Leitfaden

AWS CloudFormation

API-Version 2010-05-15

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

AWS CloudFormation: Leitfaden

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

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS CloudFormation? ... 1
Infrastrukturverwaltung vereinfachen ... 1
Schnelles Replizieren der Infrastruktur ... 1
Einfaches Kontrollieren und Nachverfolgen von Änderungen an Ihrer Infrastruktur ... 2
Erste Schritte mit CloudFormation ... 2
Ähnliche Informationen ... 3
Erste Schritte ... 4
Wie CloudFormation funktioniert ... 4
Die wichtigsten Konzepte ... 4
Wie CloudFormation funktioniert ... 9
Möglichkeiten für den Einstieg CloudFormation ... 11
Melden Sie sich an für ein AWS-Konto ... 12
Melden Sie sich für eine an AWS-Konto ... 12
Erstellen eines Benutzers mit Administratorzugriff ... 12
Erstelle deinen ersten Stack ... 14
Voraussetzungen ... 15
Erstellen Sie einen CloudFormation Stack mit der Konsole ... 15
Überwachen Sie die Stapelerstellung ... 19
Testen Sie den Webserver ... 20
Fehlerbehebung ... 21
Bereinigen ... 22
Nächste Schritte ... 23
Bewährte Methoden ... 24
Verkürzen der Feedbackschleife zur Verbesserung der Liefergeschwindigkeit ... 25
Stacks nach Lebenszyklus und Besitz organisieren ... 25
Verwenden von Stack-übergreifenden Verweisen zum Export von gemeinsam genutzten Ressourcen ... 26
Prüfen der Kontingente für alle Ressourcentypen ... 27
Verwenden von Vorlagen zum Replizieren von Stacks in mehreren Umgebungen ... 27
Verwenden von Modulen, um Ressourcenkonfigurationen wiederzuverwenden ... 28
Verwenden Sie AWS-spezifische Parametertypen ... 28
Verwenden von Parameter-Einschränkungen ... 29
Verwenden Sie Pseudoparameter, um die Portabilität zu fördern. ... 29

Wird `AWS::CloudFormation::Init` zur Bereitstellung von Softwareanwendungen auf EC2 Amazon-Instances verwendet ... 30
Verwenden der neuesten Hilfsskripts ... 30
Validieren von Vorlagen vor ihrer Verwendung ... 31
Prüfen von Vorlagen auf die Einhaltung von Organisationsrichtlinien ... 31
Verwalte alle Stack-Ressourcen über CloudFormation ... 31
Erstellen von Änderungssätzen vor der Aktualisierung von Stacks ... 32
Verwenden von Stack-Richtlinien ... 32
Verwenden von Code-Bewertungen und Revisionskontrollen für die Verwaltung Ihrer Vorlagen ... 33
Aktualisieren Sie Ihre EC2 Amazon-Instances regelmäßig ... 33
Arbeiten mit -Vorlagen ... 34
Wo Vorlagen gespeichert werden ... 35
Vorlagen werden validiert ... 35
Erste Schritte mit Vorlagen ... 2
Planen Sie, die CloudFormation Vorlagenreferenz zu verwenden ... 36
Mustervorlagen ... 37
Vorlagenformat ... 37
Struktur der Vorlage ... 38
Kommentare ... 40
Technische Daten ... 41
Weitere Informationen ... 41
Verwenden von regulären Ausdrücken ... 42
Abschnitte einer Vorlage ... 43
Resources ... 45
Parameters ... 53
Outputs ... 68
Mappings ... 74
Metadata ... 81
Rules ... 87
Conditions ... 107
Transform ... 119
Formatversion ... 120
Description ... 121
Infrastructure Composer ... 122

