-Code-Oberfläche, sodass Sie Runbooks einfacher erstellen und bearbeiten können. Weitere Informationen finden Sie unter Visuelle Designerfahrung für Automation-Runbooks.	26. November 2023

Neue Systems-Manager-Automation-Aktionen, Datenelement- und Funktionserweiterungen für Runbooks	Mit der `aws:loop`-Aktion können Sie jetzt mehrere Aktionen in einem Runbook wiederholen. Diese neue Aktion unterstützt Loops im Stil `do while` und `for each`. Darüber hinaus können Sie mithilfe des neuen Variablen-Datenelements Werte dynamisch im Kontext eines Runbooks definieren, referenzieren und aktualisieren. Verwenden Sie die neue `aws:updateVariable` -Aktion, um den Wert einer Variablen in Ihrem Runbook zu aktualisieren. Automation hat auch Unterstützung für dynamische Datentypkonvertierungen für Ausgaben hinzugefügt. Das heißt, wenn der Wert einer Ausgabe nicht dem von Ihnen angegebenen Datentyp entspricht, versucht Automation, den Datentyp zu konvertieren. Wenn der zurückgegebene Wert beispielsweise ein `Integer` ist, der angegebene `Type` jedoch ein `String` ist, ist der endgültige Ausgabewert ein `String`-Wert. Schließlich unterstützt Automation jetzt JSONPath-Filterausdrücke für Selektoren. Weitere Informati	17. November 2023

	onen finden Sie unter den folgenden Themen: • aws:loop – Über Schritte in einer Automatisierung interieren • aws:UpdateVariable – Aktualisiert einen Wert für eine Runbook-Variable • Datenelemente und Parameter – Datenelemente der obersten Ebene • Verwenden von Aktionsausgaben als Eingaben. • Verwenden von JSONPath in Runbooks.	
Die Regionsunterstützung für Remote Desktop Protocol (RDP)-Verbindungen wurde aktualisiert	Fleet Manager Remote Desktop, das von NICE DCV unterstützt wird, bietet Ihnen eine sichere Verbindung zu Ihren Windows Server-Instances direkt von der Systems-Manager-Konsole aus. Die folgenden drei zusätzlichen Regionen wurden für Fleet Manager-Remotedesktopverbindungen aktiviert: • Afrika (Kapstadt) (af-south-1) • Asien-Pazifik (Jakarta) (ap-southeast-3) • Israel (Tel Aviv) (il-central-1)	15. November 2023

Patch Manager: Erweiterte Unterstützung von Betriebssystemversionen für RHEL und macOS	Patch Manager unterstützt jetzt die folgenden zusätzlichen Betriebssystemversionen: • Red Hat Enterprise Linux: Version 8.8 • macOS: 11.5–11.7 (Big Sur) • macOS: 12.0–12.6 (Monterey) • macOS: 13.0–13.5 (Ventura)	23. Oktober 2023
Neue OpsCenter-API – OpsItem löschen	OpsCenter bietet jetzt die API „OpsItem löschen“ zum Löschen einzelner OpsItems. Weitere Informationen finden Sie unter DeleteOpsArtikel in der AWS Systems Manager API-Referenz.	20. Oktober 2023
Neuer Quick Setup Konfigurationstyp: SSM Agent Updates für die gesamte Organisation	Der neue Konfigurationstyp Standard-Host-Management-Konfiguration ermöglicht es einem Organisationsadministrator, wie unter definiert AWS Organizations, automatische Prüfungen und SSM Agent Aktualisierungen aller EC2-Instances in den Konten und Regionen der Organisation zu veranlassen. Weitere Informationen finden Sie unter Standard-Host-Verwaltung für eine Organisation.	16. Oktober 2023

Neues Titel- und Beschreibungsformat für „OpsItems Erstellt von CloudWatch Application Insights“	Der Titel und die Beschreibung von OpsItems Created by CloudWatch Application Insights werden am 16. Oktober 2023 in ein verbessertes Format geändert. Das neue Format finden Sie unter Amazon CloudWatch Application Insights.	29. September 2023

Support für mehrere Bildschirmauflösungen in Fleet Manager-RDP-Verbindungen	Wenn Sie sich über die Option Remote-Desktop-Protokoll (RDP) in Fleet Manager mit Windows Server verwalteten Knoten verbinden, können Sie jetzt die Bildschirmauflösung wählen. Bisher wurde für alle Verbindungen eine feste Auflösung von 720P (1366 x 768) verwendet. Sie können jetzt für jede Verbindung aus den folgenden Optionen wählen: • Automatisch anpassen (bestimmt anhand der erkannten Bildschirmgröße die optimale Auflösung) • 1920 x 1080 • 1400 x 900 • 1366 x 768 • 800 x 600 Weitere Informationen finden Sie unter Über Remote Desktop mit einem verwalteten Knoten verbinden.	22. September 2023

Neues Thema: Zufällige Patch-Baseline-IDs bei Patch-Richtlinien-Operationen	Wir haben Inhalte hinzugefügt, die beschreiben, wie Quick Setup-Patch-Richtlinien den `BaselineOverride` -Parameter im SSM-Command-Dokument für `AWS-RunPatchBaseline` verwenden, um bei jeder Ausführung einer Patch-Richtlinien-Operation zufällige IDs für Patch-Baselines zu erzeugen. Weitere Informationen finden Sie unter Zufällige Patch-Baseline-IDs in Patch-Richtlinien-Operationen.	22. September 2023
Ein neuer betrieblicher Einblick in die Verwaltung von OpsItems	OpsCenterenthält jetzt einen Einblick in betriebliche Abläufe mit der Bezeichnung Ressourcen, die am meisten generieren OpsItems. Ein solcher Einblick wird generiert, wenn für eine AWS Ressource mehr als 10 geöffnet sindOpsItems. Verwenden Sie diesen Einblick, um problematische Ressourcen zu lokalisieren. Verwenden Sie das `AWS-BulkResolveOpsItems` -Runbook aus einem Einblick heraus, um OpsItems im Zusammenhang mit einer Ressource schnell zu lösen. Weitere Informationen finden Sie unter Analysieren betrieblicher Einblicke zur Reduzierung von OpsItems.	22. September 2023

Öffentlicher GPG-Schlüssel aktualisiert	Ein neuer öffentlicher Schlüssel wurde erstellt, um die Signatur von SSM Agent zu verifizieren. Weitere Informationen finden Sie unter Signatur von SSM Agent verifizieren.	5. September 2023
Support für zusätzliche Versionen von AlmaLinux , Oracle LinuxRHEL, und hinzugefügt Rocky Linux	Die Listen der unterstützten Betriebssysteme für AWS Systems Manager und Patch Manager wurden aktualisiert, um die Unterstützung der folgenden zusätzlichen Betriebssystemversionen widerzuspiegeln: • AlmaLinux: 9.2 • Oracle Linux: 8.7 und 9.2 • Red Hat Enterprise Linux(RHEL): 8.7, 9.1 und 9.2 • Rocky Linux: 8.6 und 8.7, 9.0–9.2	30. August 2023

OpsCenter unterstützt nun die Markdown-Formatierung im OpsItem-Beschreibungsfeld.	OpsCenter unterstützt jetzt die Markdown-Formatierung im OpsItem-Beschreibungsfeld. Die folgenden Typen der Markdown-Formatierung werden unterstützt: • Paragraphen • Zeilenabstand • Horizontale Linien • Überschriften • Textformatierung • Links • Listen Weitere Informationen finden Sie unter Verwenden von Markdown in der Konsole im Handbuch Erste Schritte mit dem Handbuch AWS Management Console Erste Schritte.	18. August 2023

Neue Versionen der AWS Lambda-Erweiterung Parameters and Secrets	Neue Versionen der AWS Parameters and Secrets Lambda Extension sind jetzt verfügbar. Darüber hinaus wurde Erweiterungsunterstützung für die Regionen Asien-Pazifik (Melbourne) (ap-southeast-4) und Israel (Tel Aviv) (il-central-1) hinzugefügt (nur für x86_64- und x86-Architekturen). Weitere Informationen finden Sie unter Verwenden von Parameter Store Parametern in AWS Lambda Funktionen.	16. August 2023

Update: Es wurden Informationen zu den erforderlichen Berechtigungen für Patch-Richtlinien-Buckets von Quick Setup hinzugefügt	Wenn Sie eine Patch-Richtlinie erstellen, erstellt Quick Setup einen Amazon-S3 Bucket, der eine Datei mit dem Namen `baseline_overrides.json` enthält. In dieser Datei werden Informationen zu den Patch-Baselines gespeichert, die Sie für Ihre Patch-Richtlinie angegeben haben. Bei der Konfiguration der Patch-Richtlinie haben Sie die Möglichkeit, das Kontrollkästchen Erforderliche IAM-Richtlinien zu vorhandenen Instance-Profilen hinzufügen, die an Ihre Instances angehängt sind, zu aktivieren. Wenn Sie diese Option nicht auswählen, müssen Sie bestimmten Ressourcen manuell Berechtigungen für den Zugriff auf diesen Bucket gewähren. Andernfalls schlagen Ihre Richtlinienoperationen möglicherweise fehl. Weitere Informationen finden Sie unter den folgenden Themen: • Berechtigungen für den S3-Bucket mit der Patch-Richtlinie • Problem: Fehler „InvokePatchBaselineOperation : Zugriff verweigert“ oder	6. Juli 2023

	Fehler „Datei kann nicht von S3 heruntergeladen werden“ für `baseline_overrides.json`	
Verwenden Sie Quick Setup, um OpsCenter für die Mehrkontenverwaltung von OpsItem zu konfigurieren	Quick Setup für OpsCenter hilft Ihnen dabei, die folgenden Aufgaben für die Verwaltung von OpsItems über mehrere Konten hinweg zu erledigen: • Angeben des delegierten Administratorkontos • Erforderliche AWS Identity and Access Management (IAM-) Richtlinien und Rollen erstellen • Angabe einer AWS Organizations Organisation oder einer Teilmenge von Mitgliedskonten, die ein delegierter Administrator kontenübergreifend verwalten kann OpsItems Weitere Informationen finden Sie unter (Optional) Konfigurieren von OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe vonQuick Setup.	19. Juni 2023

Aktualisieren Sie Amazon-EC2-Startagenten mit Quick Setup	Sie können jetzt zulassen, dass Systems Manager alle 30 Tage nach einer neuen Version des auf Ihrer Instance installierten Startagenten sucht. Wenn eine neue Version verfügbar ist, aktualisiert Systems Manager den Agenten auf Ihrer Instance. Weitere Informationen finden Sie unter Quick Setup-Host-Verwaltung.	19. Juni 2023
Patch Manager unterstützt jetzt Ubuntu Server 22.04 LTS	Sie können jetzt Patch Manager verwenden, um Ubuntu Server 22.04. LTS-Knoten zu patchen. Wie andere unterstützte Versionen von Ubuntu Server verwendet auch Version 22.04 LTS die verwaltete Patch-Baseline. AWS `AWS-UbuntuDefaultPatchBaseline`	15. Mai 2023

Systems Manager unterstützt jetzt AlmaLinux unter anderem Patch Manager	Sie können jetzt Systems Manager verwenden, um Knoten der Versionen AlmaLinux 8.3-8.7; 9.0-9.1 zu verwalten. Viele der Regeln, die für RHEL 8 beim Patchen gelten, gelten auch für. AlmaLinux AlmaLinuxverwendet das neue`AWS-DefaultAlmaLinuxPatchBaseline` . Weitere Informationen finden Sie unter den folgenden Themen: • Manuell SSM Agent auf AlmaLinux Instanzen installieren • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • So funktionieren Patch-Basisregeln für AlmaLinux RHEL, und Rocky Linux.	8. Mai 2023
Stellen Sie den EC2Launch-v2-Agenten mit Quick Setup bereit	Sie können den EC2Launch -v2-Agenten jetzt mit Quick Setup bereitstellen. Weitere Informationen finden Sie unter Bereitstellen von Distributor-Paketen mit Quick Setup.	13. April 2023

Systems Manager unterstützt jetzt Amazon Linux 2023	Systems Manager unterstützt jetzt den neuen EC2-Instance-Typ von Amazon Linux 2023 (AL2023), einschließlich der Unterstützung von Patch Manager-Vorgängen. Viele der Regeln für Patches, die für Amazon Linux 2 gelten, gelten auch für Amazon Linux 2023. (Patch Manager unterstützt auch weiterhin die Vorabversion Amazon Linux 2022.) Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • So funktionieren Patch-Basisregeln auf Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 und Amazon Linux 2023	23. März 2023

Überarbeitetes Einrichten von Inhalten für Amazon-EC2-Instances	Wir haben die Einrichtungsinhalte für Amazon EC2-Instances überarbeitet. Es wird nun empfohlen, die neu veröffentlichte Standardkonfiguration für die Host-Verwaltung für Instance-Berechtigungen zu verwenden. Weitere Informationen finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.	15. Februar 2023
Automatische Instance-Verwaltung mit der Standardkonfiguration für die Host-Verwaltung	Mit Systems Manager können Sie jetzt Amazon-EC2-Instances in einer gesamten AWS-Region automatisch verwalten. Weitere Informationen finden Sie unter Standardkonfiguration für die Host-Verwaltung.	15. Februar 2023

Hinzufügen von SSM-Dokumenten zu Ihren Favoriten	Um Ihnen das Auffinden häufig genutzter SSM-Dokumente zu erleichtern, können Sie jetzt Dokumente zu Ihren Favoriten hinzufügen. Sie können bis zu 20 Dokumente pro Dokumenttyp, pro AWS-Konto und zu Ihren Favoriten hinzufügen AWS-Region. Sie können Ihre Favoriten über die Dokumenten-Konsole von Systems Manager auswählen, ändern und anzeigen. Weitere Informationen finden Sie unter Hinzufügen von Dokumenten zu Ihren Favoriten.	07. Februar 2023
Implementieren von Änderungskontrollen für die Automatisierung mit Change Calendar	Durch die Integration von Automation mit Change Calendar können Sie jetzt Änderungskontrollen für alle Automatisierungen in Ihrem AWS-Konto implementieren. Weitere Informationen finden Sie unter Implementierung von Änderungskontrollen für die Automatisierung.	24. Januar 2023

Neuer Change Manager-Genehmigungs-Workflow	Der Change Manager-Genehmigungs-Workflow unterstützt jetzt Genehmigungen pro Ebene anstatt Genehmigungen pro Zeile. Bisher musste jeder Genehmiger, den Sie einer Genehmigungsebene hinzugefügt haben, eine Änderungsanfrage genehmigen. Andernfalls wurde das Level nicht genehmigt. Nun können Sie festlegen, wie viele Genehmigungen pro Ebene erforderlich sind, und können so viele oder mehr Genehmiger hinzufügen. Beispielsweise können Sie drei Genehmigungen für eine Ebene anfordern, aber bis zu fünf Genehmiger angeben. Die Genehmigungen von drei dieser Genehmiger sind ausreichend, um die Ebene zu genehmigen. Weitere Informationen finden Sie unter Über Genehmigungen in Ihren Änderungsvorlagen.	23. Januar 2023

Neu: Konfigurieren von Patching für eine gesamte Organisation mithilfe einer Patch-Richtlinie in Quick Setup	Mit Quick Setup, einer Funktion von Systems Manager, können Sie jetzt Patch-Richtlinien erstellen, die von Patch Manager unterstützt werden. Eine Patch-Richtlinie definiert den Zeitplan und die Patch-Baseline, die beim automatischen Patching Ihrer verwalteten Knoten verwendet werden sollen. Mit einer einzelnen Patch-Richtlinienkonfiguration können Sie Patches für alle Konten in allen Regionen in Ihrer Organisation, nur für die von Ihnen ausgewählten Konten und Regionen oder für ein einzelnes Konto-Region-Paar definieren. Weitere Informationen finden Sie unter den folgenden Themen. • Verwenden von Quick Setup Patch-Richtlinien • Automatisieren von unternehmensweitem Patching mithilfe einer Quick Setup-Patch-Richtlinie	22. Dezember 2022

Application Manager lässt sich in Amazon EC2 integrieren, um Informationen zu Ihren Instances im Kontext einer Anwendung anzuzeigen.	Application Manager zeigt Instance-Status, -Status und den Zustand von Amazon EC2 Auto Scaling für eine ausgewählte Anwendung in einem grafischen Format an. Die Registerkarte Instances enthält auch eine Tabelle mit den folgenden Informationen für jede Instance in Ihrer Anwendung. • Instance-Status (Ausstehend, Angehalten, Wird ausgeführt, Beendet) • Ping-Status für SSM Agent • Status und Name des letzten Systems-Manager-Automation-Runbooks, das auf der Instance verarbeitet wurde • Eine Anzahl von Amazon CloudWatch Logs-Alarmen pro Bundesstaat. • `ALARM` – Die Metrik oder der Ausdruck liegt außerhalb des festgelegten Schwellenwerts. • `OK` – Die Metrik oder der Ausdruck liegt innerhalb des festgelegten Schwellenwerts. • `INSUFFICIENT_DATA` – Der Alarm wurde soeben gestartet; die Metrik ist	22. Dezember 2022

	nicht verfügbar oder es sind nicht genügend Daten verfügbar, damit die Metrik den Alarmstatus bestimmen kann. • Zustand der Auto-Scaling-Gruppe für die übergeordneten und einzelnen Auto-Scaling-Gruppen	
Planen des Starts und des Beendens Ihrer Amazon-EC 2-Instances mithilfe von Quick Setup	Sie können jetzt die Resource-Scheduler-Lösung bereitstellen, um das Starten und Beenden Ihrer Amazon-EC 2-Instances mit Quick Setup zu automatisieren. Weitere Informationen finden Sie unter Resource Scheduler.	19. Dezember 2022

OpsCenter unterstützt jetzt das kontenübergreifende Arbeiten mit OpsItems	OpsCenter unterstützt das Arbeiten mit OpsItems von einem Verwaltungskonto (entweder einem AWS Organizations -Verwaltungskonto oder einem von Systems Manager delegierten Administratorkonto) und einem Mitgliedskonto während einer Sitzung. Nach der Konfiguration können Benutzer die folgenden Arten von Aktionen ausführen: • Erstellen, Anzeigen und Aktualisieren von OpsItems in einem Mitgliedskonto • Sehen Sie sich detaillierte Informationen zu den AWS Ressourcen an, die OpsItems in einem Mitgliedskonto angegeben sind • Starten von Systems-Manager-Automation-Runbooks zur Behebung von Problemen mit AWS -Ressourcen in einem Mitgliedskonto Weitere Informationen finden Sie unter Einrichten von OpsCenter für das kontoübergreifende Arbeiten mit OpsItems.	16. November 2022

Verfolgen Sie die Details von Change Manager Änderungsanfragen mithilfe von AWS CloudTrail Lake	Sie können jetzt einen Ereignisdatenspeicher in AWS CloudTrail Lake verwenden, um Details zu den Änderungsanforderungen zu erfassen und zu überprüfen, die Change Manager für Ihre Organisation oder Ihr Konto eingereicht wurden. Zu diesen Informationen gehören überprüfbare Details über die Benutzeridentität, die die Änderungsanforderung erstellt hat, die IP-Adresse, von der aus die Anfrage gestellt wurde, den AWS-Regionen Ort, an dem die Änderungen vorgenommen wurden, die Zielressourcen und vieles mehr. Weitere Informationen finden Sie unter Überwachung der Ereignisse Ihrer Änderungsanfragen und Überprüfen von Details, Aufgaben und Zeitplänen für Änderungsanfragen.	11. November 2022

Zusätzliche Aufgabensteuerungen von Systems Manager Automation mithilfe von CloudWatch Alarmen	Sie können jetzt mithilfe von CloudWatch Alarmen zusätzliche Steuerung implementieren, wenn Sie Automatisierungen über mehrere Konten und Regionen hinweg ausführen. Indem Sie eine Metrik oder einen zusammengesetzten CloudWatch Alarm auf eine Automatisierung anwenden, können Sie anhand der von Ihnen definierten Metriken steuern, wann eine Automatisierung beendet wird. Weitere Informationen zum Anwenden eines CloudWatch Alarms auf eine Automatisierung, die über mehrere Konten und Regionen läuft, finden Sie unter Ausführen einer Automatisierung in mehreren Regionen und Konten (Konsole)	9. November 2022

Aktualisiert: Parameter Store 'Parameter in AWS Lambda Funktionen verwenden'	Wir haben zusätzliche Informationen bereitgestellt, die Ihnen helfen sollen, die Lambda-Erweiterung Parameters and Secrets zu verwenden, um Parameterwerte abzurufen und sie für die future Verwendung in Lambda-Funktionen AWS zwischenzuspeichern. Durch die Verwendung der Lambda-Erweiterung können Sie Ihre Kosten senken, indem Sie die Anzahl der API-Aufrufe auf Parameter Store reduzieren. Weitere Informationen finden Sie unter Verwenden von Parameter Store Parametern in AWS Lambda Funktionen.	25. Oktober 2022

Zusätzliche Systems Manager Manager-Aufgabenst euerungen mithilfe von CloudWatch Alarmen	Sie können jetzt mithilfe von CloudWatch Alarmen zusätzliche Steuerelemente bei der Ausführung von Automatisierungen und Befehlen implementieren. Ein CloudWatch Alarm kann auch zu einer Automatisierung oder einem Befehl hinzugefü gt werden, wenn er für eine State Manager Zuordnung s- oder Wartungsfensterauf gabe registriert ist. Indem Sie einen zusammengesetzten CloudWatch Alarm auf eine Automatisierung oder einen Befehl anwenden, können Sie anhand der von Ihnen definiert en Metrik steuern, wann eine Automatisierung oder ein Befehl beendet wird. Weitere Informationen zum Anwenden eines CloudWatch Alarms auf eine Automatisierung oder einen Befehl finden Sie in den folgenden Verfahren: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • So funktionieren Patch-Basisregeln auf Amazon Linux 1, Amazon Linux 2 und Amazon Linux 2022.	26. September 2022

Zusätzliche Systems Manager Manager-Aufgabenst euerungen mithilfe von CloudWatch Alarmen	Sie können jetzt mithilfe von CloudWatch Alarmen zusätzliche Steuerelemente bei der Ausführung von Automatisierungen und Befehlen implementieren. Ein CloudWatch Alarm kann auch zu einer Automatisierung oder einem Befehl hinzugefü gt werden, wenn er für eine State Manager Zuordnung s- oder Wartungsfensterauf gabe registriert ist. Indem Sie einen zusammengesetzten CloudWatch Alarm auf eine Automatisierung oder einen Befehl anwenden, können Sie anhand der von Ihnen definiert en Metrik steuern, wann eine Automatisierung oder ein Befehl beendet wird. Weitere Informationen zum Anwenden eines CloudWatch Alarms auf eine Automatisierung oder einen Befehl finden Sie in den folgenden Verfahren: • Ausführen einer einfachen Automatisierung • Ausführen von Befehlen über die Konsole • Erstellen einer Zuordnung • Einem Wartungsfenster Aufgaben zuweisen	26. September 2022

Klärung der Anforderungen für Advanced-Instances-Kontingente	Basierend auf Kundenfeedback haben wir die Szenarien geklärt, in denen Sie die Advanced-Instances-Kontingente in Instance-Kontingente konfigurieren aktivieren müssen.	21. September 2022
Stellen Sie den CloudWatch Amazon-Agenten bereit mit Quick Setup	Sie können den CloudWatch Amazon-Agenten jetzt mithilfe von bereitstellenQuick Setup. Weitere Informationen finden Sie unter Bereitstellen von Distributor-Paketen mit Quick Setup.	20. September 2022

Der Schlüssel PatchGroup " wird jetzt für Patch-Gruppen unterstützt, wenn EC2-Instance-Metadaten zulässig sind	Wenn Sie Tags in EC2-Instance-Metadaten zulassen, dürfen die von Ihnen erstellten Tag-Schlüssel keine Leerzeichen enthalten. Bisher hinderte dies Kunden daran, einige ihrer EC2-Instances zu Patch-Gruppen in Patch Manager hinzuzufügen, da der Tag-Schlüssel `Patch Group` auf die Instances angewendet werden musste. Patch Manager unterstützt jetzt sowohl `Patch Group` (mit einem Leerzeichen) als auch `PatchGroup` (ohne Leerzeichen) als Tag-Schlüssel zum Identifizieren von Instances für eine Patch-Gruppe. EC2-Instances, bei denen Tags in Instance-Metadaten zulässig sind, können jetzt zu Patch-Gruppen in Patch Manager hinzugefügt werden. Informationen zu Patch-Gruppen finden Sie unter Über Patch-Gruppen.	31. August 2022

Neues Thema: „So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet“	In Patch-Baselines, die von verwaltet werden AWS, werden neue Patches 7 Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt. In benutzerdefinierten Patch-Baselines, die Sie erstellen, können Sie optional angeben, wie viele Tage nach ihrer Veröffentlichung oder Aktualisierung gewartet werden sollen, um ihre Installation automatisch zu genehmigen. Für Amazon Linux 1 und Amazon Linux 2 beeinflussen verschiedene Faktoren, wie die neuesten Veröffentlichungs- und Aktualisierungstermine berechnet werden. Um unerwartete Ergebnisse bei der Auswahl von Verzögerungen bei der automatischen Genehmigung zu vermeiden, werden diese Faktoren im Thema So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet erläutert.	24. August 2022