Warum sollte Infrastructure Composer im CloudFormation Konsolenmodus verwendet werden? ... 122
Wie unterscheidet sich dieser Modus von der Infrastructure Composer-Konsole? ... 122
IaC-Generator ... 123
Überlegungen ... 124
Berechtigungen ... 125
Häufig verwendete Befehle ... 126
Migrieren Sie eine Vorlage auf die AWS CDK ... 126
Starten Sie einen Ressourcenscan ... 127
Sehen Sie sich die Scan-Zusammenfassung an ... 131
Erstellen Sie eine Vorlage aus gescannten Ressourcen ... 132
Erstellen Sie einen Stapel aus gescannten Ressourcen ... 137
Löst Eigenschaften auf, die nur über Schreibzugriff verfügen ... 138
Ruft Werte ab, die in anderen Diensten gespeichert sind ... 144
Allgemeine Überlegungen ... 144
Klartextwert von Systems Manager abrufen ... 145
Sichere Zeichenfolge von Systems Manager abrufen ... 148
Holen Sie sich den Wert von Secrets Manager ... 152
Werte abrufen AWS ... 156
Syntax ... 156
Verfügbare Pseudoparameter ... 157
Beispiele ... 159
Holen Sie sich Stack-Ausgaben ... 163
Exportieren von Stack-Ausgabewerten im Vergleich zur Verwendung geschachtelter Stacks ... 164
Überlegungen ... 164
Auflistung von exportierten Ausgabewerten ... 165
Auflistung von Stacks, die einen exportierten Ausgabewert importieren ... 166
Geben Sie vorhandene Ressourcen zur Laufzeit an ... 167
Übersicht ... 167
Beispiel ... 168
Überlegungen ... 172
Unterstützte AWS-spezifische Parametertypen ... 173
Unterstützte Systems Manager Manager-Parametertypen ... 174
Nicht unterstützte Systems Manager Manager-Parametertypen ... 175
Anleitungen ... 176

Verweisen Sie auf Ressourcenausgaben in einem anderen CloudFormation Stack 177
Verknüpfen mit einer VPC in einem anderen Konto 182
Erstellen Sie eine skalierte Anwendung mit Load Balancing 193
Stellen Sie Anwendungen auf Amazon bereit EC2 207
Aktualisieren eines Stacks 217
Führen Sie ECS-Bereitstellungen blue/green durch 252
Vorlagenausschnitte 277
Allgemeines 278
Auto-Scaling 289
AWS Abrechnungskonsole 322
AWS CloudFormation 326
CloudFront 333
CloudWatch 345
CloudWatch Logs 351
DynamoDB 390
Amazon EC2 394
Amazon ECS 433
Amazon EFS 456
Elastic Beanstalk 482
Elastic Load Balancing 486
IAM 492
AWS Lambda 510
Amazon Redshift 519
Amazon RDS 529
Route 53 538
Amazon S3 546
Amazon SNS 555
Amazon SQS 555
Amazon Timestream 556
Windowsbasierte Stacks 585
WindowsBootstrapping-Stapel 587
Verwenden Sie von CloudFormation -bereitgestellte Ressourcentypen 594
Benutzerdefinierte Ressourcen 596
Vorlagenmakros 652
Verschachtelte Stacks 670
Wartebedingungen 676

Erstellen Sie wiederverwendbare Ressourcenkonfigurationen mit Modulen 683
Überlegungen zur Verwendung von Modulen 685
Verstehen Sie die Versionierung von Modulen 685
Verwenden Sie Module in einer Vorlage 687
Verwenden Sie Parameter, um Modulwerte anzugeben 688
Ressourcen für Referenzmodule 691
Stacks erstellen und verwalten 693
Schnittstellen für die Verwaltung Ihrer Stacks 695
Erstellen eines Stacks 696
Erstellen eines Stacks 696
Konfigurieren Sie die Stack-Optionen 699
Sehen Sie sich eine Vorschau der Konfiguration Ihres Stacks an 702
Stack-Informationen anzeigen 703
Aktualisiere deine Stack-Vorlage 706
Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen 708
Stapel mithilfe von Änderungssätzen aktualisieren 710
Erstellen Sie einen Änderungssatz 712
Einen Änderungssatz anzeigen 719
Führt einen Änderungssatz aus 728
Löscht einen Änderungssatz 730
Exemplarische Änderungssätze 732
Änderungssätze für verschachtelte Stacks 745
Stacks direkt aktualisieren 752
Ein Stack-Update abbrechen 756
Um ein Stack-Update abzubrechen (Konsole) 756
Um ein Stack-Update abzubrechen (AWS CLI) 756
Löschen eines Stacks 757
Zugehörige Ressourcen 759
Gelöschte Stacks anzeigen 759
Überwachen Sie den Stapelfortschritt 759
Stack-Ereignisse anzeigen 760
Zeitdiagramm für die Stack-Bereitstellung anzeigen 769
Verstehen Sie die Ereignisse bei der Stack 770
Überwachen Sie Stack-Updates 773
Rollback einer Aktualisierung fortsetzen 775
Ermitteln Sie die Ursache eines Stack-Fehlers 778