Aktualisierter Inhalt: Patchen eines AMI und Aktualisieren einer Auto-Scaling-Gruppe	Wir haben die exemplarische Vorgehensweise zum Aktualisieren von AMIs für Auto-Scaling-Gruppen aktualisiert, um Startvorlagen anstelle von Startkonfigurationen zu verwenden. Darüber hinaus haben wir die neuesten Automatisierungsaktionen und Runtimes in den Runbook-Inhalten implementiert.	22. Juni 2022

Change Manager: Verhindern Sie, dass Benutzer automatisch genehmigbare Anfragen erstellen	Sie können Änderungsvorlagen in Change Manager so konfigurieren, dass automatische Genehmigungen unterstützt werden. Dies bedeutet, dass Benutzer mit den erforderlichen IAM-Berechtigungen die Änderungsanforderung starten können, ohne dass eine zusätzliche Genehmigung erforderlich ist. Sie können jetzt auch einzelne Benutzer, Gruppen oder IAM-Rollen daran hindern, automatisch genehmigbare Anforderungen zu übermitteln, selbst wenn sie von einer Änderungsvorlage unterstützt werden. Dies wird durch die Verwendung eines neuen IAM-Bedingungsschlüssels, `ssm:AutoApprove`, erreicht. Weitere Informationen finden Sie unter Controlling access to auto-approval runbook workflows (Steuern des Zugriffs auf Runbook-Workflows für automatische Genehmigung)	15. Juni 2022

Aktualisierte Anleitung für Wartungsfenster-Aufgaben	Zuvor bot Ihnen die Systems-Manager-Konsole die Möglichkeit, die von AWS verwaltete serviceverknüpfte IAM-Rolle `AWSServiceRoleForAmazonSSM` als Wartungsrolle für Ihre Aufgaben zu verwenden. Die Verwendung dieser Rolle und der zugehörigen Richtlinie, `AmazonSSMServiceRolePolicy` , für Wartungsfenster-Aufgaben wird nicht mehr empfohlen. Erstellen Sie stattdessen eine benutzerdefinierte Richtlinie und Rolle für Wartungsfenster-Aufgaben. Weitere Informationen erhalten Sie unter Einrichten von Maintenance Windows.	9. Juni 2022
Unterstützung für Portweiterleitung an Remote-Hosts für Session Manager	Session Manager unterstützt jetzt Port-Weiterleitungs-Sitzungen für Remote-Hosts. Der Remote-Host muss nicht von Systems Manager verwaltet werden. Weitere Informationen finden Sie unter Starting a session (port forwarding to remote host)Starten einer Sitzung (Port-Weiterleitung zum Remote-Host).	25. Mai 2022

Aktualisierter Inhalt: Anweisungen zum manuellen Installieren von SSM Agent auf Amazon-EC2-Linux-Instances	Als Reaktion auf Kundenfeedback haben wir die Themen mit Anweisungen zur manuellen Installation von SSM Agent auf Amazon-EC2-Instances überarbeitet. Diese Themen stellen jetzt Befehle bereit, die global verfügbare Dateien verwenden, die Sie für eine schnelle Installation auf EC2-Instances in jeder AWS-Region kopieren und einfügen können. Diese Themen enthalten auch Informationen, die Ihnen beim Erstellen von Installationsbefehlen helfen, die Dateien verwenden, die in Ihrer eigenen Arbeitsregion verfügbar sind. Der letztere Ansatz wird empfohlen, wenn Sie den Agenten mit einem Skript oder einer Vorlage auf mehreren Instances installieren. Weitere Informationen finden Sie in den Anweisungen für Ihr Linux-Betriebssystem im Abschnitt Manuelles Installieren von SSM Agent auf EC2-Instances für Linux.	9. Mai 2022

Neues Thema: Amazon Machine Images (AMIs) mit SSM Agent vorinstalliert	Als Reaktion auf Kundenfeedback haben wir zentralisierte Informationen darüber, welche von AWS verwaltete AMIs SSM Agent vorinstalliert haben. Dieses Thema enthält auch Anweisungen zum Überprüfen, ob eine aus diesen AMIs erstellte Amazon-EC2-Instance erfolgreich installiert wurde und ausgeführt wird. Für seltene Fälle, in denen der Agent möglicherweise nicht erfolgreich installiert wird bzw. installiert, aber nicht gestartet wird, stellen wir auch Informationen zum Starten oder manuellen Installieren des Agenten auf diesen Instances bereit. Details dazu finden Sie unter .Amazon Machine Images (AMIs) mit SSM Agent vorinstalliert.	8. Mai 2022
Neuer State Manager-Abschnitt	Es wurde ein neuer Abschnitt hinzugefügt, der die Details beschreibt, wann State Manager Zuordnugnen ausführt. Weitere Informationen finden Sie unter Über Zuordnungsplanung.	27. April 2022

Patch Manager unterstützt jetzt Rocky Linux	Sie können jetzt Patch Manager verwenden, um Rocky Linux-Knoten zu patchen. Viele der Regeln, die für RHEL 8 zum Patchen gelten, gelten auch für Rocky Linux. Rocky Linux 8 nutzt die neue `AWS-DefaultRockyLinuxPatchBaseline` . Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf RHEL, CentOS Stream und Rocky Linux.	14. April 2022

Patch Manager unterstützt jetzt CentOS Stream 8	Sie können jetzt Patch Manager verwenden, um CentOS Stream 8-Instances und Red Hat Enterprise Linux (RHEL) 4.4–4.5-Instances zu patchen. Viele der Regeln, die für RHEL 8 zum Patchen gelten, gelten auch für CentOS Stream 8. CentOS Stream 8 verwendet die `AWS-DefaultCentOSPatchBaseline` . Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf RHEL und CentOS Stream	4. April 2022

Erstellen einer Übernahmerolle für Change Manager	In einem neuen Abschnitt werden die Anforderungen für das Erstellen und Implementieren einer Übernahmerolle für Change Manager erläutert. Eine Übernahmerolle ist eine AWS Identity and Access Management (IAM)-Servicerolle, die es Change Manager ermöglicht, die in einer genehmigten Änderungsanforderung angegebenen Runbook-Workflows in Ihrem Namen sicher auszuführen. Die Rolle gewährt AWS Systems Manager (AWS STS) `AssumeRole` Vertrauen anChange Manager. Weitere Informationen finden Sie unter Konfigurieren von Rollen und Berechtigungen für Change Manager.	18. März 2022
Genehmigen oder Ablehnen von Change Manager-Änderungsanträgen auf einmal	In der Systems-Manager-Konsole können Sie jetzt mehrere Änderungsanträge auswählen, die in einem einzigen Vorgang genehmigt oder abgelehnt werden sollen. Weitere Informationen finden Sie unter Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen (Konsole).	8. März 2022

Support für von 2022 verwaltete Knoten Rocky Linux und Windows Server	Systems Manager unterstützt von 2022 verwaltete Knoten Rocky Linux und Windows Server, einschließlich Edge-Geräte und Hybrid-Maschinen, die sich On-Premises oder bei anderen Cloud-Anbietern befinden. Um Systems Manager mit diesen Betriebssystemen verwenden zu können, müssen Sie alle erforderlichen System-Manager-Einrichtungsverfahren ausführen, einschließlich Verfahren für Hybrid-Umgebungen oder Edge-Geräte, falls zutreffend. Weitere Informationen erhalten Sie unter Einrichten von Systems Manager. Für Rocky Linux-Maschinen müssen Sie auch SSM Agent manuell installieren. Weitere Informationen finden Sie unter Manuelles Installieren eines SSM Agent auf Rocky Linux-Instances. Für Amazon Elastic Compute Cloud (Amazon EC2)-Instances von Windows Server 2022 ist SSM Agent standardmäßig installiert.	1. März 2022

Erlauben Sie Automation, sich an Ihre Parallelitätsanforderungen anzupassen, und lassen Sie sich Kennzahlen zur Automation-Nutzung anzeigen	Sie können Automation jetzt erlauben, Ihr Kontingent für die gleichzeitige Automatisierung automatisch anzupassen und die Automatisierungsnutzungskennzahlen einzusehen, die veröffentlicht wurden. CloudWatch Weitere Informationen zur adaptiven Nebenläufigkeit finden Sie unter Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst. Weitere Informationen zum Anzeigen von Automation-Nutzungsmetriken finden Sie unter Automation-Metriken mit Amazon überwachen CloudWatch.	27. Januar 2022

Erlauben Sie Automation, sich an Ihre Parallelitätsanforderungen anzupassen, und sehen Sie sich die Nutzungsmetriken für Automation an	Sie können Automation jetzt erlauben, Ihr Kontingent für die gleichzeitige Automatisierung automatisch anzupassen und die Automatisierungsnutzungskennzahlen einzusehen, die veröffentlicht wurden. CloudWatch Weitere Informationen zur adaptiven Nebenläufigkeit finden Sie unter Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst. Weitere Informationen zum Anzeigen von Automation-Nutzungsmetriken finden Sie unter Automation-Metriken mit Amazon überwachen CloudWatch.	27. Januar 2022
Nach Kategorien geordnete Systems-Manager-Dokumente	Amazon-eigene Systems-Manager-Dokumente sind jetzt nach Typ und Kategorien geordnet, damit Sie die benötigten Dokumente leichter finden können.	13. Januar 2022

Integrationen für Automation erstellen und aufrufen	Ab sofort können Sie während einer Automatisierung Nachrichten über Webhooks senden, indem Sie eine Integration erstellen. Integrationen können während einer Automatisierung mit der neuen Aktion `aws:invokeWebhook` in Ihrem Runbook aufgerufen werden. Weitere Informationen zum Erstellen von Integrationen finden Sie unter Erstellen von Webhook-Integrationen für Automation. Weitere Informationen zur Aktion `aws:invokeWebhook` finden Sie unter `aws:invokeWebhook` – Automation-Webhook-Integration aufrufen.	13. Januar 2022
Funktionen, die in der neuen Version nicht verfügbar sind AWS-Region	Die folgenden Funktionen des Systems Manager sind derzeit in der neuen Region Asien-Pazifik (Jakarta) nicht verfügbar. • Application Manager • Change Calendar • Change Manager • Explorer • Fleet Manager • Incident Manager • Quick Setup	13. Dezember 2021

Anzeigen von Ressourcen-Preisdetails für eine Anwendung	Application Managerist AWS Billing and Cost Management über das Cost Explorer Explorer-Widget in integriert. Nachdem Sie den Cost Explorer in der Fakturierungs- und Kostenmanagement-Konsole aktiviert haben, zeigt das Cost-Explorer-Widget imApplication Manager Preisdaten für eine bestimmte Anwendung oder Anwendungskomponente ohne Container an. Sie können Filter im Widget verwenden, um Preisdaten nach verschiedenen Zeiträumen, Details und Preisarten in einem Balken- oder Liniendiagramm anzuzeigen. Weitere Informationen finden Sie unter Anzeigen von Übersichtsinformationen über eine Anwendung.	7. Dezember 2021
Prozesse verwalten mit Fleet Manager	Sie können jetzt Fleet Manager verwenden, um Prozesse auf Ihren Knoten zu verwalten. Weitere Informationen finden Sie unter Arbeiten mit Prozessen.	6. Dezember 2021

Terminologieänderung: verwaltete Instances sind jetzt verwaltete Knoten	Aufgrund der Unterstützung für AWS IoT Greengras s Kerngeräte wurde der Begriff verwaltete Instanz in den meisten Dokumente n von Systems Manager in Manager-Knoten geändert. Die Systems-Manager-Ko nsole, API-Aufrufe, Fehlermel dungen und SSM-Dokum ente verwenden weiterhin den Begriff Instance.	29. November 2021

Support für Edge-Geräte	Systems Manager unterstützt die folgenden Edge-Gerätekonfigurationen. • AWS IoT Greengrass: Systems Manager unterstützt jetzt jedes Gerät, das für die AWS IoT Greengrass Core-Software konfiguriert ist AWS IoT Greengrass und auf denen sie ausgeführt wird. Um Ihre AWS IoT Greengrass Kerngeräte zu integrieren, müssen Sie eine AWS Identity and Access Management (IAM-) Servicerolle erstellen. Sie müssen die AWS IoT Greengrass Konsole auch für die Bereitstellung SSM Agent als AWS IoT Greengrass Komponente auf Ihren Geräten verwenden. Weitere Informationen finden Sie unter Einrichtung AWS Systems Manager für Edge-Geräte. • Edge-Geräte in einer Hybridumgebung: Systems Manager unterstützt auch AWS IoT Core-Geräte und AWS Nicht-IoT-Geräte, nachdem Sie sie als lokale Maschinen konfiguriert haben. Für das	29. November 2021

	Onboarding Ihrer Geräte müssen Sie eine IAM-Servicerolle erstellen, eine Aktivierung für verwaltete Knoten für eine Hybrid-Umgebung erstellen und SSM Agent manuell auf Ihren Geräten installieren. Weitere Informationen finden Sie unter Einrichtung AWS Systems Manager für Hybridumgebungen	
Verbindung mit verwalteten Instances über Remote Desktop	Sie können jetzt Fleet Manager verwenden, um eine Verbindung zu verwalteten Windows-Instances mithilfe des Remote Desktop Protocol (RDP) herzustellen. Diese Remote-Desktop-Sitzungen, die von NICE DCV unterstützt werden, bieten sichere Verbindungen zu Ihren Instances direkt von Ihrem Browser aus. Weitere Informationen finden Sie unter Verbinden über Remote Desktop.	23. November 2021

Angeben einer maximalen Sitzungsdauer und Lieferun von Gründen für Sitzungen	Sie können jetzt eine maximale Sitzungsdauer für alle Session Manager-S itzungen in einer AWS-Region in Ihrem AWS-Konto festlegen . Wenn eine Sitzung die von Ihnen angegebene Dauer erreicht, wird sie beendet. Sie können jetzt auch optional einen Grund hinzufüge n, wenn Sie eine Sitzung starten. Weitere Informationen finden Sie unter Angeben der maximalen Sitzungsdauer.	16. November 2021

Patch Manager unterstützt jetzt das Raspberry Pi OS-Betriebssystem	Sie können jetzt Patch Manager verwenden, um Raspberry Pi OS-Instances zu patchen. Patch Manager unterstützt das Patchen von Raspberry Pi OS 9 (Stretch) und 10 (Buster). Da es sich bei dem Raspberry Pi OS um ein Debian-basiertes Betriebssystem handelt, gelten dafür viele der gleichen Patch-Regeln wie für Debian Server. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Debian Server und Raspberry Pi OS	16. November 2021
Zugreifen auf das Red-Hat-Knowledgebase-Portal	Verwenden Sie Fleet Manager, um auf das RHEL-Knowledgebase-Portal zuzugreifen, um Lösungen, Artikel, Dokumentationen und Videos zur Verwendung von Red-Hat-Produkten zu finden. Weitere Informationen finden Sie unter Zugriff auf das Red-Hat-Knowledgebase-Portal.	3. November 2021

Massenbearbeitung von OpsItems	OpsCenter unterstützt jetzt Massenbearbeitung von OpsItems. Sie können mehrere OpsItems auswählen und eines der folgenden Felder bearbeiten: Status (Status), Priority (Priorität), Severity (Schweregrad), Category (Kategorie). Weitere Informationen finden Sie unter Bearbeiten von OpsItems.	15. Oktober 2021
Erstellen Sie Eingabeparameter, mit denen Ressourcen aufgefüllt werden AWS	Sie können jetzt Eingabeparameter in Automation-Runbooks erstellen, die AWS -Ressourcen in AWS Management Console ausfüllen. Weitere Informationen finden Sie unter Eingabeparameter erstellen, mit denen Ressourcen aufgefüllt werden. AWS	14. Oktober 2021
Neue Abschalt-Option für Aufgabenaufrufe für Wartungsfenster	Sie können jetzt verhindern, dass neue Aufgabenaufrufe starten, nachdem die für ein Wartungsfenster angegebene Abschaltzeit erreicht wurde. Weitere Informationen finden Sie unter Zuweisen von Aufgaben zu einem Wartungsfenster (Konsole).	13. Oktober 2021

Patch Manager-Support für macOS 11.3.1 und 11.4 (Big Sur)	Amazon Elastic Compute Cloud (Amazon EC2)-Instances für macOS 11.3.1 und 11.4 (Big Sur) können jetzt mit Patch Manager gepatcht werden. Dies ist zusätzlich zur bestehenden Support für macOS 10.14.x (Mojave) und 10.15.x (Catalina). Informationen zum Arbeiten mit Patch Manager finden Sie unter AWS Systems ManagerPatch Manager.	1. Oktober 2021

Anwendungseinblicke in Application Manager	Application Managerintegriert sich in Amazon CloudWatch Application Insights. Application Insights identifiziert Schlüsselmetriken, Protokolle und Alarme und richtet diese für Ihre Anwendungsressourcen und Ihren Technologie-Stack ein. Application Insights überwacht fortlaufend Metriken und Protokolle, um Anomalien und Fehler zu erkennen und zu korrelieren. Wenn das System Fehler oder Anomalien erkennt, generiert Application Insights CloudWatch Ereignisse, anhand derer Sie Benachrichtigungen einrichten oder Maßnahmen ergreifen können. Sie können Application Insights auf den Tabs Übersicht und Überwachungin Application Manager aktivieren und ansehen. Weitere Informationen zu Application Insights finden Sie unter Was ist Amazon CloudWatch Application Insights im CloudWatch Amazon-Benutzerhandbuch.	21. September 2021

Importieren von Ereignissen aus anderen Kalendern in Change Calendar	Sie können nun die Ereignisse aus einem Drittanbieter-Kalender in einen Kalender in Change Calendar importieren. Zuvor musste jedes Ereignis manuell in einen Kalender eingegeben werden. Nachdem Sie einen Kalender von einem unterstützten Drittanbieter-Kalenderanbieter in eine iCalendar (`.ics`)-Datei exportiert haben, importieren Sie ihn in Change Calendar. Die Ereignisse sind dann in den Regeln für den offenen oder geschlossenen Kalender in Systems Manager enthalten. Zu den unterstützten Anbietern zählen iCloud-Kalender, Google-Kalender und Microsoft Outlook. Weitere Informationen finden Sie unter Importieren und Verwalten von Ereignissen aus Drittanbieter-Kalendern.	8. September 2021

Neue Tagging- und Runbook-Features in Application Manager	Tagging-Erweiterungen umfassen die Möglichkeit, Tags zu einer bestimmte n Ressource oder allen Ressourcen in einer Applicati on Manager-Anwendung hinzuzufügen oder sie zu entfernen. Runbook-E rweiterungen umfassen die Möglichkeit, eine gefilterte Liste von Runbooks für einen bestimmten Ressourcentyp anzuzeigen oder ein Runbook für alle Ressourcen desselben Typs zu initiieren. Weitere Informationen finden Sie unter Arbeiten mit Tags in Applicati on Manager und Arbeiten mit Runbooks in Application Manager.	31. August 2021
Neues Beispiel: Erstellen Sie eine Änderungsanforderung mit dem AWS CLI	Ein Beispiel für die Erstellun g einer Änderungsanforderu ng mit dem AWS CLI wurde dem Change Manager Kapitel hinzugefügt. Im Beispiel wird die `AWS-HelloWorldChan geTemplate` -Änderung svorlage und Folgendes `AWS-HelloWorld runbook` verwendet: • Erstellen von Änderungs anforderungen (AWS CLI)	20. August 2021

Neuer Abschnitt: Verwenden von Parametern in Amazon EKS	Einer neuer Abschnitt wurde dem Kapitel Parameter Store hinzugefügt. Dieses Thema enthält eine exemplarische Vorgehensweise zur Verwendung Ihrer Parameter in Amazon EKS-Clustern. Weitere Informationen finden Sie unter Verwenden von Parameter Store-Parametern im Amazon Elastic Kubernetes Service.	19. August 2021
Aktualisierte Patch Manager Lebenszyklus-Hooks	Patch Manager bietet jetzt einen Lebenszyklus-Hook – die Möglichkeit, ein Systems Manager-Befehlsdokument auszuführen – für einen zusätzlichen Punkt während der Patch-Operation Patch now (Jetzt patchen). Wenn Sie einen Neustart der Instance planen, nachdem Sie Jetzt patchen ausgeführt haben, können Sie einen Lebenszyklus-Hook angeben, der nach Abschluss des Neustarts ausgeführt werden soll. Weitere Informationen finden Sie unter Lebenszyklus-Hooks „Patch now“ (Jetzt patchen) verwenden und unter Informationen über das `AWS-RunPatchBaselineWithHooks` SSM-Dokument.	9. August 2021

Automatische Genehmigungen jetzt für Change Manager-Anforderungen unterstützt	Sie können jetzt Änderungsvorlagen in Change Manager so konfigurieren, dass automatische Genehmigungen unterstützt werden. Dies bedeutet, dass Benutzer mit den erforderlichen IAM-Berechtigungen die Änderungsanforderung starten können, ohne dass eine zusätzliche Genehmigung erforderlich ist. Benutzer, die Zugriff auf Vorlagen für automatische Genehmigungen haben, können weiterhin Genehmiger angeben, wenn sie dies wünschen. Um Ihnen bei der Kontrolle Ihrer Change Manager-Prozesse zu helfen, sind Genehmigungen weiterhin für alle Anforderungen während der Change–Freeze-Perioden erforderlich. Weitere Informationen finden Sie unter den folgenden Themen: • Erstellen von Änderungsvorlagen • Erstellen von Änderungsanforderungen • Probieren Sie die Vorlage für AWS verwaltete `Hello World` Änderungen aus	30. Juli 2021

Betriebliche OpsCenter-Einblicke	OpsCenter analysiert automatisch OpsItems in Ihrem Konto und generiert Insights. Ein Insight enthält Informationen, die Ihnen vermitteln, wie viele Duplikat-OpsItems sich in Ihrem Konto befinden und welche Quellen sie erstellen. Insights bieten auch empfohlene bewährte Methoden und Automatisierungs-Runbooks, um Duplikat-OpsItems zu beheben. Weitere Informationen finden Sie unter Arbeiten mit operativen Einblicken.	13. Juli 2021
Angehaltene Instances in Fleet Manager aufrufen	Sie können jetzt anzeigen, welche Instances in der Fleet Manager-Konsole `running` und welche Instances `stopped` sind. Weitere Informationen finden Sie unter AWS Systems ManagerFleet Manager.	12. Juli 2021
Neues Thema: Verfassen von Automation-Runbooks	Das neue Thema Verfassen von Automatisierungs-Runbooksenthält Anleitungen und erläuternde Beispiele für die Erstellung von Inhalten für benutzerdefinierte Automatisierungs-Runbooks.	8. Juli 2021

AWS CloudFormation Stapel- und Vorlagenerstellung in Application Manager	Application Managerhilft Ihnen bei der Bereitstellung und Verwaltung von Ressourcen für Ihre Anwendungen durch die Integration mit CloudFormation. Sie können AWS CloudFormation Vorlagen und Stacks in Application Manager erstellen, bearbeiten und löschen. Application Managerenthält auch eine Vorlagenbibliothek, in der Sie Vorlagen klonen, erstellen und speichern können. Application Managerund CloudFormation zeigt dieselben Informationen über den aktuellen Status eines Stacks an. Vorlagen und Vorlagenaktualisierungen werden in Systems Manager gespeichert, bis Sie den Stack bereitstellen. Zu diesem Zeitpunkt werden die Änderungen auch angezeigt CloudFormation. Weitere Informationen finden Sie unter Arbeiten mit AWS CloudFormation Stacks in Application Manager.	8. Juli 2021

Neues Thema: Automatisches Drehen privater Schlüssel für SSM Agent auf Hybrid-Instances	Das neue Thema Automatisches Drehen privater Schlüssel einrichten enthält Anweisungen zum Verbessern Ihres Sicherheitsstatus, indem Sie den SSM Agent so konfigurieren, dass private Schlüssel der hybriden Umgebung automatisch gedreht werden.	15. Juni 2021
Session ManagerPlugin für die AWS CLI Version 1.2.205.0	Eine neue Version des Session Manager Plugins für AWS CLI wurde veröffentlicht. Weitere Informationen finden Sie unter Aktuelle Version und Versionsverlauf des Session Manager-Plug-Ins.	10. Juni 2021
Neue serviceverknüpfte IAM-Rolle	Wenn Sie OpsCenter betriebliche Erkenntnisse aktivieren, erstellt Systems Manager eine neue serviceverknüpfte AWS Identity and Access Management (IAM)-Rolle namens `AWSSSMOps InsightsServiceRolePolicy` . Weitere Informationen zu dieser Rolle finden Sie unter Verwenden von Rollen, um betriebliche Einblicke OpsItems in Systems Manager zu gewinnenOpsCenter: `AWSSSMOps InsightsServiceRolePolicy` .	9. Juni 2021

Neuer Patch Manager-Fehlerbehebungsinhalt von Linux	Ein neues Thema Fehler beim Ausführen von `AWS-RunPatchBaseline` unter Linux enthält Beschreibungen und Lösungen für verschiedene Probleme, die beim Patchen verwalteter Instances mit Linux-Betriebssystemen auftreten können.	8. Juni 2021

Verbesserte Unterstützung für Wartungsfenster-Aufgaben, für die keine festgelegten Ziele erforderlich sind (Konsole)	Sie können jetzt Wartungsfenster-Aufgaben in der Konsole erstellen, ohne ein Ziel in der Aufgabe angeben zu müssen, falls dies nicht erforderlich ist. Bisher war diese Option nur verfügbar, wenn die API AWS CLI oder verwendet wurde. Diese Option gilt für die AWS Step Functions Aufgabentypen Automatisierung und. AWS Lambda Wenn Sie beispielsweise eine Automatisierungsaufgabe erstellen und die zu aktualisierenden Ressourcen in den Dokument-Parametern für Automatisierung angegeben sind, müssen Sie kein Ziel mehr in der Aufgabe selbst angeben. Weitere Informationen finden Sie unter Registrieren von Wartungsfensteraufgaben ohne Ziele, Zuweisen von Aufgaben zu einem Wartungsfenster (Konsole) und Planen von Automatisierungen mit Wartungsfenstern.	28. Mai 2021

Referenz zu Automation-Runbooks verschoben	Die Referenz zum Automatisierungs-Runbook wurde an einen neuen Speicherort verschoben. Weitere Informationen finden Sie unter Referenz zu Systems Manager Automation.	10. Mai 2021
AWS Systems Manager Incident Manager starten	Incident Manager ist eine Incident-Management-Konsole, die Benutzern hilft, Vorfälle, die sich auf ihre AWS gehosteten Anwendungen auswirken, zu minimieren und diese zu beheben. Weitere Informationen finden Sie im AWS Systems Manager Incident Manager -Benutzerhandbuch.	10. Mai 2021
State Manager unterstützt Change Calendar	Sie können jetzt Change Calendar-Namen oder Amazon-Ressourcennamen (ARNs) beim Erstellen oder Aktualisieren einer State Manager-Vereinigung angeben. State Manager wendet Zuordnungen nur an, wenn der Änderungskalender geöffnet ist, nicht wenn er geschlossen wird. Weitere Informationen finden Sie unter Erstellen von Zuordnungen und Bearbeiten und Erstellen einer neuen Version einer Zuordnung.	6. Mai 2021

Klonen von Systems Manager-Dokumenten	Mit der Systems Manager-Dokumentenkonsole können Sie nun Inhalte aus einem vorhandenen Dokument in ein neues Dokument kopieren, das Sie ändern können. Weitere Informationen hierzu finden Sie unter Klonen eines SSM-Dokuments.	4. Mai 2021
Integrieren von Security Hub mit Explorer und OpsCenter	Sie können jetzt integrieren Explorer und OpsCenter mit AWS Security Hub. Security Hub bietet einen umfassenden Überblick über Ihren Sicherheitsstatus AWS und hilft Ihnen dabei, Ihre Umgebung anhand von Industriestandards und Best Practices zu überprüfen. Bei der Integration in Explorer können Sie die Sicherheitsergebnisse im Security Hub Widget auf dem Explorer-Dashboard aufrufen. Bei der Integration in OpsCenter können Sie OpsItems zu Security Hub-Ergebnissen erstellen. Weitere Informationen finden Sie unter Empfangen von Ergebnissen von AWS Security Hub in Explorer und Empfangen von Ergebnissen von AWS Security Hub in OpsCenter.	27. April 2021

Neues Thema: Dokumentkonventionen	Wir haben ein neues Thema hinzugefügt, um den Benutzern die allgemeinen typografischen Konventionen für das AWS Systems Manager -Benutzerhandbuch zu vermitteln. Weitere Informationen finden Sie unter Document Conventions.	21. April 2021
Aktualisiertes Thema: Informationen zum Patchen von Anwendungen, die von Microsoft unter veröffentlicht wurdenWindows Server	Aus dem Thema About patching applications released by Microsoft on Windows Server (Informationen zum Patchen von Anwendungen, die von Microsoft unter Patch Manager veröffentlicht werden) geht hervor, dass die Windows-Update-Option Give me updates for other Microsoft products when I update Windows (Updates für andere Microsoft-Produkte bereitstellen, wenn ich ein Windows-Update ausführe) auf der Instance aktiviert sein muss, damit Microsoft-Anwendungen auf Ihren von Windows Server verwalteten Instances patchen kann.	12. April 2021

Neuorganisation der Automatisierungs-Runbook-Referenz	Um Ihnen dabei zu helfen, die benötigten Runbooks zu finden und effizienter durch die Referenz zu navigieren, haben wir den Inhalt in der Automation-Runbook-Referenz nach dem jeweiligen AWS-Service umorganisiert. diese Änderungen finden Sie unter Referenz für Systems Manager Automation.	12. April 2021

Patch Manager: Generieren von CSV-Patch-Compliance-Berichten	Patch Manager unterstützt jetzt die Möglichkeit, Patch-Compliance-Berichte für Ihre Instances zu erstellen und den Bericht in einem S3-Bucket Ihrer Wahl im CSV-Format zu speichern. Anschließend können Sie mit einem Tool wie Amazon QuickSight die Daten des Patch-Compliance-Berichts analysieren. Sie können einen Patch-Compliance-Bericht für eine einzelne Instance oder für alle Instances in Ihrem AWS-Konto generieren. Sie können bei Bedarf einen einmaligen Bericht generieren oder einen Zeitplan für die automatische Erstellung von Berichten einrichten. Sie können auch ein Thema zum Amazon Simple Notification Service angeben, um Benachrichtigungen zu erhalten, wenn ein Bericht erstellt wird. Weitere Informationen finden Sie unter Erstellen von CSV-Patch-Compliance-Berichten.	9. April 2021

Löschen von Parameter Store-Parameterbezeichnungen	Sie können jetzt Parameter Store-Parameterbeschriftungen mithilfe der Systems Manager Konsole oder der AWS CLI löschen. Weitere Informationen finden Sie im Artikel zum Arbeiten mit Parameterbezeichnungen.	6. April 2021
Planen eines Neustarts der Instance bei Verwendung von Patch Now (Jetzt patchen)	Patch Manager unterstützt jetzt die Planung einer Zeit für den Neustart Ihrer Instances nach der Installation von Patches mit dem „Patch now“ (Jetzt patchen)-Feature. Dies gilt zusätzlich zu den vorhandenen Optionen, Instances nur neu zu starten, wenn dies erforderlich ist, um eine Patch-Installation abzuschließen oder den Neustart nach dem Patch-Vorgang zu überspringen. Informationen finden Sie unter Instances auf Abruf patchen .	01. April 2021
Neues Thema: Entdecken von öffentlichen Parametern	Parameter StoreÖffentliche Parameter können jetzt über die AWS CLI oder Systems Manager Manager-Konsole gefunden werden. Weitere Informationen finden Sie unter Auffinden von öffentlichen Parametern.	01. April 2021

„Patch now“ (Jetzt patchen)-Updates: Protokolle in S3 speichern und Lebenszyklus-Hooks ausführen	Wenn Sie die Patch Manager Patch now (Jetzt patchen)-Operation ausführen, können Sie einen S3-Bucket auswählen, in dem Patchprotokolle automatisch gespeichert werden sollen. Darüber hinaus können Sie Systems Manager Command-Dokumente (SSM-Dokumente) an drei Punkten während des Vorgangs als Lebenszyklus-Hooks ausführen: Vor der Installation, Nach der Installation und Beim Verlassen. Weitere Informationen finden Sie unter Instances auf Abruf patchen .	31. März 2021
Systems Manager meldet jetzt Änderungen an seinen AWS verwalteten Richtlinien	Ab dem 24. März 2021 werden Änderungen an verwalteten Richtlinien im Thema Systems ManagerAktualisierungen AWS verwalteter Richtlinien veröffentlicht. Bei der ersten aufgeführten Änderung handelt es sich um die Hinzufügung der Unterstützung für die Explorer Möglichkeit, Berichte OpsData OpsItems aus mehreren Konten und Regionen zu erstellen.	24. März 2021

Explorerermöglicht automatisch alle OpsData Quellen für die Synchronisierung von Ressourcendaten auf der Grundlage von Konten in AWS Organizations	Wenn Sie eine Ressourcendatensynchronisierung erstellen und eine der AWS Organizations Optionen wählen, lässt Systems Manager automatisch alle OpsData Quellen in der ausgewählten AWS-Regionen für alle AWS-Konten in Ihrer Organisation (oder in den ausgewählten Organisationseinheiten) zu. Dies bedeutet beispielsweise, dass Systems Manager automatisch Daten aus dieser Region sammelt AWS-Region OpsData , auch wenn Sie keine zugelassen Explorer haben, wenn Sie eine AWS Organizations Option für Ihre Ressourcendatensynchronisierung auswählen. Weitere Informationen finden Sie unter Synchronisierung mehrerer Konto- und Regions-Ressourcendaten.	24. März 2021
Systems Manager Automation bietet eine neue Systemvariable für Ihre Runbooks	Mit der neuen `global:AWS_PARTITION` Systemvariablen können Sie bei der Erstellung Ihrer AWS Runbooks die Partition angeben, in der sich eine Ressource befindet. Weitere Informationen finden Sie unter Automation-Systemvariablen.	18. März 2021

Zulassen mehrerer Genehmigungsebenen für Change Manager-Änderungsanforderungen	Wenn Sie eine Change Manager-Änderungsvorlage erstellen, können Sie jetzt verlangen, dass mehr als eine Genehmigerebene die Berechtigung für die Ausführung einer Änderungsanforderung erteilen. Sie können beispielsweise verlangen, dass technische Prüfer eine Änderungsanforderung, die aus einer Änderungsvorlage erstellt wurde, zuerst genehmigen und dann eine zweite Genehmigungsebene von einem oder mehreren Managern anfordern. Weitere Informationen finden Sie unter Erstellen von Änderungsvorlagen.	4. März 2021
Patch Manager unterstützt jetzt Oracle Linux 8.x	Sie können jetzt Patch Manager verwenden, um Oracle Linux 8.x-Instances bis Version 8.3 zu patchen. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux	1. März 2021

OpsCenter zeigt andere OpsItems für eine ausgewählte Ressource an	Um Ihnen bei der Untersuchung von Problemen zu helfen und den Kontext für ein Problem bereitzustellen, können Sie sich eine Liste von Ressourcen OpsItems für eine bestimmte AWS Ressource anzeigen lassen. In der Liste werden Status, Schweregrad und Titel der einzelnen OpsItem angezeigt. Die Liste enthält auch Deep-Links zu jedem OpsItem. Weitere Informationen finden Sie unter Andere OpsItems für eine bestimmte Ressource anzeigen.	1. März 2021
Definieren von Patching-Voreinstellungen zur Laufzeit	Sie können jetzt mit dem Baseline-Override-Feature Patching-Voreinstellungen zur Laufzeit definieren. Weitere Informationen finden Sie unter Verwenden des BaselineOverride Parameters.	25. Februar 2021

Neuer Systems Manager-Dokumenttyp	AWS CloudFormation Vorlagen können jetzt als Systems Manager Manager-Dokumente gespeichert werden. Durch das Speichern von CloudFormation Vorlagen als Systems Manager Manager-Dokumente können Sie von den Funktionen von Systems Manager Manager-Dokumenten wie Versionierung, Vergleich von Versionsinhalten und Teilen mit Konten profitieren. Weitere Informationen finden Sie unter AWS Systems Manager-Dokumente.	9. Februar 2021

Patch-Instances mit optionalen Hooks	Das neue SSM-Dokument `AWS-RunPatchBaselineWithHooks` bietet Hooks, mit denen Sie SSM-Dokumente an drei Punkten während des Instance-Patching-Zyklus ausführen können. Weitere Informationen zu `AWS-RunPatchBaselineWithHooks` finden Sie unter Informationen über das SSM-Dokument `AWS-RunPatchBaselineWithHooks`. Eine exemplarische Vorgehensweise für einen Patching-Vorgang, der alle drei Hooks verwendet, finden Sie unter Exemplarische Vorgehensweise: Aktualisieren von Anwendungsabhängigkeiten, Patchen einer Instance und Durchführen einer anwendungsspezifischen Zustandsprüfung.	2. Februar 2021

Neues Thema:Überprüfen von On-Premises-Servern und virtuellen Maschinen mit einem Hardware-Fingerabdruck	SSM Agent überprüft die Identifizierung von On-Premises-Servern und virtuellen Maschinen und VMs, die Sie mit einem berechneten Fingerabdruck registrieren. Der Fingerabdruck ist eine undurchsichtige, im Vault gespeicherte Zeichenfolge, die der Agent an bestimmte Systems Manager-APIs weitergibt. Informationen zum Hardware-Fingerabdruck und Anweisungen zum Konfigurieren eines Ähnlichkeitsschwellenwerts zur Unterstützung der Maschinenverifizierung finden Sie unter Überprüfen On-Premises-Server und virtueller Computer mithilfe eines Hardware-Fingerabdrucks.	25. Januar 2021
Neues Thema: Technische SSM Agent-Referenz	Das Thema SSM Agenttechnische Referenz enthält Informationen, die Ihnen helfen sollen, den Agenten zu implementieren AWS Systems Manager SSM Agent und zu verstehen, wie er funktioniert. Dieses Thema enthält einen brandneuen Abschnitt mit SSM Agentfortlaufenden Updates von AWS-Regionen.	21. Januar 2021

SSM Agent auf Windows Server 2008	Seit dem 14. Januar 2020 wird Windows Server 2008 für Feature- oder Sicherheitsupdates von Microsoft nicht mehr unterstützt. Windows Server 2008 AMIs enthalten zwar den SSM Agent, aber der Agent wird für dieses Betriebssystem nicht mehr aktualisiert.	5. Januar 2021
Verbesserte Unterstützung für Aufgaben im Wartungsfenster, für die keine bestimmten Ziele (AWS CLI und nur API) erforderlich sind	Sie können jetzt Aufgaben im Wartungsfenster erstellen, ohne ein Ziel in der Aufgabe angeben zu müssen, falls eines nicht erforderlich ist (AWS CLI und nur über die API). Dies gilt für Automatisierung AWS Lambda und AWS Step Functions Aufgabentypen. Wenn Sie beispielsweise eine Automatisierungsaufgabe erstellen und die zu aktualisierenden Ressourcen in den Runbook-Parametern für Automatisierung angegeben sind, müssen Sie kein Ziel mehr in der Aufgabe selbst angeben. Weitere Informationen finden Sie unter Registrieren von Wartungsfensteraufgaben ohne Ziele und Planen von Automatisierungen mit Wartungsfenstern.	23. Dezember 2020

Neue Automation-Features	Eine neue freigegebene Eigenschaft wurde zu Systems Manager Automation Runbooks hinzugefügt. Mit der `onCancel`-Eigenschaft können Sie angeben, zu welchem Schritt die Automatisierung gehen soll, falls ein Benutzer die Automatisierung abbricht. Weitere Informationen finden Sie unter Eigenschaften, die von allen Aktionen gemeinsam genutzt werden.	21. Dezember 2020
Neues Thema: Arbeiten mit Zuordnungen mithilfe von IAM	Es wurde ein neues Thema zum Systems Manager State Manager-Kapitel hinzugefügt, das die bewährten Methoden zum Erstellen von Zuordnungen mit IAM beschreibt. Weitere Informationen finden Sie unter Arbeiten mit Mappings mithilfe von IAM.	18. Dezember 2020
State Manager unterstützt jetzt Multi-Regionen und Multi-Konten	Verknüpfungen können jetzt mit mehreren Regionen oder Konten erstellt oder aktualisiert werden. Weitere Informationen finden Sie unter Erstellen von Zuordnungen.	15. Dezember 2020

Neue Funktion: Fleet Manager	Fleet Manager, eine Funktion von AWS Systems Manager, ist eine einheitliche Benutzeroberfläche (UI), mit der Sie Ihre Serverflotte, die vor Ort oder vor Ort läuft AWS, remote verwalten können. Mit Fleet Manager können Sie sich den Zustand und den Leistungsstatus Ihrer gesamten Serverflotte von einer Konsole aus ansehen. Sie können auch Daten aus einzelnen Instances sammeln, um allgemeine Problembehandlungs- und Verwaltungsaufgaben über die Konsole auszuführen. Weitere Informationen finden Sie unter AWS Systems Manager.Fleet Manager	15. Dezember 2020

Neue Funktion: Change Manager	Amazon Web Services hat Change Manager, ein unternehmensweites Change-Management-Framework zum Anfordern, Genehmigen, Implementieren und Melden von Betriebsänderungen an Ihrer Anwendungskonfiguration und Infrastruktur, veröffentlicht. Wenn Sie ein einziges delegiertes Administratorkonto verwenden AWS Organizations, können Sie Änderungen an mehreren oder mehreren AWS-Konten verwalten. AWS-Regionen Alternativ können Sie mit einem lokalen Konto Änderungen für einen einzigen AWS-Konto verwalten. Wird Change Manager für die Verwaltung von Änderungen sowohl an AWS Ressourcen als auch an lokalen Ressourcen verwendet. Weitere Informationen finden Sie unter AWS Systems Manager.Change Manager	15. Dezember 2020

Neue Funktion: Application Manager	Application Managerhilft Ihnen dabei, Probleme mit Ihren AWS Ressourcen im Kontext Ihrer Anwendung en zu untersuchen und zu beheben. Application Managerfasst Betriebsi nformationen aus mehreren Funktionen AWS-Services und Systems Manager Manager-Funktionen in einer einzigen AWS Management Console zusammen. Weitere Informati onen finden Sie unter AWS Systems Manager.Application Manager	15. Dezember 2020

AWS Systems Manager unterstützt Amazon EC2 EC2-Instances für macOS	Zusammen mit der Veröffentlichung der Amazon Elastic Compute Cloud (Amazon EC2)-Unterstützung für macOS-Instances unterstützt Systems Manager jetzt viele Operationen auf EC2-Instances für macOS. Zu den unterstützten Versionen gehören macOS 10.14.x (Mojave) und 10.15.x (Catalina). Weitere Informationen finden Sie unter den folgenden Themen. • Weitere Informationen zur Installation von SSM Agent auf EC2-Instances für macOS finden Sie unter Installieren und Konfigurieren von SSM Agent auf EC2-Instances für macOS. • Weitere Informationen zum Patchen von EC2-Instances für macOS finden Sie unter Wie Patches installiert werden und So erstellen Sie eine benutzerdefinierte Patch-Baseline (macOS). • Allgemeine Informationen zur Unterstützung von EC2-Instances für macOS finden Sie unter Amazon EC2 Mac-Instances im Amazon EC2 EC2-Benutzerhandbuch.	30. November 2020

Pseudo-Parameter des Wartungsfensters: Neuer Ressourcentyp wird für `{{TARGET_ID}}` und `{{RESOURCE_ID}}` unterstützt	Ein zusätzlicher Ressourcentyp steht nun für die Verwendung mit den Pseudo-Parametern `{{TARGET_ID}}` und `{{RESOURCE_ID}}` zur Verfügung. Sie können jetzt den Ressourcentyp `AWS::RDS::DBCluster` mit diesen beiden Pseudo-Parametern verwenden. Informationen zu Pseudo-Parametern für Wartungsfenster finden Sie unter Pseudo-Parameter bei der Registrierung von Wartungsfensteraufgaben verwenden.	27. November 2020
Session ManagerPlugin für die Version 1.2.30.0 AWS CLI	Eine neue Version des Session Manager Plugins für AWS CLI wurde veröffentlicht. Weitere Informationen finden Sie unter Aktuelle Version und Versionsverlauf des Session Manager-Plug-Ins.	24. November 2020
Neues Thema: SSM-Dokumentversionen vergleichen	Sie können nun die Unterschiede im Inhalt zwischen Versionen von SSM-Dokumenten in der Systems Manager-Dokumentenkonsole vergleichen. Weitere Informationen finden Sie unter Vergleichen von SSM-Dokumentversionen.	24. November 2020

Systems Manager unterstützt jetzt VPC-Endpunktrichtlinien	Sie können jetzt Richtlinien für VPC-Schnittstellenendpunkte für Systems Manager erstellen . Weitere Informationen finden Sie unter Erstellen einer VPC-Endpunktrichtlinie.	18. November 2020
Neues Thema: Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen	Sie können nun festlegen, wie lange ein Benutzer inaktiv sein soll, bevor eine Sitzung mit Session Manager endet. Weitere Informationen finden Sie unter Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen.	18. November 2020
Neues Session Manager-Protokollierungsfeature	Sie können jetzt einen kontinuierlichen Stream von Sitzungsdatenprotokollen im JSON-Format an Amazon Logs senden. CloudWatch Weitere Informationen finden Sie unter Streaming-Sitzungsdaten mithilfe von Amazon CloudWatch Logs.	18. November 2020
Neues Thema: Überprüfen der Signatur des SSM Agent	Sie können nun die kryptografische Signatur des Installationspakets für den SSM Agent auf Linux-Instances verifizieren. Weitere Informationen finden Sie unter SSM-Dokumentschemata und -funktionen.	17. November 2020

Neues Thema: Grundlegendes zu Automatisierungsstatus	Dem Kapitel „Systems Manager Automation“ wurde ein neues Thema hinzugefügt, in dem die Status für Aktionen und Automatisierungen beschrieben werden. Weitere Informationen finden Sie unter Grundlegendes zu Automation-Status.	17. November 2020
Neue Quelltypen für den `aws:downloadContent` -Plugin	Git und HTTP werden jetzt als Quelltypen für den `aws:downloadContent` -Plugin unterstützt. Weitere Informationen finden Sie unter `aws:downloadContent` .	17. November 2020
Neues Schemafeature für Systems Manager-Dokument (SSM-Dokument)	In SSM-Dokumenten mit Schema-Version 2.2 oder neuer unterstützt der `precondition` -Parameter jetzt die Referenzierung der Eingabeparameter Ihres Dokuments. Weitere Informationen finden Sie unter SSM-Dokumentschemata und -funktionen.	17. November 2020

Neue Datenquelle inExplorer: AWS Config	Explorerzeigt jetzt Informationen zur AWS Config Konformität an, einschließlich einer allgemeinen Zusammenfassung der konformen und nicht konformen AWS Config Regeln, der Anzahl der konformen und nicht konformen Ressourcen sowie spezifische Details zu den einzelnen Regeln (wenn Sie eine nicht konforme Regel oder Ressource genauer untersuchen). Weitere Informationen finden Sie unter Bearbeiten von Systems Manager-Datenquellen.	11. November 2020
Neues Thema: Ausführen von Auto-Scaling-Gruppen mit Zuordnungen	Es wurde ein neuer Abschnitt zu State Manager hinzugefügt, der die bewährten Methoden zum Erstellen von Zuordnungen zum Ausführen von Auto-Scaling-Gruppen beschreibt. Weitere Informationen finden Sie unter Ausführen von Auto-Scaling-Gruppen mit Zuordnungen.	10. November 2020
Quick Setup unterstützt jetzt das Targeting einer Ressourcengruppe	Quick Setup unterstützt nun die Auswahl einer Ressourcengruppe als Ziel für den lokalen Einrichtungstyp. Weitere Informationen finden Sie unter Auswählen von Zielen für Quick Setup.	5. November 2020

Patch Manager bietet Unterstützung für Debian Server 10 LTS, Oracle Linux 7.9 LTS und Ubuntu Server 20.10 STR	Sie können jetzt Patch Manager verwenden, um Debian Server 10 LTS-, Oracle Linux 7.9 LTS- und Ubuntu Server 20.10 STR-Instances zu patchen. Weitere Informationen finden Sie unter den folgenden Themen: • Patch Manager-Voraussetzungen • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Debian Server • Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux • Funktionsweise von Patch-Baseline-Regeln auf Ubuntu Server	4. November 2020

Neue Unterstützung für EventBridge AWS Systems ManagerChange Calendar	Amazon bietet EventBridge jetzt Unterstützung für Change Calendar Ereignisse und Ereignisse in den Veranstal tungsregeln. Wenn sich der Status eines Kalenders ändert, EventBridge können Sie die Zielaktion einleiten, die Sie als EventBridge Regel definiert haben. Informationen zum Arbeiten mit EventBridge und Systems Manager Manager-Ereignissen finden Sie in den folgenden Themen. • Konfiguration EventBrid ge für Systems Manager Manager-Ereignisse • Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager	4. November 2020

Für CloudWatch die Erstellung anhand OpsItems von Alarmen konfigurieren	Sie können Amazon so konfigurieren CloudWatch , dass automatisch ein OpsItem Systems Manager erstellt wirdOpsCenter, wenn ein Alarm in den `ALARM` Status wechselt. Auf diese Weise können Sie Probleme mit AWS Ressourcen von einer einzigen Konsole aus schnell diagnostizieren und beheben. Weitere Informationen finden Sie unter Konfiguration für CloudWatch die Erstellung OpsItems aus Alarmen.	4. November 2020
Unterstützung für Ubuntu Server 20.10	AWS Systems Manager unterstützt jetzt Ubuntu Server 20.10 Short-Term Release (STR). Weitere Informationen finden Sie unter den folgenden Themen: • Unterstützte Betriebssysteme • Installieren von SSM Agent für eine Hybrid-Umgebung (Linux) • Manuelle Installation von SSM Agent auf Ubuntu Server-Instances • Prüfen des SSM Agent-Status und Starten des Agenten	22. Oktober 2020