Stack-Fehleroptionen ... 780
Setzen Sie Ihren Stack bei einem Alarmverstoß zurück ... 790
Fügen Sie Rollback-Trigger während der Stack-Erstellung oder Aktualisierung hinzu ... 792
Fügen Sie einem Änderungssatz Rollback-Trigger hinzu ... 792
Rollback-Trigger für einen Stack anzeigen ... 793
Rollback-Trigger für einen Änderungssatz anzeigen ... 793
Erkennen Sie nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen ... 794
Was ist Abweichung? ... 794
Statuscodes für die Abweichungserkennung ... 796
Überlegungen zur Erkennung von Abweichungen ... 798
Drift auf einem ganzen CloudFormation Stack erkennen ... 801
Erkennen von Abweichungen bei einzelnen Stack-Ressourcen ... 806
Beheben von Abweichungen mit einer Importoperation ... 809
AWS Ressourcen importieren ... 819
AWS Ressourcen manuell importieren ... 820
Automatischer Import von AWS Ressourcen ... 857
Wiederherstellen eines Importvorgangs ... 858
Refactoring von Stapeln ... 864
So funktioniert das Stack-Refactoring ... 864
Einschränkungen beim Stack-Refactoring ... 865
Refaktorierung eines Stacks mit dem AWS Command Line Interface ... 868
Unterstützung für Ressourcentypen ... 870
Verwenden Sie Schnellerstellungslinks, um Stapel zu erstellen CloudFormation ... 1076
URL-Format ... 1076
Beispiel ... 1078
Erstellen eines Stacks mithilfe eines Schnellerstellungslinks ... 1078
Beispielbefehle für AWS CLI und PowerShell ... 1080
CancelUpdateStack ... 1081
ContinueUpdateRollback ... 1082
CreateStack ... 1083
Bereitstellen ... 1087
DeleteStack ... 1088
DescribeStackEvents ... 1089
DescribeStackResource ... 1091
DescribeStackResources ... 1092
DescribeStacks ... 1096

GetTemplate 1100
ListStackResources 1101
ListStacks 1103
UpdateStack 1105
ValidateTemplate 1109
Laden Sie lokale Artefakte in einen S3-Bucket hoch 1112
Verwaltung von Stacks mit StackSets 1115
StackSets Konzepte 1116
Administrator- und Ziel-Konten 1116
AWS CloudFormation StackSets 1116
Berechtigungsmodelle für StackSets 1117
Stack-Instances 1118
StackSet Operationen 1118
StackSet Operationsoptionen 1119
Tags 1121
StackSets Statuscodes 1122
Stack-Instance-Statuscodes 1123
Voraussetzungen 1125
Aktivieren Sie AWS-Regionen , die standardmäßig deaktiviert sind 1126
Selbstverwaltete Berechtigungen erteilen 1126
Aktivieren Sie den vertrauenswürdigen Zugriff 1143
Beginnen Sie mit der Verwendung einer Beispielvorlage 1148
Voraussetzungen 1149
Erstellen Sie eine StackSet mit einer Beispielvorlage aus der Konsole 1150
Überwachen Sie StackSet die Erstellung 1153
StackSet Ergebnisse anzeigen 1153
Aktualisiere deine StackSet 1154
Füge Stapel zu deinem hinzu StackSet 1155
Bereinigen 1155
Nächste Schritte 1157
Erstellen StackSets (selbstverwaltete Berechtigungen) 1158
Erstellen Sie eine StackSet mit selbstverwalteten Berechtigungen (Konsole) 1158
Erstellen Sie eine StackSet mit selbstverwalteten Berechtigungen ()AWS CLI 1161
Erstellen StackSets (vom Service verwaltete Berechtigungen) 1162
Überlegungen 1162
Erstellen Sie eine StackSet mit vom Service verwalteten Berechtigungen (Konsole) 1163