Neues Thema: Konfigurierbare Shell-Profile zulassen	Sie können jetzt konfiguri erbare Shell-Profile mit Session Manager zulassen. Indem Sie konfigurierbare Shell-Profile zulassen, können Sie Voreinstellungen innerhalb von Sitzungen wie Shell-Einstellungen, Umgebungs variablen, Arbeitsverzeichnis sen und Ausführen mehrerer Befehle beim Starten einer Sitzung anpassen. Weitere Informationen finden Sie unter Konfigurierbare Shell-Profile zulassen.	21. Oktober 2020
Patch-Compliance-Ergebnisse berichten nun, welche CVEs mit welchen Patches aufgelöst werden	Wenn Sie bei den meisten unterstützten Linux-Systemen Patch-Compliance-Ergebnisse für Ihre verwalteten Instances anzeigen, geben die Details, die Sie jetzt sehen können, an, welche Bulletin-Probleme mit Common Vulnerabilities and Exposure (CVE) durch welche verfügbaren Patches behoben werden. Mithilfe dieser Informationen können Sie feststellen, wie dringend Sie einen fehlenden oder fehlgeschlagenen Patch installieren müssen. Weitere Informationen finden Sie unter Anzeigen von Patch-Com pliance-Ergebnissen.	20. Oktober 2020

Erweiterte Unterstützung für Linux-Patch-Metadaten	Sie können jetzt viele Details zu verfügbaren Linux-Patches in Patch Manager aufrufen. Sie können Patch-Daten wie Architektur, Epoche, Version, CVE-ID, Advisory ID, Bugzilla ID, Repository und mehr anzeigen. Darüber hinaus wurde die API-Operation DescribeAvailablePatches aktualisiert, um Linux-Bet riebssysteme und Filterung gemäß diesen neu verfügbar en Patch-Metadatentypen zu unterstützen. Weitere Informati onen finden Sie unter den folgenden Themen: • Anzeigen verfügbarer Patches • DescribeAvailablePatches und Patch in der AWS Systems Manager -API-Refe renz • describe-available-patchesim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz	16. Oktober 2020
Session ManagerPlugin für die AWS CLI Version 1.2.7.0	Eine neue Version des Session Manager Plugins für AWS CLI wurde veröffentlicht. Weitere Informationen finden Sie unter Aktuelle Version und Versionsverlauf des Session Manager-Plug-Ins.	15. Oktober 2020

Neues Thema: Schema des Sitzungsdokuments	Das neue Thema Schema des Sitzungsdokuments beschreibt die Schemaelemente für ein Sitzungsdokument. Mithilfe dieser Informationen können Sie benutzerdefinierte Sitzungsdokumente erstellen, in denen Sie Einstellungen für die Sitzungstypen festlegen, die Sie mit Session Manager verwenden.	15. Oktober 2020
Neues Thema: Freitextsuche für SSM-Dokumente	Das Suchfeld auf der Seite Systems Manager-Dokumente unterstützt jetzt die freie Textsuche. Die Freitextsuche vergleicht den bzw. die eingegebenen Suchbegriffe mit dem Dokumentnamen in jedem SSM-Dokument. Weitere Informationen finden Sie unter Verwenden Der Freitextsuche.	15. Oktober 2020
Neues Thema: Fehlerbehebung bei der Verfügbarkeit verwalteter Amazon EC2-Instances	Das neue Thema Fehlerbehebung bei der Verfügbarkeit verwalteter Amazon EC2-Instances hilft Ihnen zu untersuchen, warum eine Amazon EC2-Instance, deren Ausführung Sie bestätigt haben, nicht in Listen verfügbarer verwalteter Instances in Systems Manager verfügbar ist.	6. Oktober 2020

Neuorganisation des Kapitels Parameter Store	Um Ihnen dabei zu helfen, die benötigten Informationen effizienter zu finden, haben wir die Inhalte im Kapitel Parameter Store des AWS Systems Manager -Benutzer handbuch neuorganisiert. Die meisten Inhalte sind jetzt in den Abschnitten Einrichte n vonParameter Store und Arbeiten mit Parameter Store organisiert. Darüber hinaus wurde das Thema AWS Systems ManagerParameter Store um folgende Unterabsc hnitte erweitert: • Welche Vorteile bietet Parameter Store meiner Organisation? • An wen richtet sich Parameter Store? • Über welche Features verfügt Parameter Store? • Was ist ein Parameter?	1. Oktober 2020

Neue Themen im Zusammenhang mit Patch-Compliance	Die folgenden Themen wurden hinzugefügt, um Ihnen zu helfen, verwaltete Instances zu identifizieren, die keine Patch-Compliance haben, die verschiedenen Arten von Patch-Compliance-Scans zu verstehen und die entsprechenden Schritte zu ergreifen, damit Ihre Instances die Compliance erfüllen. • Identifizierung nicht konformer Instances • Patchen nicht konformer Instances • Anzeigen der Patch-Compliance-Ergebnisse	24. September 2020
SSM Agent Version 3.0	Systems Manager hat eine neue Version von SSM Agent veröffentlicht.	21. September 2020

Neue und aktualisierte Themen: Amazon EventBridge ersetzt CloudWatch Events für Eventmanagement	CloudWatch Events und EventBridge sind derselbe zugrunde liegende Service und dieselbe API, EventBridge bieten aber mehr Funktionen und sind jetzt die bevorzugte Methode zur Verwaltung Ihrer Veranstaltungen in AWS. (Änderungen, die Sie in einer der beiden CloudWatch oder in jeder Konsole vornehmen, EventBridge spiegeln sich in jeder Konsole wider.) Die Verweise auf CloudWatch Ereignisse und bestehende Verfahren im gesamten AWS Systems Manager Benutzerhandbuch wurden aktualisiert, um der EventBridge Unterstützung Rechnung zu tragen. Darüber hinaus wurden die folgenden neuen Themen hinzugefügt. • Überwachen von Systems Manager-Ereignissen • Konfiguration EventBridge für Systems Manager Manager-Ereignisse • Beispiele für Zieltypen von Systems Manager • Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager	18. September 2020

Integrieren AWS Security Hub und Patch Manager	Sie können jetzt integrieren Patch Manager mit AWS Security Hub. Security Hub bietet einen umfassenden Überblick über Ihren Sicherheitsstatus AWS und hilft Ihnen dabei, Ihre Umgebung anhand von Industriestandards und Best Practices zu überprüfen. Bei der Integration in Patch Manager überwacht Security Hub den Sicherheitsaspekt des Patching-Status Ihrer Flotten. Weitere Informationen finden Sie unter Integration Patch Manager mit AWS Security Hub.	17. September 2020

Pseudo-Parameter des Wartungsfensters: Neue Ressourcentypen werden für `{{TARGET_ID}}` und `{{RESOURCE_ID}}` unterstützt	Beim Registrieren einer Wartungsfenster-Aufgabe geben Sie mithilfe der Option `--task-invocation-parameters` die Parameter an, die für jede der vier Arten von Aufgaben eindeutig sind. Sie können auch mithilfe der Pseudoparameter-Syntax wie `{{TARGET_ID}}` und `{{RESOURCE_ID}}` auf bestimmte Werte verweisen. Während der Ausführung übergibt die Wartungsfenster-Aufgabe anstelle der Pseudoparameter-Platzhalter richtige Werte. Zwei zusätzliche Ressourcentypen stehen nun für die Verwendung mit den Pseudo-Parametern `{{TARGET_ID}}` und `{{RESOURCE_ID}}` zur Verfügung. Sie können jetzt die Ressourcentypen `AWS::RDS::DBInstance` und `AWS::SSM::ManagedInstance` mit diesen beiden Pseudo-Parametern verwenden. Hinweise zu Pseudo-Parametern für Wartungsfenster finden Sie unter Pseudo-Parameter bei der Registrierung von Wartungsfensteraufgaben verwenden.	14. September 2020

Patchen von Instances auf Abruf mit der neuen Option „Patch now“ (Jetzt patchen)	Sie können jetzt die Systems Manager-Konsole verwenden, um Instances zu patchen oder nach fehlenden Patches zu suchen. Sie können dies tun, ohne einen Zeitplan erstellen oder ändern zu müssen, oder vollständige Patch-Konfigurationsoptionen angeben, um einem sofortigen Patch-Bedarf gerecht zu werden. Sie müssen nur angeben, ob Patches gescannt oder installiert werden sollen, und müssen die Ziel-Instances angeben. Patch Manager wendet automatisch die aktuelle Standard-Patch-Baseline für Ihre Instance-Typen an und wendet bewährte Methoden dazu, wie viele Instances gleichzeitig gepatcht werden und wie viele Fehler zulässig sind, bevor der Vorgang fehlschlägt. Weitere Informationen finden Sie unter Instances auf Abruf patchen .	9. September 2020

Neues Thema: Prüfen des SSM Agent-Status und Starten des Agenten	Das neue Thema Prüfen des SSM Agent-Status und Starten des Agentenbietet Befehle, um zu überprüfen, ob SSM Agent auf jedem unterstützenden Betriebssystem ausgeführt wird. Es enthält auch die Befehle, mit denen der Agent gestartet wird, wenn er nicht ausgeführt wird.	7. September 2020
Patch Manager unterstützt jetzt Ubuntu Server 20.04 LTS	Sie können jetzt Patch Manager verwenden, um Ubuntu Server 20.04 LTS-Instances zu patchen. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Ubuntu Server	31. August 2020
Neues Thema für häufige Anwendungsfälle und bewährte Methoden	Wir haben ein neues Thema hinzugefügt, um den Benutzern schnell die Unterschiede zwischen Maintenance Windows und State Manager zu vermitteln. Weitere Informationen finden Sie unter Wählen zwischen State Manager und Maintenance Windows.	28. August 2020

Neue OpsCenter-Features	OpsCenter enthalten neue Features, mit denen Sie Automatisierungs-Runbooks schnell finden und ausführen können, um Probleme zu beheben. Weitere Informationen finden Sie unter Automatisierungs-Runbook-Funktionen in OpsCenter.	19. August 2020
Neue Datenquelle in: Fällen ExplorerAWS Support	Explorerzeigt jetzt Informationen über AWS Support Fälle an. Sie müssen entweder ein Unternehmens- oder ein Geschäftskonto mit eingerichtet haben AWS Support. Weitere Informationen finden Sie unter Bearbeiten von Systems Manager-Datenquellen.	13. August 2020
Distributor bietet jetzt ein Drittanbieter-Paket von Trend Micro.	Distributor enthält jetzt ein Drittanbieter-Paket von Trend Micro. Sie können Distributor verwenden, um den Trend Micro Cloud One Agent auf Ihren verwalteten Instances zu installieren. Trend Micro Cloud One hilft Ihnen, Ihre Workloads in der Cloud zu sichern. Weitere Informationen finden Sie unter AWSDistributor.	12. August 2020

Der `aws:configurePackage` -Dokument-Plugin enthält jetzt den Parameter AdditionalArguments.	Das Systems Manager Command-Dokument-Plugin `aws:configurePackage` unterstützt jetzt die Bereitstellung zusätzlicher Parameter für Ihre Skripte (Installation, Deinstallation und Update) mit dem neuen `additionalArguments` -Parameter. Weitere Informationen finden Sie im Thema `aws:configurePackage` .	11. August 2020
AppConfigDer -Inhalt wurde in ein separates Benutzerhandbuch verschoben	Informationen zu AWS AppConfig wurden in ein separates Benutzerhandbuch verschoben. Weitere Informationen finden Sie unter Was ist AWSAppConfig? AppConfig hat auch eine separate Landingpage zur Dokumentation mit Links zum Benutzerhandbuch, zur AppConfig API-Referenz und zu einem neuen AppConfig Workshop.	3. August 2020

Quick Setupunterstützt jetzt AWS Organizations	Quick Setupunterstützt jetzt AWS Organizations die schnelle Konfigura tion der erforderlichen Sicherheitsrollen und häufig verwendeten Systems Manager Manager-Funktionen für mehrere Konten und Regionen. Weitere Informati onen finden Sie unter AWS Systems ManagerQuick Setup.	23. Juli 2020
Neue Datenquelle inExplorer: Zuordnungs-Compliance	Explorer zeigt jetzt Zuordnung s-Compliance-Daten von State Manager an. Weitere Informationen finden Sie unter Bearbeiten von Systems Manager-Datenquellen.	23. Juli 2020

Neues Systems Manager-Befehlsdokument zum Ein- und Ausschalten von Kernel Live Patching	Das Dokument `AWS-ConfigureKernelLivePatching` ist nun für die Verwendung mit Run Command verfügbar, wenn Sie Kernel Live Patching auf Amazon Linux 2-Instances ein- oder ausschalten möchten. Dieses Dokument ersetzt die Notwendigkeit, eigene benutzerdefinierte Befehlsdokumente für diese Aufgaben zu erstellen. Weitere Informationen finden Sie unter Verwenden von Kernel-Live-Patching auf Amazon Linux 2-Instances	22. Juli 2020
Aktualisierte Automation-Kontingente	Service quotas für Automation wurden aktualisiert, einschließlich einer separaten Warteschlange für die Automatisierung der Ratenregelung. Weitere Informationen finden Sie unter AWS Systems Manager Automation.	20. Juli 2020

Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster mithilfe der Konsole	Über die Systems Manager-Konsole können Sie nun eine Anzahl von Tagen angeben, die nach dem in einem CRON-Ausdruck angegebenen Datum und der angegebenen Uhrzeit gewartet werden soll, bevor ein Wartungsfenster ausgeführt wird. (Bisher war diese Option nur verfügbar , wenn Sie ein AWS SDK oder ein Befehlszeilentool verwendeten.) Wenn Ihr CRON-Ausdruck beispiels weise die Ausführung eines Wartungsfensters am dritten Dienstag jedes Monats um 23:30 Uhr plant – `cron(0 30 23 ? * TUE#3 *)` – und Sie einen Zeitplanversatz von 2 angeben, wird das Fenster erst zwei Tage später um 23:30 Uhr ausgeführt. Weitere Informationen finden Sie unter Cron- und Rate-Ausd rücke für Systems Manager und Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster.	17. Juli 2020

Aktualisierung PowerShell mit Run Command	Um Ihnen bei der Aktualisierung PowerShell auf Version 5.1 auf Ihren Windows Server 2012- und 2012 R2-Instances zu helfen, haben wir dem AWS Systems Manager Benutzerhandbuch eine exemplarische Vorgehensweise hinzugefügt. Weitere Informationen finden Sie unter Update PowerShell mit Run Command.	30. Juni 2020
Patch Manager unterstützt jetzt CentOS 8.0 und 8.1	Sie können jetzt Patch Manager verwenden, um CentOS 8.0- und 8.1-Instances zu patchen. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Wie Patch-Baseline-Regeln auf CentOS funktionieren • Manuelle Installation von SSM Agent auf CentOS-Instances • Wie installiert man das SSM Agent auf Hybrid-Linux-Knoten	27. Juni 2020

AppConfigintegriert mit AWS CodePipeline	AppConfigist eine integrierte Bereitstellungsaktion für AWS CodePipeline (CodePipeline). CodePipeline ist ein vollständig verwalteter Continuous Delivery Service, der Sie bei der Automatisierung Ihrer Release-Pipelines für schnelle und zuverlässige Anwendungs- und Infrastrukturupdates unterstützt. CodePipelineautomatisiert die Erstellungs-, Test- und Bereitstellungsphasen Ihres Release-Prozesses bei jeder Codeänderung auf der Grundlage des von Ihnen definierten Release-Modells. Die Integration von AppConfig mit CodePipeline bietet die folgenden Vorteile. Weitere Informationen finden Sie unter AppConfigIntegration mit CodePipeline. • Kunden, die früher CodePipeline die Orchestrierung verwalteten, verfügen jetzt über eine einfache Möglichkeit, Konfigurationsänderungen an ihren Anwendungen vorzunehmen, ohne ihre gesamte Codebasis bereitstellen zu müssen. • Kunden, die AppConfig verwenden möchten, um	25. Juni 2020

	Konfigurationsbereitstellungen zu verwalten, jedoch nur begrenzt handeln können, da AppConfig ihren aktuellen Code oder Konfigurationsspeicher nicht unterstützt, haben jetzt zusätzliche Optionen. CodePipeline unterstützt AWS CodeCommitGitHub, und BitBucket (um nur einige zu nennen).	
Neues Kapitel: Produkt- und Service-Integrationen	Damit Sie besser verstehen , wie Systems Manager mit AWS-Services anderen Produkten und Services integriert werden kann, wurde dem AWS Systems Manager Benutzerhandbuch ein neues Kapitel hinzugefügt. Weitere Informationen finden Sie unter Produkt- und Service-Integrationen mit Systems Manager.	23. Juni 2020

Umstrukturierung des Automation-Kapitels	Damit Sie leichter finden, was Sie suchen, haben wir Themen im Automation-Kapitel des AWS Systems Manager - Benutzerhandbuch umstrukturiert. Beispielsweise sind die Automation-Aktionen und -Runbooks jetzt Abschnitte der obersten Ebene des Kapitels. Weitere Informationen finden Sie unter AWS Systems Manager Automation.	23. Juni 2020

Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster	Mit einem Befehlszeilentool oder AWS SDK können Sie jetzt angeben, wie viele Tage nach dem in einem CRON-Ausdruck angegebenen Datum und Uhrzeit gewartet werden sollen, bevor ein Wartungsfenster ausgeführt wird. Wenn Ihr CRON-Ausdruck beispielsweise die Ausführung eines Wartungsfensters am dritten Dienstag jedes Monats um 23:30 Uhr plant – `cron(0 30 23 ? * TUE#3 *)` – und Sie einen Zeitplanversatz von 2 angeben, wird das Fenster erst zwei Tage später um 23:30 Uhr ausgeführt. Weitere Informationen finden Sie unter Cron- und Rate-Ausdrücke für Systems Manager und Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster.	19. Juni 2020

Patch Manager-Unterstützung für Kernel-Live-Patching auf Amazon Linux 2-Instances	Kernel-Live-Patching für Amazon Linux 2 ermöglicht es Ihnen, Patches für Schwachstellen und kritische Fehler auf einen laufenden Linux-Kernel anzuwenden, ohne Neustarts oder Unterbrechungen der laufenden Anwendungen. Sie können jetzt das Feature aktivieren und Kernel-Live-Patches mithilfe von Patch Manager anwenden. Informationen finden Sie unter Verwenden von Kernel-Live-Patching auf Amazon Linux 2-Instances.	16. Juni 2020
Patch Manager erhöht die Oracle Linux-Versionsunterstützung	Bisher hat Patch Manager nur Version 7.6 von Oracle Linux unterstützt. Wie in den Patch Manager-Voraussetzungen aufgeführt, deckt der Support jetzt die Versionen 7.5 - 7.8 ab.	16. Juni 2020

Beispielszenario für die Verwendung des `InstallOverrideList` -Parameters in Patch-Operationen	Das neue Thema Beispielszenario für die Verwendung des Parameters `InstallOverrideList` beschreibt eine Strategie für die Verwendung des Parameters `InstallOverrideList` im Dokument `AWS-RunPatchBaseline` , um verschiedene Patch-Typen auf eine Zielgruppe in verschiedenen Wartungsfenstern bei Verwendung einer einzelnen Patch-Baseline anzuwenden.	11. Juni 2020
Vordefinierte Bereitstellungsstrategien für AppConfig	AppConfig bietet jetzt vordefinierte Bereitstellungsstrategien. Weitere Informationen finden Sie unter Erstellen einer Bereitstellungsstrategie.	10. Juni 2020

Patch Manager unterstützt jetzt Red Hat Enterprise Linux (RHEL) 7.8-8.2	Sie können Patch Manager jetzt zum Patchen von RHEL 7.8–8.2-Instances verwenden . Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf RHEL • Manuelle Installation von SSM Agent auf Red Hat Enterprise Linux-Instances • Wie installiert man das SSM Agent auf Hybrid-Linux-Knoten	9. Juni 2020

Explorer unterstützt das Delegieren der Administration	Wenn Sie Explorer Daten aus mehreren zusammenf ühren AWS-Regionen und AWS-Konten die Ressource ndatensynchronisierung mit verwenden AWS Organizat ions, empfehlen wir Ihnen, einen delegierten Administr ator für Explorer zu konfiguri eren. Ein delegierter Administr ator verbessert die Explorer-Sicherheit, indem die Anzahl der Explorer-Administratoren, die die Synchronisierung von Ressourcendaten für mehrere Konten und Regionen erstellen oder löschen können, auf eine einzelne Person eingeschr änkt wird. Sie müssen auch nicht mehr am AWS Organizat ions -Hauptkonto angemeldet sein, um die Synchronisierung von Ressourcendaten in Explorer zu verwalten. Weitere Informationen finden Sie unter Konfigurieren eines delegiert en Administrators.	3. Juni 2020

State Manager-Zuordnung erst für das nächste angegebene Cron-Intervall übernehmen	Wenn Sie nicht möchten, dass eine State Manager-Zuordnung unmittelbar nach der Erstellung ausgeführt wird, wählen Sie die Option Apply association only at the next specified Cron interval (Zuordnung erst beim nächsten angegebenen Cron-Intervall anwenden) in der Systems Manager-Konsole aus. Weitere Informationen finden Sie unter Erstellen von Zuordnungen.	3. Juni 2020
Neue Datenquelle inExplorer: AWS Compute Optimizer	Explorerzeigt jetzt Daten von an AWS Compute Optimizer . Hierzu gehören die Zahl der EC2-Instances, die Under provisioned (Zu wenig bereitgestellt) und Over provisioned (Zu viel bereitgestellt) wurden, Optimierungsergebnisse, Details zu On-Demand-Preisen und Empfehlungen für Instance-Typ und Preis. Weitere Informationen finden Sie AWS Compute Optimizer in den Details zur Einrichtung unter Verwandte Dienste einrichten.	26. Mai 2020

Neues Kapitel: Markieren von Systems Manager-Ressourcen	Das neue Kapitel Markieren von Systems Manager-Ressourcen bietet eine Übersicht, wie Sie Tags mit den sechs markierbaren Ressourcentypen in Systems Manager verwenden können. Das Kapitel enthält auch umfassende Anweisungen zum Hinzufügen von Tags zu den folgenden Ressource ntypen und ihre Entfernung: • -Documents • Wartungsfenster • Verwaltete Instances • OpsItems • Parameter • Patch-Baselines	25. Mai 2020

Installieren von Windows Service Packs und Linux-Nebenversionsupgrades mit Patch Manager	Das neue Thema Tutorial: Erstellen einer Patch-Baseline für die Installation von Windows Service Packs (Konsole) zeigt, wie Sie eine Patch-Baseline erstellen, die ausschließlich der Installation von Windows Service Packs gewidmet ist. Das Thema Erstellen einer benutzerdefinierten Patch-Baseline (Linux) wurde mit Informationen zum Einschluss von Nebenversionsupgrades für Linux-Betriebssysteme in Patch-Baselines aktualisiert.	21. Mai 2020
Neuorganisation des Kapitels Parameter Store	Alle Themen, die sich mit Konfigurations- oder Einrichtungsoptionen für Parameter Store-Operationen befassen, wurden im Abschnitt Einrichtung von Parameter Store zusammengefasst. Dies umfasst auch die Themen Verwalten von Parameterstufen und Erhöhen des Parameter Store-Durchsatzes, die aus anderen Teilen des Kapitels hierher verschoben wurden.	18. Mai 2020

Neues Thema zum Erstellen von Datums- und Uhrzeitzeichenfolgen für die Interaktion mit Systems Manager-API-Operationen.	Im neuen Thema Erstellen formatierter Datums- und Uhrzeitzeichenfolgen für Systems Manager wird die Erstellung formatierter Datums- und Uhrzeitzeichenfolgen für die Interaktion mit Systems Manager-API-Operationen beschrieben.	13. Mai 2020
Informationen zu Berechtigungen zum Verschlüsseln von Parametern SecureString	Im neuen Thema Beschränken des Zugriffs auf Systems Manager Manager-Parameter mithilfe von IAM-Richtlinien wird der Unterschied zwischen der Verschlüsselung Ihrer `SecureString` Parameter mit AWS KMS key und der Von AWS verwalteter Schlüssel Verwendung von bereitgestellt von erklärt. AWS	13. Mai 2020

Patch Manager unterstützt jetzt Debian Server und Oracle Linux 7.6 Betriebssysteme	Sie können jetzt Patch Manager verwenden, um Debian Server- und Oracle Linux-Instances zu patchen. Patch Managerunterstützt Patchen der Versionen Debian Server 8.x und 9.x und Oracle Linux 7.6. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Debian Server • Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux	7. Mai 2020
Erstellen Sie Zuordnungen, State Manager die auf Folgendes abzielen AWS Resource Groups	Zusätzlich zur Ausrichtung auf Tags, einzelne Instances und alle Instances in Ihrem AWS-Konto können Sie jetzt State Manager-Zuordnungen erstellen, die Instances in AWS Resource Groups zum Ziel haben. Weitere Informationen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager-Zuordnungen.	7. Mai 2020