Erstellen Sie eine StackSet mit vom Service verwalteten Berechtigungen ()AWS CLI 1167
Automatische Bereitstellungen aktivieren/deaktivieren 1171
Wie funktionieren automatische Bereitstellungen 1171
Überlegungen 1171
Automatische Bereitstellungen aktivieren oder deaktivieren (Konsole) 1172
Automatische Bereitstellungen aktivieren oder deaktivieren ()AWS CLI 1173
Aktualisiere StackSets 1173
Aktualisieren Sie Ihre StackSet (Konsole) 1174
Aktualisieren Sie Ihre StackSet (AWS CLI) 1176
Stapel hinzufügen zu StackSets 1177
Stapel zu einer StackSet (Konsole) hinzufügen 1178
Fügen Sie Stacks zu einem StackSet () hinzu AWS CLI 1180
Überschreiben Sie Parameter auf Stacks 1181
Parameter auf Stacks überschreiben (Konsole) 1182
Parameter auf Stacks überschreiben ()AWS CLI 1184
Stapel löschen von StackSets 1185
Löschen Sie Stacks von Ihrer StackSet (Konsole) 1185
Lösche Stacks von deinem StackSet ()AWS CLI 1187
Löschen StackSets 1189
Löschen Sie eine StackSet (Konsole) 1189
Löschen Sie ein StackSet (AWS CLI) 1190
Löschen von Servicerollen (optional) 1190
Zielkonto-Gates 1191
Voraussetzungen 1192
CloudFormation Vorlagen für die Erstellung von Lambda-Funktionen 1192
Wählen Sie den Parallelitätsmodus 1193
So funktionieren die einzelnen Parallelitätsmodi 1194
Wählen zwischen strikter und loser Fehlertoleranz auf Grundlage der Bereitstellungsgeschwindigkeit 1196
Wählen Sie Ihren Parallelitätsmodus (Konsole) 1198
Wählen Sie Ihren Parallelitätsmodus ()AWS CLI 1198
Drift wird erkannt an StackSets 1199
Wie CloudFormation funktioniert die Drifterkennung auf einem StackSet 1200
Erkennen Sie Drift auf einer StackSet (Konsole) 1201
Erkennt Drift auf einem StackSet (AWS CLI) 1203
Stoppen der Drifterkennung auf einem StackSet 1210

Stapel importieren in StackSets ... 1210
Selbstverwalteter Stack-Import ... 1211
Vom Service verwalteter Stack-Import ... 1215
Stack-Importe rückgängig machen ... 1221
Bewährte Methoden ... 1221
Definieren der Vorlage ... 1222
Erstellen oder Hinzufügen von Stacks zum StackSet ... 1222
Aktualisierung von Stacks in einem StackSet ... 1222
Mustervorlagen ... 1223
Fehlerbehebung ... 1225
Häufige Gründe für Störungen der Stack-Operationen ... 1225
Wiederholen fehlgeschlagenen Operationen zum Erstellen und Aktualisieren von Stacks ... 1226
Löschen von Stack-Instances schlägt fehl ... 1227
Stack-Importvorgang schlägt fehl ... 1227
Anzahl der Stack-Instance-Fehler für StackSets Operationen ... 1227
Stacks mit Git-Quellcode synchronisieren ... 1234
So funktioniert Git Sync ... 1235
So funktioniert Git Sync ... 1235
Kommentare zu Pull-Anforderungen ... 1237
Stack-Bereitstellungsdatei ... 1237
CloudFormation Vorlagendatei ... 1238
Vorlagendefinitions-Repository ... 1238
Voraussetzungen ... 1239
Git-Repository ... 1239
CloudFormation Vorlage ... 1239
Git-Sync-Dienstrolle ... 1239
IAM-Berechtigungen für Konsolenbenutzer ... 1243
Erstellen Sie einen Stack aus dem Repository-Quellcode ... 1246
Erstellen Sie einen Stack aus dem Quellcode des Repositorys ... 1246
Aktualisiere deinen Stack aus deinem Git-Repository ... 1249
Kommentare zu Pull-Requests aktivieren ... 1250
Status-Dashboard ... 1250
Git-Sync-Status ... 1251
Letzte Synchronisierungsereignisse ... 1252
Unterstützte Stack-Status ... 1252
Verwaltung von Erweiterungen mit der CloudFormation Registrierung ... 1257

Verwandte Dokumentation ... 1257
Konzepte ... 1258
Arten von Erweiterungen ... 1258
Öffentliche Erweiterungen ... 1259
Aktivierte Erweiterungen ... 1259
Sehen Sie sich die verfügbaren und aktivierten Erweiterungen an ... 1260
Öffentliche Erweiterungen ... 1261
IAM-Rolle ... 1263
Automatisch neue Versionen von Erweiterungen verwenden ... 1265
Verwenden Sie Aliase, um auf Erweiterungen zu verweisen ... 1266
Häufig verwendete AWS CLI Befehle für die Arbeit mit öffentlichen Erweiterungen ... 1266
Aktiviere eine öffentliche Erweiterung ... 1267
Aktualisieren Sie eine öffentliche Erweiterung ... 1272
Deaktivieren Sie öffentliche Erweiterungen ... 1273
Private Erweiterungen ... 1274
IAM-Berechtigungen ... 1275
Häufig verwendete AWS CLI Befehle für die Arbeit mit privaten Erweiterungen ... 1275
Registrieren Sie eine private Erweiterung ... 1276
Private Erweiterungen abmelden ... 1278
Bearbeiten Sie die Konfigurationsdaten für Erweiterungen ... 1280
Für die Verwendung dynamischer Verweise sind Berechtigungen erforderlich ... 1280
Bearbeiten Sie die Konfigurationsdaten für eine Erweiterung (Konsole) ... 1281
Bearbeiten Sie die Konfigurationsdaten für eine Erweiterung (AWS CLI) ... 1282
Aufzeichnen von Ressourcentypen in AWS Config ... 1283
Verhindern, dass vertrauliche Eigenschaften in einem Konfigurationselement aufgezeichnet werden ... 1284
Fortlaufende Lieferung ... 1285
Walkthrough: Erstellen einer Pipeline für Test- und Produktions-Stacks ... 1285
Voraussetzungen ... 1286
Walkthrough-Übersicht ... 1286
Schritt 1: Artefakt bearbeiten und in einen S3-Bucket hochladen ... 1287
Schritt 2: Pipeline-Stack erstellen ... 1289
Schritt 3: Sehen Sie sich den WordPress Stapel an ... 1295
Schritt 4: Bereinigen von Ressourcen ... 1296
Weitere Informationen finden Sie auch unter ... 1297
Referenz der Konfigurationseigenschaften ... 1298

Konfigurationseigenschaften (Konsole) ... 1298
Konfigurationseigenschaften (JSON-Objekt) ... 1301
Weitere Informationen finden Sie auch unter ... 1306
AWS CloudFormation Artefakte ... 1307
Stack-Vorlagendatei ... 1307
Vorlagenkonfigurationsdatei ... 1307
Weitere Informationen finden Sie auch unter ... 1309
Verwenden von Funktionen zum Überschreiben von Parametern mit CodePipeline Pipelines 1309
`Fn::GetArtifactAtt` ... 1309
`Fn::GetParam` ... 1311
Weitere Informationen finden Sie auch unter ... 1314
Referenzhandbuch für Vorlagen ... 1315
Sicherheit ... 1316
Schützt Stapel davor, gelöscht zu werden ... 1317
Benutzer festlegen, die den Beendigungsschutz von Stacks ändern dürfen ... 1319
Verhindern von Aktualisierungen der Stack-Ressourcen ... 1319
Beispiel-Stack-Richtlinie ... 1320
Definieren einer Stack-Richtlinie ... 1321
Festlegen einer Stack-Richtlinie ... 1326
Aktualisieren geschützter Ressourcen ... 1327
Ändern einer Stack-Richtlinie ... 1330
Weitere Beispiel-Stack-Richtlinien ... 1331
Datenschutz ... 1335
Verschlüsselung im Ruhezustand ... 1336
Verschlüsselung während der Übertragung ... 1336
Richtlinie für den Datenverkehr zwischen Netzwerken ... 1337
Steuern Sie den Zugriff mit IAM ... 1337
Definition identitätsbasierter IAM-Richtlinien für CloudFormation ... 1337
Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen ... 1347
Verwaltung von Anmeldeinformationen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden ... 1348
Temporären Zugriff gewähren (Verbundzugriff) ... 1349
Beispiele für identitätsbasierte Richtlinien ... 1350
AWS CloudFormation Servicerolle ... 1359
Serviceübergreifende Confused-Deputy-Prävention ... 1360
FAS-Anfragen und Bewertung von Berechtigungen ... 1362