Neuer `aws:ec2:image` - Datentyp in Parameter Store zum Validieren von AMI-IDs	Wenn Sie einen `String`-Parameter erstellen, können Sie jetzt einen Datentyp als `aws:ec2:image` angeben, um sicherzustellen, dass der eingegebene Parameterwert ein gültiges Amazon Machine Image (AMI)-ID-Format aufweist. Durch die Unterstützung von AMI-ID-Formaten können Sie vermeiden, dass alle Skripts und Vorlagen jedes Mal mit einer neuen ID aktualisiert werden, wenn sich das AMI ändert, das Sie in Ihren Prozessen verwenden möchten. Sie können einen Parameter mit dem Datentyp `aws:ec2:image` erstellen und für seinen Wert die ID eines AMI eingeben. Dies ist das AMI, von dem Sie neue Instances erstellen möchten. Sie verweisen dann in Ihren Vorlagen und Befehlen auf diesen Parameter. Wenn Sie ein anderes AMI verwenden möchten, aktualisieren Sie den Parameterwert. Parameter Store validiert die neue AMI-ID und Sie müssen Ihre Skripts und Vorlagen nicht aktualisieren. Weitere Informationen finden Sie unter Native Parameterunterstützung für Amazon Machine Image-IDs.	5. Mai 2020

Verwalten von Beendigungscodes in Run Command-Befehlen	In Run Command können Sie definieren, wie Beendigungscodes in Ihren Skripts gehandhabt werden. Standardmäßig wird der Beendigungscode des letzten in einem Skript ausgeführten Befehls als Beendigungscode für das gesamte Skript gemeldet. Sie können jedoch mit der folgenden Methode eine bedingte Shell-Anweisung einschließen, damit das Skript beendet wird, wenn ein Befehl vor dem letzten Befehl fehlschlägt. Beispiele finden Sie im neuen Thema Verwalten von Beendigungscodes in Run Command-Befehlen.	5. Mai 2020

Neue öffentliche Parameter für Availability Zones und lokale Zonen freigegeben	Öffentliche Parameter wurden veröffentlicht, um Informationen zu AWS Availability Zones und lokalen Zonen programmgesteuert verfügbar zu machen. Diese Parameter ergänzen die bestehenden öffentlichen Parameter der globalen Infrastruktur für AWS-Services und AWS-Regionen. Weitere Informationen finden Sie unter Öffentliche Parameter aufrufen für Regionen AWS-Services, Endpunkte, Availability Zones, Local Zones und Wavelength Zones.	4. Mai 2020
Neue Datenquelle inExplorer: AWS Trusted Advisor	Explorerzeigt jetzt Daten von an AWS Trusted Advisor. Dies umfasst den Status von Überprüfungen zu bewährten Methoden und Empfehlungen in den folgenden Bereichen: Kostenoptimierung, Sicherheit, Fehlertoleranz, Leistung und Service Quotas. Weitere Informationen finden Sie Trusted Advisor in den Details zur Einrichtung unter Verwandte Dienste einrichten.	4. Mai 2020

Erstellen Sie State Manager Verknüpfungen, die Chef Rezepte ausführen	Mithilfe des `AWS-ApplyChefRecipes` Dokuments können Sie State Manager Verknüpfungen erstellen, die Chef Kochbücher und Rezepte ausführen. Dieses Dokument bietet die folgenden Vorteile beim Ausführen von Chef Rezepten: • Unterstützt mehrere Versionen von Chef (Chef11 bis Chef 14). • Installiert die Chef Client-Software automatisch auf Zielinstanzen. • Führt optional Systems Manager-Compliance-Prüfungen für Ziel-Instances aus und speichert die Ergebnisse der Compliance-Prüfungen in einem S3-Bucket. • Führt mehrere Cookbooks und Rezepte in einem einzigen Durchlauf des Dokuments aus. • Führt optional Rezepte im `why-run`-Modus aus, um anzuzeigen, welche Rezepte sich auf Ziel-Instances ändern, ohne Änderungen vorzunehmen. • Wendet optional benutzerdefinierte JSON-Attribute auf	19. März 2020

	`chef-client` -Durchläufe an. Weitere Informationen finden Sie unter Verknüpfungen erstellen, die Chef Rezepte ausführen	
Synchronisieren Sie Inventardaten von mehreren AWS-Konten zu einem zentralen Amazon S3 S3-Bucket	Sie können Systems Manager Manager-Inventardaten von mehreren AWS-Konten zu einem zentralen S3-Bucket synchronisieren. Die Konten müssen in definiert sein AWS Organizations. Weitere Informationen finden Sie unter Erstellen einer Inventory Resource Data Sync für mehrere Konten, die in AWS Organizations definiert sind.	16. März 2020
Speichern von AppConfig-Konfigurationen in Amazon S3	Zuvor unterstützte AppConfig nur Anwendungskonfigurationen, die in Systems Manager-(SSM-)Dokumenten oder Parameter Store-Parametern gespeichert wurden. Zusätzlich zu diesen Optionen unterstützt AppConfig jetzt das Speichern von Konfigurationen in Amazon S3. Weitere Informationen finden Sie unter Informationen über Konfigurationen in Amazon S3.	13. März 2020

SSM Agent ist standardmäßig auf Amazon ECS-optimierten installiertAMIs	SSM Agent ist jetzt standardmäßig auf Amazon ECS-optimierten AMIs installiert. Weitere Informationen finden Sie unter Arbeiten mit SSM Agent.	25. Februar 2020
Erstellen von AppConfig-Konfigurationen in der Konsole	Mit AppConfig können Sie jetzt bei Erstellung eines Konfigurationsprofils eine Anwendungskonfiguration in der Konsole erstellen. Weitere Informationen finden Sie unter Erstellen einer Konfiguration und eines Konfigurationsprofils.	13. Februar 2020

Nur Patches automatisch genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden	Neben der Option, Patches eine bestimmte Anzahl von Tagen nach der Veröffentlichung automatisch für die Installation zu genehmigen, unterstützt Patch Manager nun die Möglichkeit, nur Patches automatisch zu genehmigen, die an oder vor einem von Ihnen angegebenen Datum veröffentlicht wurden. Wenn Sie beispielsweise den 7. Juli 2020 als Stichtag in Ihrer Patch-Baseline angeben, werden keine Patches automatisch installiert, die an oder nach dem 8. Juli 2020 veröffentlicht wurden. Weitere Informationen finden Sie unter Informationen zu benutzerdefinierten Baselines und Arbeiten mit benutzerdefinierten Patch-Baselines (Konsole).	12. Februar 2020

Verwenden Sie den Pseudoparameter {{RESOURCE_ID}} in Wartungsfensteraufgaben	Geben Sie beim Registrieren einer Aufgabe im Wartungsfenster die Parameter an, die für den Aufgabentyp eindeutig sind. Sie können mithilfe der Pseudoparameter-Syntax wie `{{TARGET_ID}}`, `{{TARGET_TYPE}}` und `{{WINDOW_TARGET_ID}}` auf bestimmte Werte verweisen. Während der Ausführung übergibt die Wartungsfenster-Aufgabe anstelle der Pseudoparameter-Platzhalter richtige Werte. Zur Unterstützung von Ressourcen, die als Ziel Teil einer Ressourcengruppe sind, können Sie den `{{RESOURCE_ID}}`-Pseudoparameter verwenden, um Werte für Ressourcen wie DynamoDB-Tabellen, S3-Buckets und andere unterstützte Typen zu übergeben. Weitere Informationen finden Sie in den folgenden Themen in der Anleitung: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI): • Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfenstern	6. Februar 2020

	• Beispiele: Registrieren von Aufgaben für ein Wartungsfenster	
Befehle schnell erneut ausführen	Systems Manager enthält zwei Optionen, mit denen Sie einen Befehl von der Run CommandSeite in der AWS Systems Manager Konsole erneut ausführen können. Rerun (Erneut ausführen): Über diese Schaltfläche können Sie denselben Befehl ausführen, ohne Änderungen daran vorzunehmen. Copy to new (In neu kopieren): Über diese Schaltfläche kopieren Sie die Einstellungen eines Befehls in einen neuen Befehl und erhalten die Möglichkeit, diese Einstellungen zu bearbeiten, bevor Sie den Befehl ausführen. Weitere Informationen finden Sie unter Befehle erneut ausführen.	5. Februar 2020

Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances	Wenn Sie zuvor alle On-Premises-Instances, die in Ihrer Hybrid-Umgebung ausgeführt werden, für die Verwendung des Kontingents für erweiterte Instances konfiguriert haben, können Sie diese Instances nun schnell so konfigurieren, dass sie das Kontingent für Standard-Instances verwenden. Die Rückkehr zur Stufe „Standard instanzen“ gilt für alle Hybrid-Instances in einer und einer AWS-Konto einzigen. AWS-Region Das Zurücksetzen auf Kontingent für Standard-Instances wirkt sich auf die Verfügbarkeit einiger Systems Manager-Funktionen aus. Weitere Informationen finden Sie unter Zurücksetzen des Kontingents für erweiterte Instances auf das Kontingent für Standard-Instances.	16. Januar 2020

Neue Option zum Überspringen von Instance-Neustarts nach der Patch-Installation	Zuvor wurden verwaltete Instances immer neu gestartet, nachdem der Patch Manager Patches auf ihnen installiert hatte. Mit einem neuen `RebootOption` -Parameter im SSM-Dokument `AWS-RunPatchBaseline` können Sie angeben, ob die Instances nach der Installation neuer Patches automatisch neu gestartet werden sollen. Weitere Informationen finden Sie unter Parametername: RebootOption im Thema Über das SSM-Dokument. `AWS-RunPatchBaseline`	15. Januar 2020
Neues Thema: „ PowerShell Skripts auf Linux-Instanzen ausführen“	Ein neues Thema, das beschreibt, wie Sie PowerShell Skripts auf Run Command Linux-Instances ausführen können. Weitere Informationen finden Sie unter PowerShell Skripts auf Linux-Instances ausführen.	10. Januar 2020

Aktualisierungen zum ‚Konfigurieren von SSM Agent zur Nutzung eines Proxys‘	Die Werte, die bei der Konfiguration von SSM Agent zur Nutzung eines Proxys anzugeben sind, wurden aktualisiert, um die Optionen sowohl für HTTP-Proxyserver als auch für HTTPS-Pro xyserver wiederzugeben. Weitere Informationen finden Sie unter Konfigurieren von SSM Agent zur Nutzung eines Proxys.	9. Januar 2020

Das neue Kapitel „Sicherheit“ beschreibt Verfahren zur Sicherung von Systems Manager-Ressourcen	Im neuen Kapitel Security (Sicherheit) im AWS Systems Manager User Guide (Benutzerhandbuch von) wird erläutert, wie Sie das shared responsibility model (Modell der geteilten Verantwortung) (Modell der geteilten Verantwortung) (Modell der geteilten Verantwortung) beim Einsatz von Systems Manager anwenden können. Die Themen in diesem Kapitel zeigen Ihnen, wie Sie Systems Manager konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie lernen auch, wie Sie andere verwenden können AWS-Services , die Ihnen helfen, Ihre Systems Manager Manager-Ressourcen zu überwachen und zu sichern. **Note** Als Teil dieses Updates wurde das Kapitel „Authentifizierung und Zugriffskontrolle“ im Benutzerhandbuch durch einen neuen, einfacheren Abschnitt Identity and Access Managemen	24. Dezember 2019

t für AWS Systems Manager ersetzt.

Neue eigene Automation-Beispielrunbooks	Eine Reihe von eigenen Automation-Beispielrunbooks wurde dem Benutzerhandbuch hinzugefügt. Diese Beispiele zeigen, wie verschiedene Automation-Aktionen verwendet werden können, um die Bereitstellung, Fehlerbehebung und Wartungsaufgaben zu vereinfachen, und sollen Ihnen helfen, Ihre eigenen benutzerdefinierten Automation-Runbooks zu schreiben. Weitere Informationen finden Sie unter Beispiele für benutzerdefinierte Automation-Runbooks. Sie können den Inhalt des von Amazon verwalteten Automation-Runbooks auch in der Systems Manager-Konsole anzeigen. Weitere Informationen finden Sie unter Referenz zu Systems Manager Automation.	23. Dezember 2019

Support für Oracle Linux	Systems Manager unterstützt jetzt Oracle Linux 7.5 und 7.7. Informationen zur manuellen Installation von SSM Agent auf EC2-Instances für Oracle Linux-Instances finden Sie unter Oracle Linux. Informationen zur Installation SSM Agent auf Oracle Linux Servern in einer Hybridumgebung finden Sie unter So installieren Sie den SSM-Agent auf Hybrid-Linux-Knoten.	19. Dezember 2019

Starten von Session Manager-Sitzungen von der Amazon EC2-Konsole	Sie können Session Manager-Sitzungen jetzt von der Amazon Elastic Compute Cloud (Amazon EC2)-Konsole aus starten. Die Arbeit mit sitzungsbezogenen Aufgaben über die Amazon EC2-Konsole erfordert unterschiedliche IAM-Berechtigungen für Benutzer und Administratoren. Sie können Berechtigungen für die Verwendung der Session Manager Konsole und AWS CLI nur für die Verwendung der Amazon EC2 EC2-Konsole oder für die Verwendung aller drei Tools bereitstellen. Weitere Informationen finden Sie unter den folgenden Themen. • Schnellstart: Standard IAM-Richtlinien für Session Manager • Starten einer Sitzung (Amazon EC2-Konsole)	18. Dezember 2019

CloudWatch Unterstützung für Run Command Metriken und Alarme	AWS Systems Manager veröffentlicht jetzt Metriken über den Status von Run Command Befehlen an CloudWatch, sodass Sie Alarme auf der Grundlage dieser Metriken einrichten können. Zu den Terminals tatuswerten für Befehle, für die Sie Metriken verfolgen können, gehören `Success`, `Failed` und `Delivery Timed Out`. Weitere Informationen finden Sie unter Run CommandMetriken mit Amazon überwachen CloudWatch.	17. Dezember 2019
Neue Systems Manager-Funktion: Change Calendar	Verwenden Sie Systems Manager Change Calendar, um Zeiträume (Ereignisse) anzugeben, in denen Sie Codeänderungen (z. B. von Systems Manager Automation-Runbooks oder AWS Lambda -Funktionen) an Ressourcen einschränken oder verhindern möchten. Ein Änderungskalender ist ein neuer Systems Manager-Dokumenttyp, der iCalendar 2.0-Daten im Klartextformat speichert. Weitere Informationen finden Sie unter AWS Systems Manager Change Calendar.	11. Dezember 2019

Neue Systems Manager Manager-Funktion: AWSAppConfig	Mit AppConfig können Sie Anwendungskonfigurationen schnell erstellen, verwalten und bereitstellen. AppConfig unterstützt kontrollierte Bereitstellungen für Anwendungen jeder Größe. Sie können es AppConfig mit Anwendungen verwenden, die auf EC2-Instances, Containern AWS Lambda, mobilen Anwendungen oder IoT-Geräten gehostet werden. Um Fehler bei der Bereitstellung von Anwendungskonfigurationen zu vermeiden, enthält AppConfig Validierungen. Eine Validierung stellt durch eine syntaktische oder semantische Prüfung sicher, dass die Konfiguration, die Sie bereitstellen möchten, wie beabsichtigt funktioniert. Während einer Konfigurationsbereitstellung überwacht AppConfig die Anwendung, um sicherzustellen, dass die Bereitstellung erfolgreich ist. Falls im System ein Fehler auftritt oder die Bereitstellung einen Alarm startet, setzt AppConfig die Änderung zurück, um die Auswirkungen auf die Benutzer Ihrer Anwendung zu minimieren.	25. November 2019

Weitere Informationen finden Sie unter AWSAppConfig.

Neue Systems-Manager-Funktion: Systems Manager Explorer	AWS Systems Manager Explorer ist ein anpassbares Operations-Dashboard, das Informationen über Ihre AWS Ressourcen meldet. Explorerzeigt eine aggregierte Ansicht der Betriebsdaten (OpsData) für Sie AWS-Konten und Across AWS-Regionen an. OpsData Enthält Metadaten zu Ihren EC2-Instances, Details zur Patch-Konformität und betriebliche Arbeitselemente ()OpsItems. Explorer Explorerbietet Informationen darüber, wie sie auf Ihre Geschäftsbereiche oder Anwendungen verteilt OpsItems sind, wie sie sich im Laufe der Zeit entwickeln und wie sie sich je nach Kategorie unterscheiden. Sie können Informationen in Explorer gruppieren und filtern, um sich auf die Elemente zu konzentrieren, die für Sie relevant sind und eine Aktion erfordern. Wenn Sie Probleme mit hoher Priorität erkennen, können Sie mit Systems Manager OpsCenter Automation-Runbooks ausführen und die Probleme schnell beheben. Weitere Informationen finden Sie unter AWS Systems ManagerExplorer.	18. November 2019

> **Note**
>
> Die Einrichtung für Systems Manager OpsCenter ist in die Einrichtung für Explorer integriert. Wenn Sie OpsCenter bereits eingerichtet haben, müssen Sie das integrierte Setup trotzdem ausführen, um die Einstellungen und Optionen zu überprüfen. Wenn Sie OpsCenter nicht eingerichtet haben, können Sie mit dem integrierten Setup beide Funktionen einrichten. Weitere Informationen finden Sie unter Erste Schritte mit Explorer und OpsCenter.

Verbesserte Parameter suchfunktionen	Mit den Werkzeugen für die Parametersuche können Sie Parameter jetzt leichter finden, wenn Sie viele Parameter in Ihrem Konto haben oder sich nicht an den genauen Namen eines Parameters erinnern. Mit dem Suchwerkzeug können Sie nach `contains` filtern. Zuvor unterstützten die Suchwerkzeuge die Suche nach Parameternamen nur mit `equals` und `begins-with` . Weitere Informationen finden Sie unter Suche nach öffentlichen Systems Manager-Parametern.	15. November 2019

Neuer konsolenbasierter Document Builder für Automation | Unterstützung für die Ausführung von Skripts in Automation-Schritten	Sie können Systems Manager Automation jetzt verwenden, um standardisierte Betriebsp laybooks zu erstellen und gemeinsam zu nutzen, um die Konsistenz zwischen Benutzern AWS-Konten, und AWS-Regionen sicherzus tellen. Mit der Möglichkeit, Skripts auszuführen und Ihren Automation-Runbooks mit Markdown eingebundene Dokumentation hinzuzufügen, können Sie Fehler reduziere n und manuelle Schritte wie das Navigieren in schriftlicher Prozeduren in Wikis und das Ausführen von Terminalb efehlen eliminieren. Weitere Informationen finden Sie unter den folgenden Themen. • Walkthrough: Verwenden von Document Builder zum Erstellen eines benutzerd efinierten Automatisierungs-Runbooks • `aws:executeScript` (Referenz zu Automation-Aktionen) • Erstellen von Automation-Runbooks mit Document Builder	14. November 2019

	• Neue Automation-Funktionen in Systems Manager im AWS News-Blog	
Ausführen einer direkten Paketaktualisierung mit Distributor	Wenn Sie früher ein Update für ein Paket mit Distributor installieren wollten, bestand Ihre einzige Möglichkeit darin, das gesamte Paket zu deinstallieren und die neue Version neu zu installieren. Jetzt können Sie stattdessen eine direkte Aktualisierung durchführen. Während einer direkten Aktualisierung installiert Distributor entsprechend dem Aktualisierungsskript, das Sie Ihrem Paket hinzufügen, nur Dateien, die neu sind oder seit der letzten Installation geändert wurden. Bei Verwendung dieser Option muss Ihre Paketanwendung während der Aktualisierung nicht offline geschaltet werden, sodass sie weiterhin verfügbar ist. Weitere Informationen finden Sie unter den folgenden Themen. • Erstellen eines Pakets • Installieren oder Aktualisieren von Paketen	11. November 2019

Neues Feature für automatische SSM Agent-Aktualisierungen	Mit einem Klick können Sie alle Instanzen in Ihrem so konfigurieren, AWS-Konto dass automatisch nach neuen Versionen von SSM Agent gesucht und diese heruntergeladen werden. Wählen Sie dazu auf der Seite Verwaltete Instanzen in der AWS Systems Manager Konsole die Option auto Agentenaktualisierung aus. Weitere Informationen finden Sie unter Automatische Aktualisierungen von SSM Agent.	5. November 2019

Beschränken Sie Session Manager den Zugriff mithilfe der von AWS Ihnen bereitgestellten Tags	Eine zweite Methode zur Steuerung des Benutzerzugriffs auf Sitzungsaktionen ist jetzt verfügbar. Mit dieser neuen Methode können Sie IAM-Zugriffsrichtlinien mithilfe von AWS-bereitgestellten Sitzungs-Tags erstellen, anstatt die Variable `{aws:username}` zu verwenden. Durch die Verwendung dieser von AWS-bereitgestellten Sitzungs-Tags können Organisationen, die Federated IDs verwenden, den Benutzerzugriff auf Sitzungen steuern. Weitere Informationen finden Sie unter Benutzer können nur von ihnen gestartete Sitzungen beenden.	2. Oktober 2019

Neues SSM-Befehlsdokument zum Anwenden von Playbooks Ansible	Mithilfe des Dokuments können Sie State Manager Verknüpfungen erstellen, die Ansible Playbooks ausführen . `AWS-ApplyAnsiblePlaybooks` Dieses Dokument bietet die folgenden Vorteile für die Ausführung von Playbooks: • Unterstützung für die Ausführung komplexer Playbooks • Support für das Herunterladen von Playbooks von GitHub und Amazon Simple Storage Service (Amazon S3) • Unterstützung für komprimierte Playbook-Struktur • Erweiterte Protokollierung • Möglichkeit, anzugeben , welches Playbook ausgeführt werden soll, wenn Playbooks gebündelt werden Weitere Informationen finden Sie unter Verknüpfungen erstellen, die Playbooks ausführen Ansible	24. September 2019

Unterstützung der Port-Weiterleitung für Session Manager	Session Manager unterstützt jetzt Port-Weiterleitungssitzungen. Die Port-Weiterleitung ermöglicht es Ihnen, Tunnel zwischen Ihren in privaten Subnetzen bereitgestellten Instances sicher zu erstellen, ohne den SSH-Service auf dem Server zu starten, den SSH-Port in der Sicherheitsgruppe zu öffnen oder einen Bastion-Host zu verwenden. Ähnlich wie bei SSH-Tunneln ermöglicht Ihnen die Port-Weiterleitung, Datenverkehr auf Ihrem Laptop weiterzuleiten, um Ports auf Ihrer Instance zu öffnen. Sobald die Port-Weiterleitung konfiguriert ist, können Sie eine Verbindung mit dem lokalen Port herstellen und auf die Serveranwendung zugreifen, die in der Instance ausgeführt wird. Weitere Informationen finden Sie unter den folgenden Themen: • Port-Weiterleitung mit AWS Systems ManagerSession Manager im AWS News Blog • Starten einer Sitzung (Port-Weiterleitung)	29. August 2019

Festlegen einer Standardparameterebene oder Automatisieren der Stufefenauswahl	Sie können jetzt eine Standardparameterebene angeben, die für Anforderungen zum Erstellen oder Aktualisieren eines Parameters verwendet werden soll, die keine Stufe angeben. Sie können die Standardstufe auf Standardparameter, erweiterte Parameter oder eine neue Option, Intelligent-Tiering, festlegen. Intelligent-Tiering wertet jede PutParameter Anfrage aus und erstellt nur bei Bedarf einen erweiterten Parameter. (Erweiterte Parameter sind erforderlich, wenn die Größe des Parameterwerts mehr als 4 KB beträgt, dem Parameter eine Parameterrichtlinie zugeordnet ist oder die maximal 10.000 für die Standardstufe unterstützten Parameter bereits erstellt wurden.) Weitere Informationen zum Angeben einer Standardstufe und zur Verwendung von Intelligent-Tiering finden Sie unter Angeben einer Standardparameterstufe.	27. August 2019

Der Abschnitt „Arbeiten mit Assoziationen“ wurde mit CLI und PowerShell Verfahren aktualisiert	Der Abschnitt „Mit Assoziationen arbeiten“ wurde aktualisiert und enthält nun eine Verfahrensdokumentation für die Verwaltung von Verknüpfungen mithilfe von AWS CLI oder AWS Tools for PowerShell. Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.	26. August 2019
Der Abschnitt „Arbeiten mit Automatisierungsausführungen“ wurde mit CLI und PowerShell Verfahren aktualisiert	Der Abschnitt „Mit Automatisierungsausführungen arbeiten“ wurde aktualisiert und enthält nun eine prozedurale Dokumentation für die Ausführung von Automatisierungsworkflows mit dem AWS CLI oder. AWS Tools for PowerShell Weitere Informationen finden Sie unter Arbeiten mit Automation-Ausführungen.	20. August 2019

OpsCenter lässt sich in Application Insights integrieren	OpsCenterlässt sich in Amazon CloudWatch Application Insights für .NET und SQL Server integrieren. Das bedeutet, dass Sie automatisch OpsItems für Probleme erstellen können, die in Ihren Anwendungen erkannt wurden. Informationen zur Konfiguration von Application Insights zum Erstellen OpsItems finden Sie unter Einrichten, Konfigurieren und Verwalten Ihrer Anwendung für die Überwachung im CloudWatch Amazon-Benutzerhandbuch.	7. August 2019

Neue Konsolenfunktion: AWS Systems ManagerQuick Setup	Quick Setup ist ein neues Feature in der Systems Manager-Konsole, mit dem Sie mehrere Systems Manager-Komponenten auf Ihren EC2-Instances schnell konfiguri eren können. Mit Quick Setup können Sie insbesondere die folgenden Komponenten auf den Instances konfiguri eren, die Sie mithilfe von Tags auswählen oder als Ziel festlegen: • Eine AWS Identity and Access Management (IAM-) Instanzprofilrolle für Systems Manager. • Eine geplante, zweimonat liche Aktualisierung von SSM Agent. • Eine geplante Sammlung von Inventory-Metadaten alle 30 Minuten. • Ein täglicher Scan Ihrer Instances, um fehlende Patches zu identifizieren. • Eine einmalige Installat ion und Konfiguration des CloudWatch Amazon-Ag enten. • Ein geplantes, monatlich es Update des CloudWatch Agenten.	7. August 2019

Weitere Informationen finden Sie unter AWS Systems Manager Quick Setup.

Registrieren einer Ressourcengruppe als Ziel eines Wartungsfensters	Neben der Registrierung verwalteter Instanzen als Ziel eines Wartungsfensters können Sie jetzt auch eine Ressourcengruppe als Ziel für ein Wartungsfenster registrieren. Maintenance Windowsunterstützt alle AWS Ressourcentypen, die AWS Resource Groups vo nAWS::EC2::Instance , AWS::DynamoDB::Table AWS::OpsWorks::Instance AWS::Redshift::Cluster , und mehr unterstützt werden. Mit dieser Version können Sie auch Befehle an eine Ressourcengruppe senden, z. B. mithilfe der Run Command Konsole oder des AWS CLI send-commandBefehls. Weitere Informationen finden Sie unter den folgenden Themen: • Zuweisen von Zielen zu einem Wartungsfenster (Konsole) • Beispiele: Registrieren von Zielen für ein Wartungsfenster • Verwenden von Zielen und Häufigkeitskontrollen zum Senden von Befehlen an eine Gruppe von Instances	23. Juli 2019

Vereinfachte Paketerstellung und Versionierung mit AWS Systems ManagerDistributor	Distributor verfügt über eine neuen, vereinfachten Workflow zur Paketerstellung, der für ein Paket das Manifest, Skripts und Datei-Hashes generiere n kann. Sie können auch den vereinfachte Workflow nutzen, wenn Sie einem bereits vorhandenen Paket eine Version hinzufügen.	22. Juli 2019
Bereich „New Document categories (Neue Dokumentk ategorien)“ für Systems Manager Automation	Systems Manager umfasst einen Bereich „New Document categories (Neue Dokumentk ategorien)“, der angezeigt wird, wenn Sie in der Konsole eine Automatisierung ausführen. Verwenden Sie diesen Bereich zum Filtern der Automatisierungs-Runbooks nach Zweck.	18. Juli 2019

Überprüfen der Berechtigungen eines Benutzers für den Zugriff auf das Session Manager-Standardkonfigurationsdokument	Wenn ein Benutzer in Ihrem Konto das verwendet AWS CLI , um eine Session Manager Sitzung zu starten, und im Befehl kein Konfigurationsdokument angibt, verwendet Systems Manager das Standardkonfigurationsdokument`SSM-SessionManagerRunShell` . Sie können jetzt überprüfen, ob dem Benutzer die Berechtigung zum Zugriff auf dieses Dokument erteilt wurde, indem Sie der Richtlinie für `ssm:SessionDocumentAccessCheck` das ein Bedingungselement für hinzufügen.AWS Identity and Access Management (IAM) - Entität (Benutzer, Gruppe oder Rolle). Weitere Informationen finden Sie unter Durchsetzen von Überprüfungen der Dokumentberechtigungen beim Standard-CLI-Szenario.	9. Juli 2019

Unterstützung für das Starten von Session Manager-Sitzungen mit Betriebssystem-Anmeldeinformationen	Standardmäßig werden Session Manager-Sitzungen unter Verwendung der Anmeldeinformationen eines vom System generierten `ssm-user`-Kontos gestartet werden, der auf einer verwalteten Instance erstellt wird. Auf Linux-Computern können Sie jetzt stattdessen Sitzungen unter Verwendung der Anmeldeinformationen für ein Betriebssystemkonto starten. Weitere Informationen finden Sie unter Aktivieren der Run As-Unterstützung für Linux-Instances.	9. Juli 2019
Unterstützung für das Starten von Session Manager-Sitzungen mit SSH	Sie können den jetzt verwenden AWS CLI , um eine SSH-Sitzung auf einer verwalteten Instanz zu starten, indem Sie. Session Manager Weitere Informationen zur Aktivierung von SSH-Sitzungen mit Session Manager finden Sie unter (Optional) Aktivieren von SSH-Session Manager-Sitzungen. Weitere Informationen zum Starten einer SSH-Sitzung mit Session Manager finden Sie unter Starten einer Sitzung (SSH).	9. Juli 2019

Unterstützung für das Ändern von Passwörtern auf verwalteten Instances	Sie können jetzt Passwörter auf Computern zurücksetzen, die Sie mit Systems Manager verwalten (verwaltete Instances). Sie können das Passwort über die Systems Manager-Konsole oder die AWS CLI zurücksetzen. Weitere Informationen finden Sie unter Zurücksetzen von Passwörtern auf verwalteten Instances.	9. Juli 2019
Änderungen von „Was ist AWS Systems Manager?“	Die einführenden Inhalte in Was ist AWS Systems Manager? wurden erweitert, um eine breitere Einführung in den Service zu bieten und kürzlich veröffentlichten Systems Manager-Funktionen Rechnung zu tragen. Darüber hinaus wurden andere Inhalte in einzelne Themen verschoben, sodass sie leichter auffindbar sind.	10. Juni 2019

Neue Systems Manager-Funktion: OpsCenter	OpsCenterbietet einen zentralen Ort, an dem Betriebsingenieure und IT-Experten betriebliche Arbeitsaufgaben (OpsItems) im Zusammenhang mit AWS Ressourcen einsehen, untersuchen und lösen können. OpsCenter wurde entwickelt, um die durchschnittliche Zeit bis zur Lösung von Problemen zu reduzieren, die sich auf AWS Ressourcen auswirken . Diese Systems Manager-Funktion aggregiert und standardisiert OpsItems über Services hinweg und bietet gleichzeitig kontextbezogene Untersuchungsdaten über jedes OpsItem, verwandte OpsItems und verwandte Ressourcen. OpsCenter bietet außerdem Systems-Automatisierungsdokumente (Runbooks), die Sie verwenden können, um Probleme zu lösen. Sie können durchsuchbare, benutzerdefinierte Daten für jedes OpsItem angeben. Sie können auch automatisch generierte Zusammenfassungsberichte über OpsItems nach Status und Quelle anzeigen. Weitere Informationen finden Sie unter	6. Juni 2019

	AWS Systems ManagerOpsCenter.	
Änderungen am linken Navigationsbereich von Systems Manager im AWS Management Console	Der linke Navigationsbereich von Systems Manager im AWS Management Console enthält neue Überschriften, darunter eine neue Überschrift für Ops Center, die eine logischere Gruppierung der Systems Manager Manager-Funktionen ermöglichen.	6. Juni 2019

Überarbeitete Anleitung zum Erstellen und Konfigurieren eines Wartungsfensters mithilfe der AWS CLI	Tutorial: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI) wurde überarbeitet, um den Pfad durch die Übungsschritte zu vereinfachen. Sie erstellen ein einziges Wartungsfenster, identifizieren ein einziges Ziel und richten eine einfache Aufgabe ein, die im Wartungsfenster ausgeführt werden soll. Dabei stellen wir Informationen und Beispiele zur Verfügung, mit denen Sie eigene Befehle zur Aufgabenregistrierung erstellen können. Dazu zählen auch Informationen zur Verwendung von Pseudoparametern wie `{{TARGET_ID}}` . Zusätzliche Informationen und Beispiele finden Sie in den folgenden Themen: • Beispiele: Registrieren von Zielen für ein Wartungsfenster • Beispiele: Registrieren von Aufgaben für ein Wartungsfenster • Informationen zu den Optionen von register-task-with-maintenance -windows • Verwendung von Pseudo-Parametern bei der Registrierung von Wartungsfensteraufgaben	31. Mai 2019

Benachrichtigungen über SSM Agent-Updates	Um über SSM Agent Updates informiert zu werden, abonnieren Sie die Seite mit den SSM AgentVersionshinweisen unterGitHub.	24. Mai 2019
Erhalten von Benachrichtigungen oder Auslösen von Aktionen basierend auf Änderungen in Parameter Store	Das Thema Benachrichtigungen einrichten oder Aktionen auf der Grundlage von Parameter Store Ereignissen auslösen hilft Ihnen jetzt dabei, EventBridge Amazon-Regeln einzurichten, um auf Änderungen in zu reagierenParameter Store. Sie können Benachrichtigungen erhalten oder andere Aktionen auslösen, wenn eines der folgenden Ereignisse eintritt: • Ein Parameter wird erstellt, aktualisiert oder gelöscht. • Eine Parameterbezeichnungsversion wird erstellt, aktualisiert oder gelöscht. • Ein Parameter läuft ab, läuft in Kürze ab oder wurde in einem angegebenen Zeitraum nicht geändert.	22. Mai 2019

Hauptrevidierungen an Inhalten zu Einrichtung oder ersten Schritten	Wir haben die Inhalte zu Einrichtung und Erste Schritte im AWS Systems Manager - Benutzerhandbuch erweitert und neu organisiert. Die Inhalte zur Einrichtung wurden in zwei Abschnitte unterteilt. Ein Abschnitt konzentriert sich auf Aufgaben für das Einrichten von Systems Manager zum Konfigurieren und Verwalten von EC2-Instances. Der andere konzentriert sich auf Aufgaben für das Einrichten von Systems Manager zum Konfigurieren und Verwalten Ihrer On-Premises-Server und virtuellen Maschinen (VMs) in einer Hybrid-Umgebung. Beide Abschnitte präsentieren jetzt alle Einrichtungsthemen als wesentliche, nummerierte Schritte, in der empfohlenen Reihenfolge der Durchführung. Ein neues Kapitel Erste Schritte ist darauf ausgelegt, Endbenutzern bei den ersten Schritte mit Systems Manager zu helfen, nachdem Konto- und Service-Konfigurationsaufgaben abgeschlossen sind. • Einrichtung AWS Systems Manager	15. Mai 2019

- Einrichtung AWS Systems Manager für Hybridumgebungen
- Erste Schritte mit AWS Systems Manager

Patches für von Microsoft veröffentlichte Anwendungen jetzt in Patch-Baselines (Windows) enthalten	Patch Manager unterstützt jetzt Patch-Updates für von Microsoft veröffentlichte Anwendungen auf Windows Server-Instances. Bisher wurden nur Patches für das Windows Server-Betriebssystem unterstützt. Patch Managerbietet zwei vordefinierte Patch-Baselines für Windows Server-Instances. Die Patch-Baseline `AWS-WindowsPredefinedPatchBaseline-OS` gilt nur für Betriebssystem-Patches. `AWS-WindowsPredefinedPatchBaseline-OS-Applications` gilt sowohl für das Windows Server-Betriebssystem als auch für von Microsoft veröffentlichte Anwendungen unter Windows. Weitere Informationen zum Erstellen einer benutzerdefinierten Patch-Baseline, die Patches für von Microsoft veröffentlichte Anwendungen enthält, finden Sie im ersten Verfahren unter Erstellen einer benutzerdefinierten Patch-Baseline. Im Rahmen dieses Updates werden auch die Namen der AWS bereitgestellten vordefinierten Patch-Baselines geändert. Weitere Informati	7. Mai 2019

	onen finden Sie unter Vordefinierte Baselines.	
Beispiele für die Registrierung von Wartungsfensterzielen mit dem AWS CLI	Das neue Thema über Beispiele: Registrieren von Zielen für ein Wartungsfenster bietet drei Beispielbefehle, um verschiedene Möglichkeiten zu zeigen, wie Sie die Ziele für ein Wartungsfenster angeben können, wenn Sie die AWS CLI verwenden. Darüber hinaus wird in diesem Thema der beste Anwendungsfall für jeden der Beispielbefehle erläutert.	3. Mai 2019

Updates für Themen für Patch-Gruppen	Das Thema Patch-Gruppen wurde aktualisiert und enthält nun einen Abschnitt darüber, wie verwaltete Instances die entsprechende Patch-Baseline bestimmen, die während der Patch-Vorgänge zu verwenden ist. Darüber hinaus wurden Anweisungen hinzugefügt, wie Sie mithilfe der AWS CLI oder der Systems Manager Manager-Konsole Patchgrup pen oder PatchGroupTags zu Ihren verwalteten Instances hinzufügen können und wie Sie eine Patchgruppe oder PatchGroupeine Patch-Bas eline hinzufügen. (Sie müssen **`PatchGroup`** ohne Leerzeich en verwenden, wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben.) Weitere Informationen finden Sie unter Erstellen einer Patch-Gruppe und Hinzufügen einer Patch-Gruppe zu einer Patch-Bas eline.	1. Mai 2019

Neue Parameter Store-Funktionen	Parameter Store bietet die folgenden neuen Features: • Erweiterte Parameter : Mit Parameter Store können Sie jetzt Parameter einzeln konfigurieren, um entweder ein Standardparameterkontingent (das Standardkontingent) oder ein Kontingent für erweiterte Parameter zu verwenden. Erweiterte Parameter bieten ein größeres Kontingent für den Parameterwert, ein höheres Kontingent für die Anzahl der Parameter, die Sie pro AWS-Konto und erstellen können AWS-Region, und die Möglichkeit, Parameterrichtlinien zu verwenden. Weitere Informationen über erweiterte Parameter finden Sie unter Über erweiterte Parameter von Systems Manager. • Parameterrichtlinien: Parameterrichtlinien unterstützen Sie bei der Verwaltung einer wachsenden Menge von Parametern, indem Sie einem Parameter bestimmte Kriterien zuweisen können, wie etwa Ablaufdatum	25. April 2019

oder Time to Live (Gültigkeitsdauer). Parameterrichtlinien sind besonders hilfreich, da sie Sie zwingen, in Parameter Store gespeicherte Passwörter und Konfigurationsdaten zu aktualisieren oder zu löschen. Parameterrichtlinien sind nur verfügbar für Parameter, die das Kontingent für erweiterte Parameter verwenden. Weitere Informationen finden Sie im Artikel zum Arbeiten mit Parameterrichtlinien.

- Höherer Durchsatz: Sie können jetzt das Durchsatzkontingent von Parameter Store auf ein Maximum von 1 000 Transaktionen pro Sekunde erhöhen. Weitere Informationen finden Sie im Artikel zum Erhöhen des Parameter Store-Durchsatzes.

Aktualisierungen des Abschnitts über Automatisierungen	Der Abschnitt über Automatisierungen wurde aktualisiert und ist nun übersichtlicher. Darüber hinaus wurden dem Abschnitt drei neue Themen hinzugefügt: • Führen Sie eine Automatisierung manuell aus • Eine Automatisierung mit Genehmigern ausführen • Planung von Automatisierungen	17. April 2019

Verschlüsseln Sie Sitzungsdaten mit einem Schlüssel AWS KMS	Standardmäßig verwendet Session Manager TLS 1.2 zum Verschlüsseln von Sitzungsdaten, die zwischen lokalen Computern von Benutzern in Ihrem Konto und Ihren EC2-Instances übertragen werden. Jetzt können Sie wählen, ob Sie diese Daten mit einem, der in erstellt wurde AWS KMS key , weiter verschlüsseln möchten. AWS Key Management Service Sie können einen KMS-Schlüssel verwenden, der in Ihrem AWS-Konto erstellt wurde, oder einen Schlüssel , der von einem anderen Konto für Sie freigegeben wurde. Informationen zum Angeben eines KMS-Schlüssels zum Verschlüsseln von Sitzungsdaten finden Sie unter Aktivieren der AWS KMS Schlüsselverschlüsselung von Sitzungsdaten (Konsole) , Session ManagerEinstellungen erstellen (AWS CLI) oder Session ManagerEinstellungen aktualisieren (AWS CLI).	4. April 2019

Konfiguration von Amazon SNS SNS-Benachrichtigungen für AWS Systems Manager	Es wurden Anweisungen zur Verwendung der AWS CLI oder Systems Manager Manager-Konsole zur Konfiguration von Amazon SNS SNS-Benachrichtigungen Run Command und Run Command Aufgaben hinzugefügt, die für ein Wartungsfenster registriert sind. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Benachrichtigungen für AWS Systems Manager.	6. März 2019

Erweiterte Instances für Server und virtuelle Maschinen in hybriden Umgebungen	AWS Systems Manager bietet eine Stufe „Standard-Instances“ und eine Stufe „Advanced Instances“ für Server und VMs in Ihrer Hybridumgebung. Die Stufe „Standard-Instances“ ermöglicht es Ihnen, maximal 1.000 Server oder VMs pro Person zu registrieren. AWS-Konto AWS-Region Wenn Sie mehr als 1.000 Server oder VMs in einem einzigen Konto und einer Region registrieren müssen, verwenden Sie das Advanced-Instances-Kontingent. In der Advanced-Instance-Stufe können Sie beliebig viele Instanzen erstellen, aber alle für Systems Manager konfigurierten Instanzen sind auf einer pay-per-use Basis verfügbar. Mit erweiterten Instanzen können Sie auch eine Verbindung zu Ihren Hybrid-Computern herstellen, indem Sie AWS Systems Manager Session Manager Session Managerbietet interaktiven Shell-Zugriff auf Ihre Instanzen. Weitere Informationen zum Aktivieren von erweiterten Instances finden Sie im Artikel zum Verwenden des Kontingents für erweiterte Instances.	4. März 2019

Erstellen von State Manager-Zuordnungen, die freigegebene SSM-Dokumente verwenden	Sie können State Manager Verknüpfungen erstellen, die SSM-Command and Automation-Runbooks verwenden, die von anderen gemeinsam genutzt werden. AWS-Konten Das Erstellen von Zuordnungen durch das Verwenden freigegebener SSM-Dokumente hilft Ihnen, Amazon EC2 und die Hybrid-Infrastruktur in einem konsistenten Zustand zu halten, auch wenn sich diese Instances nicht in demselben Konto befinden. Weitere Informationen zur Freigabe von SSM-Dokumenten finden Sie unter AWS Systems Manager -Dokumente. Weitere Informationen zum Erstellen einer State Manager-Zuordnung finden Sie unter Erstellen einer Zuordnung.	28. Februar 2019
Listen der Systems Manager Manager-Ereignisse anzeigen, die für EventBridge Amazon-Regeln unterstützt werden	Das neue Thema Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge bietet eine Zusammenfassung der verschiedenen von Systems Manager ausgegebenen Ereignisse, für die Sie Regeln zur Ereignisüberwachung einrichten können EventBridge.	25. Februar 2019

Hinzufügen von Tags beim Erstellen von Systems Manager-Ressourcen	Systems Manager unterstützt jetzt die Möglichkeit, bestimmten Ressourcentypen Tags hinzuzufügen, wenn Sie sie erstellen. Zu den Ressourcen, die Sie taggen können, wenn Sie sie mit dem AWS CLI oder einem SDK erstellen, gehören Wartungsfenster, Patch-Baselines, Parameter Store Parameter und SSM-Dokumente. Sie können auch einer verwalteten Instance Tags zuweisen, wenn Sie eine Aktivierung dafür erstellen. Wenn Sie die Systems Manager-Konsole verwenden, können Sie Wartungsfenstern, Patch-Baselines und Parametern Tags hinzufügen.	24. Februar 2019

Automatische IAM-Rollenerstellung für Systems Manager-Inventory	Bisher mussten Sie eine AWS Identity and Access Management (IAM-) Rolle erstellen und dieser Rolle separate Richtlinien zuordnen, um Inventardaten auf der Seite „Inventardetailansicht“ in der Konsole anzuzeigen. Es ist nicht mehr erforderlich, diese Rolle zu erstellen oder ihr Richtlinien zuzuweisen. Wenn Sie auf der Seite „Inventardetailansicht“ eine Remote-Datensynchronisierung wählen, erstellt Systems Manager automatisch die `Amazon-GlueServicePolicyForSSM` Rolle und weist ihr die Amazon- GlueService PolicyFor SSM- {S3 bucket name} -Richtlinie und die AWSGlueServiceRoleRichtlinie zu. Weitere Informationen finden Sie unter Abfragen von Bestandsdaten aus mehreren Regionen und Konten.	14. Februar 2019

Maintenance Windows-Anleitungen zum Aktualisieren von SSM Agent	Der Maintenance Windows-Dokumentation wurden zwei neue Anleitungen hinzugefügt. In den exemplarischen Vorgehensweisen wird detailliert beschrieben, wie Sie die Systems Manager Manager-Konsole verwenden oder ein Wartungsfenster erstellen, das automatisch beibehalten wird SSM Agent up-to-date . AWS CLI Weitere Informationen finden Sie im Artikel über Maintenance Windows-Anleitungen.	11. Februar 2019
Verwenden von öffentlichen Parameter Store-Parametern	Ein kurzer Abschnitt mit einer Beschreibung der öffentlichen Parameter Store-Parameter wurde hinzugefügt. Weitere Informationen finden Sie unter Verwenden von öffentlichen Systems Manager-Parametern .	31. Januar 2019

Verwenden Sie die, AWS CLI um Einstellungen zu erstellen Session Manager	Es wurden Anweisungen zur Verwendung von hinzugefügt, AWS CLI um Session Manager Einstellungen wie CloudWatch Protokolle, S3-Bucket-Protokollierungsoptionen und Einstellungen für die Sitzungsverschlüsselung zu erstellen. Weitere Informationen finden Sie unter Verwenden von, AWS CLI um Session Manager Einstellungen zu erstellen.	22. Januar 2019
Ausführen von Systems Manager-Automatisierungsworkflows mithilfe von State Manager	AWS Systems Manager State Managerunterstützt jetzt das Erstellen von Zuordnungen, die SSM Automation-Runbooks verwenden. State Managerbisher wurden nur `command` `policy` Dokumente unterstützt, was bedeutete, dass Sie nur Verknüpfungen erstellen konnten, die auf verwaltete Instanzen abzielten. Mit Unterstützung für SSM Automation-Runbooks können Sie jetzt Zuordnungen erstellen, die unterschiedliche Arten von AWS -Ressourcen zum Ziel haben. Weitere Informationen finden Sie im Artikel über das Ausführen von Systems Manager Automation-Workflows mithilfe von State Manager.	22. Januar 2019

Referenz-Updates für Cron- und Rate-Ausdrücke und Planungsoptionen für Wartungsfenster	Das Referenzthema Cron- und Rate-Ausdrücke für Systems Manager wurde überarbeitet. Die neue Version stellt weitere Beispiele und verbesserte Erläuterungen zur Verwendung von Cron- und Rate-Ausdrücken zum Planen Ihrer Wartungsfenster und State Manager-Zuordnungen bereit. Darüber hinaus wird im neuen Thema Maintenance Windows-Optionen für Planung und aktive Zeiträume erläutert, wie die verschiedenen Zeitplanoptionen für Wartungsfenster (Startdatum, Enddatum, Zeitzone, Zeitplanhäufigkeit) miteinander in Verbindung stehen.	6. Dezember 2018
Aktivieren von SSM Agent-Debugging-Protokollierung	Sie können die SSM Agent-Debug-Protokollierung aktivieren, indem Sie die Datei seelog.xml.template auf der verwalteten Instance bearbeiten. Weitere Informationen finden Sie unter Aktivieren der SSM Agent-Debug-Protokollierung.	30. November 2018

Unterstützung für ARM64-Prozessorarchitekturen	AWS Systems Manager unterstützt jetzt ARM64-Versionen der Betriebssysteme Amazon Linux 2, Red Hat Enterprise Linux 7.6 und Ubuntu Server (18.04 LTS und 16.04 LTS). Weitere Informationen finden Sie in den Anweisungen zum Installieren von Amazon Linux 2, RHEL und Ubuntu Server 18.04 und 16.04 LTS mit Snap-Paketen. Weitere Informationen zum Instance-Typ A1 finden Sie unter General Purpose Instances im Amazon EC2 EC2-Benutzerhandbuch.	26. November 2018

Erstellen und Bereitstellen von Paketen mithilfe von AWS Systems ManagerDistributor	Mithilfe von AWS Systems Manager Distributor Paketen Sie Ihre eigene Software — oder suchen nach AWS bereitgestellten Agent-Softwarepaketen, z. B. `AmazonCloudWatchAgent` —, um sie auf verwalteten Instanzen zu installieren. AWS Systems Manager Distributorveröffentlicht Ressourcen, z. B. Softwarepakete, auf verwalteten Instanzen. AWS Systems Manager Bei der Veröffentlichung eines Pakets werden spezifische Versionen des Dokuments dieses Pakets – ein Systems Manager-Dokument, das Sie erstellen, wenn Sie das Paket in Distributor hinzufügen – in verwalteten Instances veröffentlicht, die Sie über die IDs der verwalteten Instances, die IDs von AWS-Konto , Tags oder eine AWS-Region identifizieren. Weitere Informationen finden Sie unter AWS Systems ManagerDistributor.	20. November 2018

Führen Sie AWS Systems Manager Automatisierungs-Workflows gleichzeitig für mehrere Benutzer AWS-Regionen und AWS-Konten von einem zentralen Konto aus aus	Sie können AWS Systems Manager Automatisierungsworkflows gleichzeitig über mehrere AWS-Regionen AWS-Konten und/oder AWS Organisationseinheiten (OUs) von einem Automatisierungsverwaltungskonto aus ausführen. Die gleichzeitige Ausführung von Automatisierungen in mehreren Regionen und Konten oder OUs verkürzt die Zeit für die Bereitstellung Ihrer AWS -Ressourcen und verbessert die Sicherheit Ihrer Computingumgebung. Weitere Informationen finden Sie unter Automatisierungsworkflows in mehreren AWS-Regionen und AWS-Konten ausführen.	19. November 2018

Inventardaten von mehreren abfragen AWS-Regionen und AWS-Konten	Systems Manager Inventory ist in Amazon Athena integriert, sodass Sie Inventardaten von mehreren AWS-Regionen und AWS-Konten abfragen können. Die Athena-Integration verwendet die Ressourcendatensynchronisierung, sodass Sie Inventardaten von all Ihren verwalteten Instanzen auf der Seite „Inventardetailansicht“ in der AWS Systems Manager Konsole anzeigen können. Weitere Informationen finden Sie unter Abfragen von Bestandsdaten aus mehreren Regionen und Konten.	15. November 2018

Erstellen von State Manager-Zuordnungen, die MOF-Dateien ausführen	Sie können Managed Object Format (MOF)-Dateien ausführen, um einen gewünschten Status auf von Windows Server verwalteten Instances mit State Manager mithilfe des SSM-Dokuments `AWS-ApplyDSCMofs` zu erzwingen. Das `AWS-ApplyDSCMofs` -Dokument weist zwei Ausführungsmodi auf. Mit dem ersten Modus können Sie die Zuordnung so konfigurieren, dass sie scannt und meldet, wenn sich die verwalteten Instances derzeit in dem Zielstatus befinden, der in den angegebenen MOF-Dateien definiert ist. Im zweiten Modus können Sie die MOF-Dateien ausführen und die Konfiguration Ihrer Instances basierend auf den Ressourcen und ihren in den MOF-Dateien definierten Werten ändern. Mit dem `AWS-ApplyDSCMofs` -Dokument können Sie MOF-Konfigurationsdateien von Amazon Simple Storage Service (Amazon S3), einem lokal freigegebenen Verzeichnis, oder einer sicheren Website mit einer HTTPS-Domain herunterladen und ausführen. Weitere Informationen	15. November 2018

	finden Sie unter Erstellen von Zuordnungen, die MOF-Dateien ausführen.	
Einschränken des administrativen Zugriff in Session Manager-Sitzungen	Session Manager-Sitzungen werden unter Verwendung der Anmeldeinformationen für ein Benutzerkonto namens `ssm-user` gestartet, das mit dem Root- oder Administrator-Berechtigungen erstellt wurde. Informationen zum Einschränken der administrativen Kontrolle für dieses Konto sind jetzt im Thema Deaktivieren oder Aktivieren von administrativen Berechtigungen für das Konto ssm-user verfügbar.	13. November 2018
YAML-Beispiele in Automatisierungsaktionsreferenz	Die Automatisierungsaktionsreferenz enthält jetzt ein YAML-Beispiel für die einzelnen Aktionen, die bereits ein JSON-Beispiel enthalten.	31. Oktober 2018

Zuweisen von Compliance-Schweregraden zu Assoziationen	Sie können State Manager-Zuordnungen jetzt Compliance-Schweregrade zuweisen. Diese Schweregrade werden im Compliance-Dashboard gemeldet und können auch zum Filtern Ihrer Compliance-Berichte verwendet werden. Die Schweregrade, die Sie zuweisen können, sind Kritisch, Hoch, Mittel, Niedrig und Nicht angegeben. Weitere Informationen finden Sie unter Erstellen einer Zuordnung (Konsole).	26. Oktober 2018
Verwenden von Ziel- und Ratensteuerungen mit Automatisierung und State Manager	Steuern Sie die Ausführung von Automatisierungen und State Manager-Zuordnungen innerhalb Ihrer Flotte von Ressourcen mithilfe von Zielen, Gleichzeitigkeit und Fehlerschwellenwerten. Weitere Informationen finden Sie unter Verwenden von Ziel- und Ratensteuerungen für die Ausführung von Automatisierungsworkflows zu einer Flotte und Verwenden von Zielen und Ratensteuerungen bei State Manager-Zuordnungen.	23. Oktober 2018

Festlegen von aktiven Zeitbereichen und internationalen Zeitzonen für Wartungsfenster	Sie können auch Daten festlegen, vor oder nach denen Wartungsfenster nicht ausgeführt werden sollte (Start- und Enddatum), und Sie können die internationale Zeitzone als Grundlage für den Wartungsfenster-Zeitplan festlegen. Weitere Informationen finden Sie unter Erstellen eines Wartungsfensters (Konsole) und Aktualisieren eines Wartungsfensters (AWS CLI).	9. Oktober 2018
Führen einer benutzerdefinierten Liste mit Patches für Ihre Patch-Baseline in einem S3-Bucket	Mit dem neuen Parameter 'InstallOverrideList' im SSM-Befehlsdokument `AWS-RunPatchBaseline` können Sie eine HTTPS-URL oder eine URL im Pfadstil von Amazon Simple Storage Service (Amazon S3) für eine Liste von zu installierenden Patches angeben. Diese in einem S3-Bucket im YAML-Format geführte Patch-Installationsliste überschreibt die von der Standard-Patch-Baseline angegebenen Patches. Weitere Informationen finden Sie unter Parametername:. InstallOverrideList	5. Oktober 2018

Erweiterte Kontrolle über die Installation von Patch-Abhängigkeiten	Wenn ein Patch zuvor in Ihrer Liste für abgelehnte Patches als Abhängigkeit eines anderen Patches identifiziert wurde, wäre es trotzdem installiert worden. Jetzt können Sie wählen, ob Sie diese Abhängigkeiten installieren möchten oder nicht. Weitere Informationen finden Sie unter Erstellen einer Patch-Baseline.	5. Oktober 2018
Erstellen dynamischer Automatisierungsworkflows mit bedingten Verzweigungen	Die `aws:branch` -Automatisierungsaktion ermöglicht das Erstellen eines dynamischen Automatisierungsworkflows, der verschiedene Auswahlmöglichkeiten in einem einzigen Schritt evaluiert und dann auf der Grundlage dieser Evaluierung zu einem anderen Schritt in dem Automatisierungs-Runbook springt. Weitere Informationen finden Sie unter Verwendung bedingter Anweisungen in Runbooks.	26. September 2018

Verwenden Sie die AWS CLI , um die Session Manager Einstellungen zu aktualisieren	Anweisungen zur Verwendung der CLI zur Aktualisierung von Session Manager Einstellungen wie CloudWatch Logs und S3-Bucket-Logging-Optionen wurden dem AWS Systems Manager Benutzerhandbuch hinzugefügt. Weitere Informationen finden Sie unter Verwenden von, AWS CLI um Session Manager Einstellungen zu aktualisieren.	25. September 2018
Aktualisierte SSM Agent-Voraussetzung für Session Manager	Session Manager erfordert jetzt SSM Agent Version 2.3.68.0 oder höher. Weitere Informationen zu Session Manager-Voraussetzungen finden Sie unter Abschließen der Session Manager-Voraussetzungen.	17. September 2018

Verwalten von Instances, ohne eingehende Ports öffnen oder Bastion-Hosts mithilfe von Session Manager pflegen zu müssen	Mithilfe Session Manager der vollständig verwalteten Funktion von AWS Systems Manager können Sie Ihre EC2-Instances über eine interaktive browserbasierte Shell mit einem Klick oder über die verwalten. AWS CLISession Managerbietet eine sichere und überprüfbare Instanzverwaltung, ohne dass eingehende Ports geöffnet, Bastion-Hosts verwaltet oder SSH-Schlüssel verwaltet werden müssen. Session Managerermöglicht es Ihnen außerdem, Unternehm ensrichtlinien einzuhalten, die einen kontrollierten Zugriff auf Instances, strenge Sicherhei tsvorkehrungen und vollständ ig überprüfbare Protokolle mit Instanzzugriffsdetails vorschrei ben, und bietet Endbenutzern gleichzeitig einen einfachen plattformübergreifenden Zugriff mit einem Klick auf Ihre EC2-Instances. Weitere Informati onen finden Sie unter Weitere Informationen zu Session Manager.	11. September 2018

Andere AWS-Services aus einem Systems Manager Automation-Workflow aufrufen	Sie können andere AWS-Services und andere Systems Manager Manager-Funktionen in Ihrem Automatisierungs-Workflow aufrufen, indem Sie drei neue Automatisierungsaktionen (oder Plug-ins) in Ihren Automations-Runbooks verwenden. Weitere Informationen finden Sie unter Verwenden von Aktionsausgaben als Eingaben.	28. August 2018
Verwenden von Systems Manager-spezifischen Bedingungsschlüsseln in IAM-Richtlinien	Das Thema Angeben von Bedingungen in einer Richtlinie wurde um eine Liste der IAM-Bedingungsschlüssel für Systems Manager ergänzt, die Sie in Richtlinien integrieren können. Verwenden Sie diese Schlüssel zum Angeben der Bedingungen, unter denen eine Richtlinie wirksam werden soll. Das Thema enthält auch Links zu Beispielrichtlinien und anderen verwandten Themen.	18. August 2018

Aggregieren von Bestandsdaten mit Gruppen, um zu sehen, welche Instances zur Erfassung eines Bestandstyps konfiguriert sind und welche nicht	Gruppen ermöglichen es Ihnen, schnell eine Anzahl der verwalteten Instances zu sehen, die für das Erfassen eines oder mehrerer Bestandstypen konfiguriert sind bzw. nicht konfiguriert sind. Mit Gruppen geben Sie einen oder mehrere Inventory-Typen sowie einen Filter an, der den `exists`-Operator verwendet. Weitere Informationen finden Sie unter Aggregieren von Bestandsdaten.	16. August 2018
Anzeigen von Verlauf Änderungsnachverfolgung für Inventory und Configuration Compliance	Sie können den Verlauf und die Änderungsnachverfolgung für von Ihren verwalteten Instances erfasstes Inventory anzeigen. Sie können auch den Verlauf und die Änderungsnachverfolgung für Patch Manager-Patch-Einspielungen und State Manager-Zuordnungen, die von Configuration Compliance gemeldet werden, ansehen. Weitere Informationen finden Sie unter Anzeigen von Inventory-Verlauf und Änderungsnachverfolgung.	9. August 2018

Parameter Store integriert sich mit Secrets Manager	Parameter Storeist jetzt integriert, AWS Secrets Manager sodass Sie Secrets Manager abrufen können, wenn Sie andere verwenden AWS-Services , die bereits Verweise auf Parameter Store Parameter unterstüt zen. Zu diesen Services gehören Amazon EC2, Amazon Elastic Container Service,, AWS Lambda, AWS CloudFormation AWS CodeBuild AWS CodeDeplo y, und andere Systems Manager Manager-Funktionen . Indem Sie Parameter Store zum Verweisen auf Secrets Manager-Geheimnisse verwenden, erstellen Sie einen konsistenten und sicheren Prozess zum Aufrufen und Verwenden von Geheimnis sen und zum Referenzieren von Daten in Ihrem Code und den Konfigurationsskri pts. Weitere Informationen finden Sie unter Referenzieren von AWS Secrets Manager Geheimnissen anhand von Parameter Store Parametern.	26. Juli 2018

Anfügen von Bezeichnungen zu Parameter Store-Parametern	Eine Parameter-Bezeichnung ist ein benutzerdefinierter Alias, mit dem Sie verschiedene Versionen eines Parameters verwalten können. Wenn Sie einen Parameter ändern, speichert Systems Manager automatisch eine neue Version und erhöht die Versionsnummer um 1. Dank einer Bezeichnung können Sie sich den Zweck einer Parameterversion merken, wenn mehrere Versionen vorhanden sind. Weitere Informationen finden Sie unter Bezeichnen von Parametern.	26. Juli 2018

Erstellen dynamischer Automatisierungsworkflows	Standardmäßig werden die Schritte (oder Aktionen), die Sie im Abschnitt „mainSteps“ eines Automatisierungs-Runbooks definieren, nacheinander ausgeführt. Wenn eine Aktion abgeschlossen ist, beginnt die nächste, im Abschnitt „mainSteps“ angegebene Aktion. Mit dieser Version können Sie jetzt Automatisierungsworkflows erstellen, die eine bedingte Verzweigung durchführen. Dies bedeutet, dass Sie Automatisierungsworkflows erstellen können, die dynamisch auf geänderte Bedingungen reagieren und zu einem angegebenen Schritt springen. Informationen finden Sie unter Verwendung bedingter Anweisungen in Runbooks.	18. Juli 2018

SSM Agent ist jetzt auf Ubuntu Server 16.04 AMIs unter Verwendung von Snap vorinstalliert	Beginnend mit Instances, die anhand von Ubuntu Server 16.04-AMIs (identifiziert durch `20180627`) erstellt wurden, ist nun der SSM Agent unter Verwendung von Snap-Pakete vorinstalliert. Auf Instances, die aus früheren AMIs erstellt wurden, sollten Sie weiterhin deb-Installationsprogrammpakete verwenden. Weitere Informationen finden Sie unter Über SSM Agent-Installationen auf 64-Bit-Ubuntu Server 16.04-Instances.	7. Juli 2018
Überprüfen der für den SSM Agent erforderlichen minimalen S3-Bucket-Berechtigungen	Das neue Thema Minimale S3-Bucket-Berechtigungen für SSM Agent enthält Informationen zu den Amazon Simple Storage Service (Amazon S3)-Buckets, auf die Ressourcen möglicherweise zugreifen müssen, um Systems Manager-Operationen ausführen zu können. Sie können diese Buckets in einer benutzerdefinierten Richtlinie angeben, wenn Sie den S3-Bucket-Zugriff für ein Instance-Profil oder einen VPC-Endpunkt auf das für die Verwendung von Systems Manager erforderliche Minimum beschränken möchten.	5. Juli 2018

Anzeigen des vollständigen Ausführungsverlaufs für eine bestimmte State Manager-Zuordnungs-ID	Das neue Thema Anzeigen von Zuordnungsverläufen beschreibt, wie alle Ausführungen für eine bestimmte Zuordnungs-ID und anschließend Ausführungsdetails für eine oder mehrere Ressourcen angezeigt werden.	2. Juli 2018
Patch Manager bietet Support für Amazon Linux 2	Sie können nun mit Patch Manager Patches auf Amazon Linux 2-Instances anwenden. Allgemeine Informationen zur Betriebssystemunterstützung von Patch Manager finden Sie unter Voraussetzungen für Patch Manager. Informationen zu den unterstützten Schlüssel-Wert-Paaren für Amazon Linux 2 bei der Definition eines Patch-Filters finden Sie PatchFilterin der AWS Systems Manager API-Referenz.	26. Juni 2018
Befehlsausgabe an Amazon CloudWatch Logs senden	Das neue Thema Konfiguration von Amazon CloudWatch Logs für Run Command beschreibt, wie die Run Command Ausgabe an CloudWatch Logs gesendet wird.	18. Juni 2018

Schnelles Erstellen oder Löschen von Resource Data Sync für Inventory unter Verwendung von AWS CloudFormation	Sie können AWS CloudFormation es verwenden, um eine Ressourcendatensynchronisierung für Systems Manager Inventory zu erstellen oder zu löschen. Um sie zu verwenden AWS CloudFormation, fügen Sie die AWS::SSM::ResourceDataSync-Ressource zu Ihrer AWS CloudFormation Vorlage hinzu. Weitere Informationen finden Sie unter Arbeiten mit AWS CloudFormation - Vorlagen im AWS CloudFormation -Benutzerhandbuch. Sie können auch manuell eine Ressource Data Sync für Inventory erstellen. Einzelheiten dazu finden Sie unter Konfigurieren von Resource Data Sync für Inventory.	11. Juni 2018

AWS Systems Manager Aktualisierungsbenachrichtigungen für das Benutzerhandbuch sind jetzt über RSS verfügbar	Die HTML-Version des Systems Manager-Benutzerhandbuchs unterstützt jetzt einen RSS-Feed für Aktualisierungen, die auf der Seite Aktualisierungsverlauf der Systems Manager-Dokumentation dokumentiert sind. Der RSS-Feed umfasst Aktualisierungen ab Juni 2018 und später. Zuvor angekündigte Aktualisierungen stehen nach wie vor auf der Seite Aktualisierungsverlauf der Systems Manager-Dokumentation zur Verfügung. Verwenden Sie die RSS-Schaltfläche in der oberen Menüanzeige, um den Feed zu abonnieren.	6. Juni 2018
Angabe eines Beendigungscodes in Skripts, um verwaltete Instances neu zu starten	Das neue Thema Neustarten von verwalteten Instances von Skripts beschreibt, wie Sie Systems Manager anweisen, verwaltete Instances neu zu starten, indem Sie einen Beendigungscode in Skripts angeben, die Sie mit Run Command ausführen.	3. Juni 2018

Erstellen Sie ein Ereignis in Amazon, EventBridge wenn benutzerdefiniertes Inventar gelöscht wird	Das neue Thema Aktionen zum Löschen von Lagerbest and anzeigen in EventBridge beschreibt, wie Sie Amazon so konfigurieren, EventBrid ge dass jedes Mal, wenn ein Benutzer benutzerdefinierte s Inventar löscht, ein Ereignis erstellt wird.	1. Juni 2018

Updates vor Juni 2018

In der folgenden Tabelle sind wichtige Änderungen in jeder Version des AWS Systems Manager - Benutzerhandbuchs vor Juni 2018 beschrieben.

Änderung	Beschreibung	Datum der Veröffent lichung
Inventarisieren Sie alle verwalteten Instanzen in Ihrem AWS-Konto	Sie können alle verwalteten Instanzen in Ihrem inventari sieren, AWS-Konto indem Sie eine globale Inventarz uordnung erstellen. Weitere Informationen finden Sie unter Inventarisieren Sie alle verwalteten Knoten in Ihrem AWS-Konto. **Note** Globale Bestandszuordnungen sind in SSM Agent-Version 2.0.790.0 oder höher verfügbar. Weitere Informationen zur Aktualisierung von SSM Agent auf Ihren Instances finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.	3. Mai 2018
SSM Agent ist standardmäßig auf	SSM Agent ist standardmäßig auf Ubuntu Server 18.04 LTS 64-Bit- und 32-Bit-AMIs installiert.	2. Mai 2018

Änderung	Beschreibung	Datum der Veröffentlichung
Ubuntu Server 18 installiert.		
Neues Thema	In dem neuen Thema Ausführen von Befehlen mit einer bestimmten Dokumentversion wird beschrieben, wie mit dem document-version-Parameter angegeben wird, welche Version eines SSM-Dokuments bei der Befehlsausführung verwendet werden soll.	1. Mai 2018
Neues Thema	Im neuen Thema Löschen eines benutzerdefinierten Bestands wird beschrieben, wie Sie benutzerdefinierte Bestandsdaten mithilfe der AWS CLI aus Amazon S3 löschen. Außerdem wird beschrieben, wie Sie mit `SchemaDeleteOption` benutzerdefinierten Bestand verwalten, indem Sie einen benutzerdefinierten Bestandstyp deaktivieren oder löschen. Diese neue Funktion verwendet den DeleteInventoryAPI-Vorgang.	19. April 2018
Amazon-SNS-Benachrichtigungen für SSM Agent	Abonnieren Sie ein Amazon SNS-Thema, um Benachrichtigungen zu erhalten, wenn eine neue Version von SSM Agent verfügbar ist. Weitere Informationen finden Sie unter Abonnieren von SSM Agent-Benachrichtigungen.	9. April 2018
CentOS Patch-Support	Systems Manager unterstützt jetzt das Patchen von CentOS-Instances. Weitere Informationen zu unterstützten CentOS-Versionen finden Sie unter Patch Manager-Voraussetzungen. Weitere Information zur Funktionsweise von Patch-Vorgängen finden Sie unter So funktionieren Patch Manager-Operationen.	29. März 2018

Änderung	Beschreibung	Datum der Veröffentlichung
Neuer -Abschnitt	Um eine einzige Quelle für Referenzinformationen im AWS Systems Manager -Benutzerhandbuch bereitzustellen, wurde der neue Abschnitt AWS Systems Manager Referenz eingeführt. Zusätzliche Inhalte werden diesem Abschnitt hinzugefügt, sobald diese verfügbar werden.	15. März 2018
Neues Thema	Das neue Thema Paketnamen-Formate für Listen genehmigter und abgelehnter Patches erläutert die Paketnamenformate, die Sie in die Listen der genehmigten und abgelehnten Patches für eine benutzerdefinierte Patch-Baseline eingeben können. Beispiel-Formate werden für jeden Betriebssystem-Typ bereitgestellt, der Patch Manager unterstützt.	9. März 2018
Neues Thema	Systems Manager ist jetzt in Chef integriertChef InSpec. InSpec ist ein Open-Source-Runtime-Framework, mit dem Sie menschenlesbare Profile auf GitHub Amazon S3 erstellen können. Anschließend können Sie Systems Manager verwenden, um Compliance-Scans auszuführen und konforme und nicht konforme Instances anzuzeigen. Weitere Informationen finden Sie unter Verwenden von Chef InSpec Profilen mit Systems Manager Compliance.	7. März 2018
Neues Thema	In dem neuen Thema Verwenden von serviceverknüpften Rollen für Systems Manager wird beschrieben, wie Sie eine AWS Identity and Access Management (IAM) -Servicev erknüpfte Rolle mit Systems Manager verwenden. Derzeit sind serviceverknüpfte Rollen nur erforderlich, wenn Sie Systems Manager Inventory zum Sammeln von Metadaten über Tags und Ressourcengruppen verwenden.	27. Februar 2018

Änderung	Beschreibung	Datum der Veröffentlichung
Neue und aktualisierte Themen	Sie können nun Patch Manager verwenden, um Patches zu installieren, die sich in einem anderen Quell-Repository befinden als die Standardversion, die in der Instance konfiguriert ist. Dies ist nützlich für das Einspielen von Patches in Instances mit Updates, die sich nicht auf Sicherheit beziehen, mit dem Content von Personal Package Archives (PPA) für Ubuntu Server, mit Updates für interne Unternehmensanwendungen und so weiter. Sie geben beim Erstellen einer benutzerdefinierten Patch-Baseline alternative Patch-Quell-Repositorys an. Weitere Informationen finden Sie unter den folgenden Themen: • So geben Sie ein alternatives Patch-Quell-Repository an (Linux) • Arbeiten mit benutzerdefinierten Patch-Baselines • Erstellen einer Patch-Baseline mit benutzerdefinierten Repositorys für verschiedene Betriebssystemversionen Darüber hinaus können Sie jetzt Patch Manager verwenden, um SUSE Linux Enterprise Server-Instances zu pathen. Patch Manager unterstützt das Patchen von SLES 12.*-Versionen (nur 64-Bit). Weitere Informationen finden Sie in den SLES-spezifischen Informationen in den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf SUSE Linux Enterprise Server	6. Februar 2018

Änderung	Beschreibung	Datum der Veröffentlichung
Neues Thema	In dem neuen Thema Über SSM-Dokumente für das Patchen von verwalteten Knoten werden die sieben verfügbaren SSM-Dokumente, die Ihnen dabei helfen, Ihre verwalteten Instances mit den neuesten sicherheitsrelevanten Updates zu patchen, beschrieben.	10. Januar 2018
Wichtige Updates zur Linux-Unterstützung	Verschiedene Themen wurden mit den folgenden Informationen aktualisiert: • SSM Agentist standardmäßig auf Amazon Linux 1 Base AMIs von 2017.09 und höher installiert. • Auf anderen Versionen von Linux müssen Sie den SSM Agent manuell installieren, auch auf Nicht-Basis-Images wie Amazon ECS-optimierten AMIs.	9. Januar 2018
Neues Thema	Das neue Thema Informationen über das `AWS-RunPatchBaseline` SSM-Dokument enthält Einzelheiten dazu, wie dieses SSM-Dokument auf Windows- und Linux-Systemen funktioniert. Außerdem erhalten Sie Informationen zu den zwei verfügbaren Parametern im Dokument `AWS-RunPatchBaseline` , `Operation` und `Snapshot ID`.	5. Januar 2018
Neue Themen	Der neue Abschnitt So funktionieren Patch Manager-Operationen enthält technische Details, die erklären, wie Patch Manager bestimmt, welche Sicherheitspatches installiert werden und wie er sie auf dem jeweiligen unterstützten Betriebssystem installiert. Er bietet außerdem Informationen darüber, wie Patch-Baseline-Regeln auf verschiedenen Verteilungen des Linux-Betriebssystems funktionieren.	2. Januar 2018

Änderung	Beschreibung	Datum der Veröffentlichung
Systems Manager-Automatisierungsaktionsreferenz umbenannt und verschoben	Auf Grundlage des Kundenfeedbacks wird die Automatisierungsaktionsreferenz jetzt Systems Manager Automation–Runbook-Referenz genannt. Außerdem haben wir die Referenz in den Knoten „Freigegebene Ressourcen > Dokumente“ verschoben, sodass sie sich näher an Referenz für Befehlsdokument-Plug-ins befinden. Weitere Informationen finden Sie unter Systems Manager Automation Aktionen-Referenz.	20. Dezember 2017
Neue Kapitel und Inhalte zur Überwachung	Ein neues Kapitel,Überwachung AWS Systems Manager, enthält Anweisungen zum Senden von Metriken und Protokolldaten an Amazon CloudWatch Logs. Ein neues ThemaSenden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent), enthält Anweisungen für die Migration von Aufgaben zur Instanzüberwachung (nur für Windows Server 64-Bit-Instances) vom Agenten SSM Agent zum CloudWatch Agenten.	14. Dezember 2017
Neues Kapitel	Ein neues Kapitel enthält umfassende Informationen zur Verwendung von AWS Identity and Access Management (IAM) und AWS Systems Manager zur Sicherung des Zugriffs auf Ihre Ressourcen mithilfe von Anmeldeinformationen. Identity and Access Management für AWS Systems Manager Diese Anmeldeinformationen bieten die erforderlichen Berechtigungen für den Zugriff auf AWS Ressourcen, z. B. für den Zugriff auf Daten, die in S3-Buckets gespeichert sind, und für das Senden von Befehlen an und das Lesen der Tags auf EC2-Instances.	11. Dezember 2017
Änderungen an der linken Navigationsleiste	Wir haben die Überschriften im linken Navigationsbereich dieses Benutzerhandbuchs geändert und an die Überschriften in der neuen AWS Systems Manager -Konsole angepasst.	8. Dezember 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Mehrere Änderungen für re:Invent 2017	• Offizieller Start von AWS Systems Manager: AWS Systems Manager (ehemals Amazon EC2 Systems Manager) ist eine einheitliche Oberfläche, mit der Sie Betriebsdaten zentralisieren und Aufgaben ressourcenübergreifend automatisieren AWS können. Sie können hier auf die neue AWS Systems Manager Konsole zugreifen. Weitere Informationen finden Sie unter Was ist AWS Systems Manager? • YAML-Support: Sie können SSM-Dokumente im YAML-Format erstellen. Weitere Informationen finden Sie unter AWS Systems Manager-Documents.	29. November 2017
Verwendung von Run Command zur Aufnahme VSS-fähiger Snapshots von EBS-Volumes	Sie können mit Run Command anwendungskonsistente Snapshots aller Amazon Elastic Block Store (Amazon EBS)-Volumes erstellen, die Ihren Amazon-EC2-Windows-Instances angefügt sind. Der Snapshot-Vorgang erstellt mit dem Windows Volume Shadow Copy Service (VSS) Backups VSS-fähiger Anwendungen auf Image-Ebene. Dazu gehören auch Daten von schwebenden Transaktionen zwischen diesen Anwendungen und dem Datenträger. Des Weiteren müssen Sie Ihre Instances herunterfahren oder trennen, wenn Sie eine Sicherung aller angefügten Volumes durchführen möchten. Weitere Informationen finden Sie unter Verwenden von Microsoft VSS-fähigen Snapshots AWS Systems Manager im Amazon EC2 EC2-Benutzerhandbuch.	20. November 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Verbesserte Systems Manager-Sicherheit durch Verwendung von VPC-Endpunkten	Sie können die Sicherheitslage Ihrer verwalteten Instances (einschließlich verwalteter Instances in Ihrer Hybrid-Umgebung) verbessern, indem Sie Systems Manager so konfigurieren, dass ein Schnittstellen-VPC-Endpunkt verwendet wird. Schnittstellenendpunkte werden von einer Technologie unterstützt PrivateLink, mit der Sie über private IP-Adressen privat auf Amazon EC2- und Systems Manager Manager-APIs zugreifen können. PrivateLink schränkt den gesamten Netzwerkverkehr zwischen Ihren verwalteten Instances, Systems Manager und EC2 auf das Amazon-Netzwerk ein (verwaltete Instances haben keinen Zugriff auf das Internet). Zudem benötigen Sie kein Internet-Gateway, kein NAT-Gerät und kein Virtual Private Gateway. Weitere Informationen finden Sie unter Verbessern der Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.	7. November 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Inventory-Support für Dateien, Services, Windows-Rollen und die Windows-Registry	SSM Inventory unterstützt jetzt das Sammeln folgender Informationen von Ihren verwalteten Instances. • Files (Dateien): Name, Größe, Version, Installationsdatum, Änderung und Zeitpunkt der letzten Zugriffe usw. • Services: Name, Anzeigename, Status, abhängige Services, Servicetyp, Starttyp usw. • Windows-Registry: Registry-Schlüsselpfad, Wertname, Werttyp und Wert. • Windows-Rollen: Name, Anzeigename, Pfad, Featuretyp, Installationsstatus usw. Bevor Sie versuchen, Informationen für diese Bestandstypen zu sammeln, aktualisieren Sie SSM Agent auf den Instances, die Sie für den Bestand wünschen. Durch Ausführen der neuesten Version von SSM Agent stellen Sie sicher, dass Sie Metadaten für alle unterstützten Bestandstypen sammeln können. Informationen zur Aktualisierung von SSM Agent mithilfe von State Manager finden Sie unter Anleitung: Automatische Aktualisierung von SSM Agent (CLI). Weitere Informationen zu Inventory finden Sie unter Weitere Informationen über Systems Manager Inventory.	6. November 2017
Aktualisierungen der Automation-Dokumentation	Mehrere Probleme mit der Information über die Einrichtung und Konfiguration des Zugriffs für Systems Manager Automation behoben. Weitere Informationen finden Sie unter Einrichten der Automatisierung.	31. Oktober 2017

Änderung	Beschreibung	Datum der Veröffentlichung
GitHubund Amazon S3 S3-Integration	Remote-Skripts ausführen: Systems Manager unterstützt jetzt das Herunterladen und Ausführen von Skripts aus einem privaten oder öffentlichen GitHub Repository sowie von Amazon S3. Mit dem `AWS-RunRemoteScript` vordefinierten SSM-Dokument oder dem `aws:downloadContent` Plugin in einem benutzerdefinierten SSM-Dokument können Sie Ansible Playbooks und Skripte in Python, Ruby oder, um nur einige zu nennen PowerShell, ausführen. Diese Änderungen verbessern weiterhin Infrastruktur in Form von Code bei der Verwendung von Systems Manager, um die Konfiguration und Bereitstellung von EC2-Instances und verwalteten On-Premises-Instances in Ihrer Hybrid-Umgebung zu automatisieren. Weitere Informationen finden Sie unter Ausführen von Skripts von GitHub und Ausführen von Skripts von Amazon S3. Zusammengesetzte SSM-Dokumente erstellen: Systems Manager unterstützt jetzt die Ausführung von einem oder mehreren sekundären SSM-Dokumenten über ein primäres SSM-Dokument. Diese primären Dokumente, die andere Dokumente ausführen, werden als zusammengesetzte Dokumente bezeichnet. Mit Composite Documents können Sie einen Standardsatz sekundärer SSM-Dokumente AWS-Konten für allgemeine Aufgaben wie das Booten von Antivirensoftware oder den Beitritt zu Domänen erstellen und gemeinsam nutzen. Sie können zusammengesetzte und sekundäre Dokumente ausführenGitHub, die in Systems Manager oder Amazon S3 gespeichert sind. Nach dem Erstellen eines zusammengesetzten Dokuments können Sie es mithilfe des vordefinierten `AWS-RunDocument` -SSM-Dokuments ausführen. Weitere Informationen finden Sie unter Erstellen von zusammeng	26. Oktober 2017

Änderung	Beschreibung	Datum der Veröffentlichung
	esetzten Dokumenten und Ausführen von -Dokumenten von Remote-Standorten. SSM-Dokumenten-Plugin-Referenz: Zum einfacheren Zugriff haben wir die SSM-Plugin-Referenz für SSM-Dokumente aus der Systems Manager-API-Referenz in das Benutzerhandbuch verschoben. Weitere Informationen finden Sie unter Referenz für Befehlsdokument-Plug-ins.	
Support für Parameterversionen in Parameter Store	Wenn Sie einen Parameter bearbeiten, erhält Parameter Store jetzt automatisch die Versionsnummer 1. Sie können einen Parameternamen und eine bestimmte Versionsnummer in API-Aufrufen und SSM-Dokumenten angeben. Wenn Sie keine Versionsnummer angeben, verwendet das System automatisch die neueste Version. Parameterversionen bieten eine Schutzebene für den Fall, dass ein Parameter versehentlich geändert wird. Sie können die Werte aller Versionen anzeigen und bei Bedarf auf ältere Versionen verweisen. Sie können auch Parameterversionen verwenden, um zu sehen, wie oft ein Parameter im Lauf eines bestimmten Zeitraums geändert wurde. Weitere Informationen finden Sie unter Arbeiten mit Parameterversionen.	24. Oktober 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Unterstützung für Markierungen von Systems Manager-Dokumenten	Sie können jetzt die AddTagsToResourceAPI, die oder die verwenden, AWS Tools for PowerShell um Systems Manager Manager-Dokumente mit Schlüssel-Wert-Paaren zu kennzeichnen. AWS CLI Das Markieren hilft bei der schnellen Identifizierung bestimmter Ressourcen basierend auf den ihnen zugewiesenen Tags. Dies ist ein Zusatz zur vorhandenen Markierungsunterstützung für verwaltete Instances, Wartungsfenster, Parameter Store-Parameter und Patch-Baselines. Weitere Informationen finden Sie unter Markierungen von Systems Manager-Dokumenten.	3. Oktober 2017
Verschiedene Dokumentations-Aktualisierungen zur Korrektur von Fehlern oder Aktualisierung von Inhalt basierend auf Feedback	• Verwendung von Systems Manager in Hybrid- und Multi-Cloud-Umgebungen wurde mit Informationen zu Raspbian Linux aktualisiert. • Systems Manager mit EC2-Instances verwendenMit neuen Anforderungen für Windows Server Instanzen aktualisiert. SSM Agenterfordert Windows PowerShell 3.0 oder höher, um bestimmte SSM-Dokumente auf Windows Server Instanzen auszuführen (z. B. das ältere `AWS-ApplyPatchBaseline` SSM-Dokument). Vergewissern Sie sich, dass Ihre Windows Server-Instances auf Windows Management Framework 3.0 oder höher ausgeführt werden. Das Framework umfasst PowerShell. Weitere Informationen finden Sie unter Windows Management Framework 3.0	2. Oktober 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Beheben von nicht erreichbaren Windows-Instances mit dem EC2Rescue-Automation-Workflow	EC2Rescue kann Ihnen bei der Diagnose und Behebung von Problemen auf Amazon EC2 Windows Server-Instances helfen. Sie können das Tool als Systems Manager Automation-Workflow ausführen, indem Sie das Dokument AWSSupport-executeEC2Rescue verwenden. Das Dokument AWSSupport-executeEC2Rescue wurde entwickelt, um eine Kombination aus Systems Manager Manager-Aktionen, AWS CloudFormation Aktionen und Lambda-Funktionen auszuführen, die die Schritte automatisieren, die normalerweise für die Verwendung von EC2Rescue erforderlich sind. Weitere Informationen finden Sie unter Ausführen des EC2Rescue-Tools auf nicht erreichbaren Instances.	29. September 2017
SSM Agent standardmäßig auf Amazon Linux installiert	SSM Agent ist standardmäßig auf Amazon Linux AMIs von September 2017 und später, die auf Amazon Linux basieren, installiert. Installieren Sie SSM Agent auf anderen Linux-Versionen manuell. Dies wird in Arbeiten mit SSM Agent auf EC2-Instances für Linux beschrieben.	27. September 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Run Command-Verbesserungen	Run Command umfasst die folgenden Erweiterungen. • Sie können die Ausführung von Befehlen auf bestimmte Instances beschränken, indem Sie eine IAM-Richtlinie erstellen und zuweisen, die eine Bedingung enthält, dass der Benutzer Befehle nur auf Instanzen ausführen kann, die mit bestimmten Amazon-EC2-Tags gekennzeichnet sind. Weitere Informationen finden Sie unter Den Zugriff von Run Command anhand von Tags beschränken. • Sie haben mehrere Möglichkeiten zur Auswahl von Instances mithilfe von Amazon EC2-Tags. Sie können jetzt beim Senden von Befehlen mehrere Tag-Schlüssel und mehrere Tag-Werte angeben. Weitere Informationen finden Sie unter Ausführen von Befehlen in großem Maßstab.	12. September 2017
Auf Raspbian unterstützter Systems Manager	Systems Manager kann jetzt auf Raspbian Jessie- und Raspbian Stretch-Geräten, einschließlich Raspberry Pi (32-Bit), ausgeführt werden.	7. September 2017
Automatisches Senden von SSM Agent Protokollen an Amazon CloudWatch Logs	Sie können jetzt eine einfache Konfigurationsänderung an Ihren Instances vornehmen, an die Protokolldateien SSM Agent gesendet werden sollen CloudWatch. Weitere Informationen finden Sie unter Senden von SSM Agent-Protokollen an CloudWatch Logs.	7. September 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Resource Data Sync verschlüsseln	Mit Systems Manager Resource Data Sync können Sie die auf Hunderten von verwalteten Instances erfassten Bestandsdaten in einem zentralen S3-Bucket zusammenfassen. Sie können Resource Data Sync jetzt mit einem AWS Key Management Service -Schlüssel verschlüsseln. Weitere Informationen finden Sie unter Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten.	1. September 2017
Neue Anleitungen zu State Manager	Der State Manager-Dokumentation wurden zwei neue Anleitungen hinzugefügt: Anleitung: Automatische Aktualisierung von SSM Agent (CLI) Anleitung: Automatische Aktualisierung von PV-Treibern auf EC2-Instances für Windows Server (Konsole)	31. August 2017
Configuration Compliance für Systems Manager	Mit Configuration Compliance können Sie Ihre Flotte verwalteter Instances auf Patch-Compliance und Konfigurationsinkonsistenzen prüfen. Sie können Daten aus mehreren Bereichen sammeln AWS-Konten und AWS-Regionen aggregieren und dann nach bestimmten Ressourcen suchen, die nicht den Vorschriften entsprechen. Standardmäßig werden von Configuration Compliance die Compliance-Daten zu Patch Manager-Patches und State Manager-Zuordnungen angezeigt. Sie können auch den Service anpassen und Ihre eigenen Compliance-Typen auf Grundlage Ihrer IT- oder Business-Anforderungen erstellen. Weitere Informationen finden Sie unter AWS Systems Manager-Compliance.	28. August 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Neue Automatisierungsaktion: `aws:executeAutomation`	Führt einen sekundären Automatisierungsworkflow durch Aufrufen eines sekundären Automatisierungs-Runbooks aus. Mit dieser Aktion können Sie die Automatisierungs-Runbooks für die gängigsten Workflows erstellen und während der Ausführung der Automatisierung auf diese Dokumente verweisen. Mit dieser Aktion können Sie Ihre Automatisierungs-Runbooks vereinfachen, indem Sie die Notwendigkeit für wiederholte Schritte bei ähnlichen Runboks entfernen. Weitere Informationen finden Sie unter `aws:executeAutomation` - Führen Sie eine weitere Automatisierung durch.	22. August 2017
Automatisierung als Ziel eines CloudWatch Ereignisses	Sie können einen Automatisierungs-Workflow starten, indem Sie ein Automation-Runbook als Ziel eines CloudWatch Amazon-Ereignisses angeben. Sie können Workflows nach einem Zeitplan oder bei Eintreten eines bestimmten AWS Systemereignisses starten. Weitere Informationen finden Sie unter Ausführen von Automatisierungen basierend auf Ereignissen.	21. August 2017
State Manager-Zuordnungs-Versioning und allgemeine Aktualisierungen	Sie können jetzt verschiedene State Manager-Zuordnungsversionen erstellen. Es gilt ein Kontingent von 1 000 Versionen pro Zuordnung. Sie können auch Namen für Ihre Zuordnungen angeben. Die State Manager-Dokumentation wurde aktualisiert, um veraltete Informationen und Inkonsistenzen zu behandeln. Weitere Informationen finden Sie unter AWS Systems Manager State Manager.	21. August 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Änderungen an Maintenance Windows	Maintenance Windows umfasst die folgenden Änderungen oder Verbesserungen: • Vorher konnte Maintenance Windows Aufgaben nur mithilfe von Run Command durchführen. Sie können jetzt Aufgaben mithilfe von Systems Manager Automation AWS Lambda, und ausführen AWS Step Functions. • Sie können die Ziele eines Wartungsfensters bearbeiten sowie einen Zielnamen, eine Beschreibung und einen Eigentümer angeben. • Sie können Aufgaben in einem Wartungsfenster bearbeiten. Dies schließt die Angabe eines neuen SSM-Dokuments für Run Command- und Automation-Aufgaben ein. • Alle Run Command Parameter werden jetzt unterstützt DocumentHash, einschließlich DocumentHashType, TimeoutSeconds, Kommentar und NotificationConfig. • Sie können jetzt eine `safe`-Kennzeichnung verwenden, wenn Sie versuchen, ein Ziel abzumelden. Wenn diese Option aktiviert ist, wird vom System ein Fehler zurückgegeben, falls eine beliebige Aufgabe auf das Ziel verweist. Weitere Informationen finden Sie unter AWS Systems Manager Maintenance Windows.	16. August 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Neue Automatisierungsaktion: `aws:approve`	Diese neue Aktion für Automation-Runbooks hält eine Automation-Ausführung zeitweise an, bis die Aktion von designierten Prinzipalen genehmigt oder abgelehnt wird. Nach Erreichen der erforderlichen Anzahl an Genehmigungen wird die Ausführung der Automatisierung fortgesetzt. Weitere Informationen finden Sie unter Systems Manager Automation Aktionen-Referenz.	10. August 2017
Automatisierung übernimmt Rolle nicht mehr erforderlich	Bisher mussten Sie für die Automatisierung eine Servicerolle (oder assume-Rolle) festlegen, damit der Service in Ihrem Auftrag Aktionen ausführen konnte. Diese Rolle ist für die Automatisierung nicht mehr erforderlich, da der Service jetzt den Kontext des Benutzers verwendet, der die Ausführung aufgerufen hat. In den folgenden Situationen müssen Sie jedoch nach wie vor eine Servicerolle zur Automatisierung angeben: • Wenn Sie die Zugriffsberechtigungen eines Benutzers für eine Ressource einschränken, aber dem Benutzer die Ausführung eines Automation-Workflows gestatten möchten, der höhere Berechtigungen erfordert. In diesem Szenario können Sie eine Servicerolle mit höheren Berechtigungen erstellen und dem Benutzer das Ausführen des Workflows gestatten. • Vorgänge, deren Ausführung erwartungsgemäß länger als 12 Stunden dauern, erfordern eine Servicerolle. Weitere Informationen finden Sie unter Einrichten der Automatisierung.	3. August 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Konfigurations-Compliance	Mit Amazon EC2 Systems Manager Configuration Compliance können Sie Ihre Flotte verwalteter Instances auf Patch-Compliance und Konfigurationsinkonsistenzen prüfen. Sie können Daten aus mehreren Bereichen sammeln AWS-Konten und AWS-Regionen aggregieren und dann nach bestimmten Ressourcen suchen, die nicht den Vorschriften entsprechen. Weitere Informationen finden Sie unter AWS Systems Manager-Compliance.	8. August 2017
SSM-Dokumentverbesserungen	SSM-Befehls- und Richtliniendokumente bieten jetzt plattformübergreifende Unterstützung. Das bedeutet, dass ein einzelnes SSM-Dokument Plug-ins für die Betriebssysteme Windows und Linux verarbeiten kann. Mit plattformübergreifender Unterstützung können Sie die Anzahl der verwalteten Dokumente konsolidieren. Plattformübergreifende Unterstützung wird in SSM-Dokumenten mit Schema-Version 2.2 oder höher angeboten. SSM Command-Dokumente mit Schema-Version 2.0 oder höher können jetzt mehrere Plug-ins desselben Typs enthalten. Beispiel: Sie können ein Command-Dokument erstellen, mit dem das Plug-in `aws:runRunShellScript` mehrmals aufgerufen wird. Weitere Informationen zu den Änderungen bei Schema-Version 2.2 finden Sie unter AWS Systems Manager - Dokumente. Weitere Informationen zu SSM-Plug-Ins finden Sie in der Referenz zu Befehlsdokumenten-Plug-Ins.	12. Juli 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Linux-Patching	Patch Manager kann jetzt die folgenden Linux-Distributionen patchen: 64-Bit- und 32-Bit-Systeme • Amazon Linux 2014.03, 2014.09 oder höher • Ubuntu Server 16.04 LTS, 14.04 LTS oder 12.04 LTS • Red Hat Enterprise Linux (RHEL) 6.5 oder höher Nur 64-Bit-Systeme • Amazon Linux 2015.03, 2015.09 oder höher • Red Hat Enterprise Linux(RHEL) 7.x oder höher Weitere Informationen finden Sie unter AWS Systems Manager Patch Manager. Note • Für das Einspielen von Patches auf Linux-Instances muss der SSM Agent mit Version 2.0.834.0 oder höher auf Ihren Instances ausgeführt werden. Informationen zur Aktualisierung des Agenten finden Sie im Abschnitt Beispiel: Aktualisierung von SSM Agent unter Ausführen von Befehlen über die Konsole. • Das `AWS-ApplyPatchBaseline` SSM-Dokument wird durch das `AWS-RunPatchBaseline` -Dokument ersetzt.	6. Juli 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Ressourcen-Datensynchronisierung	Sie können mit Systems Manager Resource Data Sync Bestandsdaten aus allen Ihren verwalteten Instances in einen einzelnen Amazon S3-Bucket senden. Resource Data Sync aktualisiert die Daten dann automatisch, wenn neue Bestandsdaten erfasst werden. Wenn alle Inventardaten in einem Ziel-S3-Bucket gespeichert sind, können Sie Dienste wie Amazon Athena und Amazon verwenden , QuickSight um die aggregierten Daten abzufragen und zu analysieren.Weitere Informationen finden Sie unter. Konfigurieren von Resource Data Sync für Inventory Ein Beispiel für die Arbeit mit Resource Data Sync finden Sie unter Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten.	29. Juni 2017
Systems Manager-Parameterhierarchien	Das Verwalten Dutzender oder Hunderter von Systems Manager-Parametern als unsortierte Liste ist zeitaufwendig und fehleranfällig. Mit Parameterhierarchien können Sie Systems Manager-Parameter leichter organisieren und verwalten. Bei einer Hierarchie handelt es sich um einen Parameternamen mit einem Pfad, den Sie mit Schrägstrichen definieren. Hier finden Sie ein Beispiel mit drei Hierarchieebenen im Namen. Damit wird Folgendes identifiziert: /Umgebung/Computertyp/Anwendung/Daten `/Dev/DBServer/MySQL/db-string13` Weitere Informationen finden Sie unter Arbeiten mit Parameterhierarchien. Ein Beispiel für die Arbeit mit Parameterhierarchien finden Sie unter Arbeiten mit Parameterhierarchien.	22. Juni 2017

Änderung	Beschreibung	Datum der Veröffentlichung
SSM Agent-Unterstützung für SUSE Linux Enterprise Server	Sie können SSM Agent auf 64-Bit SUSE Linux Enterprise Server (SLES) installieren. Weitere Informationen finden Sie unter Arbeiten mit SSM Agent auf EC2-Instances für Linux.	14. Juni 2017

Dokumentkonventionen

Nachfolgend finden Sie allgemeine typografischen Konventionen für das AWS Systems Manager - Benutzerhandbuch.

Differenzierte Beispiele für lokale Betriebssysteme oder Befehlszeilensprachen

Wir verwenden Registerkarten zur Darstellung verschiedener Befehlsbeispiele, die auf dem lokalen Betriebssystemtyp eines Benutzers basieren. In den Beispielen für Linux und macOS verwenden wir den umgekehrten Schrägstrich \), um lange Befehle in mehrere Zeilen aufzuteilen. In den Windows Server-Beispielen verwenden wir das Caret-Zeichen (^), um Befehle in mehrere Zeilen aufzuteilen.

Beispiel:

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
```

Elemente auf der Benutzeroberfläche

Formatierung: fett ausgezeichneter Text

Beispiel: Wählen Sie File, Properties.

Benutzereingabe (Text, den ein Benutzer eingibt)

Format: Text in einer nicht proportionalen Schriftart

Beispiel: Geben Sie als Namen **`my-new-resource`** ein.

Platzhaltertext für einen erforderlichen Wert

Formatierung: *kursiver* Text

Beispiel:

```
aws ec2 register-image --image-location DOC-EXAMPLE-BUCKET/image.manifest.xml
```

AWS Glossar

Die neueste AWS Terminologie finden Sie im AWS Glossar in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.