Protokollieren von API-Aufrufen ... 1364
CloudFormation Informationen in CloudTrail ... 1365
CloudFormation Logdateieinträge verstehen ... 1366
Sicherheit der Infrastruktur ... 1370
Ausfallsicherheit ... 1371
Compliance-Validierung ... 1371
Konfigurations- und Schwachstellenanalyse ... 1373
Bewährte Methoden für die Gewährleistung der Sicherheit ... 1373
Verwendung von IAM für die Zugriffskontrolle ... 1373
Keine Anmeldeinformationen in Vorlagen einbetten ... 1374
Wird verwendet AWS CloudTrail , um CloudFormation Anrufe zu protokollieren ... 1374
AWS PrivateLink ... 1374
Überlegungen zu CloudFormation VPC-Endpunkten ... 1375
Erstellen eines Schnittstellen-VPC-Endpunkts für CloudFormation ... 1376
Erstellen einer VPC-Endpunktrichtlinie für CloudFormation ... 1376
Überwachung mit EventBridge ... 1378
CloudFormation und Übersicht über Git-Sync-Ereignisse ... 1378
Berechtigungen ... 1381
Ein benutzerdefiniertes Ereignismuster erstellen ... 1381
Detailreferenz zu Ereignissen ... 1382
Änderung des Ressourcenstatus ... 1383
Änderung des Stack-Status ... 1386
Änderung des Status der Drifterkennung ... 1388
StackSet Änderung des Status ... 1390
StackSet Änderung des Status der Stack-Instance ... 1392
StackSetÄnderung des Betriebsstatus ... 1394
Änderung des Repository-Synchronisierungsstatus ... 1396
Änderung des Status der Ressourcensynchronisierung ... 1399
Kontingente ... 1403
Verfügbarkeit von Funktionen ... 1412
StackSets und Makros ... 1412
Fehlerbehebung ... 1413
Anleitung zur Fehlerbehebung ... 1413
Behebung von Fehlern ... 1414
Stack-Fehler löschen ... 1415
Abhängigkeitsfehler ... 1416

AWS Config und AWS Systems Manager Konflikte 1416
Fehler bei der Analyse der Parameter bei der Übergabe einer Liste 1417
IAM Unzureichende Berechtigungen 1417
Ungültiger Wert oder nicht unterstützte Ressourceneigenschaft 1417
Kontingent überschritten 1418
Rollback-Fehler bei verschachtelten Stacks 1418
Keine Aktualisierungen zur Ausführung vorhanden 1419
Ressource konnte während eines Erstellens, Aktualisierens oder Löschens des Stack-Vorgangs nicht stabilisiert werden. 1419
Sicherheitsgruppe ist in VPC nicht vorhanden. 1420
Rollback-Aktualisierung fehlgeschlagen 1420
Die Wartebedingung hat nicht die erforderliche Anzahl von Signalen von einer EC2 Amazon-Instance empfangen 1422
Ressource wurde vom Stack entfernt, aber nicht gelöscht. 1423
Den -Support kontaktieren 1423
Problembehandlung mit Amazon Q 1425
Features 1425
Funktionsweise 1426
Preisgestaltung 1426
Regionale Verfügbarkeit 1426
Einschränkungen 1426
Dokumentverlauf 1427
Archivierte Updates 1457
..... mdlxxxviii

Was ist AWS CloudFormation?

AWS CloudFormation ist ein Service, der Sie bei der Modellierung und Einrichtung Ihrer AWS Ressourcen unterstützt, sodass Sie weniger Zeit mit der Verwaltung dieser Ressourcen verbringen und sich mehr auf Ihre Anwendungen konzentrieren können, die in ausgeführt AWS werden. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen beschreibt (wie EC2 Amazon-Instances oder Amazon RDS-DB-Instances), und CloudFormation kümmert sich um die Bereitstellung und Konfiguration dieser Ressourcen für Sie. Sie müssen AWS Ressourcen nicht einzeln erstellen und konfigurieren und herausfinden, was wovon abhängt; CloudFormation kümmert sich darum. Die folgenden Szenarien zeigen, wie CloudFormation Sie helfen können.

Infrastrukturverwaltung vereinfachen

Für eine skalierbare Webanwendung, die eine Back-End-Datenbank umfasst, können Sie eine Auto-Scaling-Gruppe, einen Elastic Load Balancing Load Balancer und eine Datenbank-Instance von Amazon Relational Database Service verwenden. Sie können jeden einzelnen Service verwenden, um diese Ressourcen bereitzustellen, allerdings müssen Sie die Ressourcen konfigurieren, nachdem sie erstellt worden sind, damit sie zusammenarbeiten können. Alle diese Aufgaben können kompliziert und zeitaufwendig sein, bevor Ihre Anwendung überhaupt startfähig ist.

Stattdessen können Sie eine CloudFormation Vorlage erstellen oder eine vorhandene ändern. Eine Vorlage beschreibt alle Ressourcen und deren Eigenschaften. Wenn Sie diese Vorlage verwenden, um einen CloudFormation Stack zu erstellen, stellt sie die CloudFormation Auto Scaling Scaling-Gruppe, den Load Balancer und die Datenbank für Sie bereit. Nachdem der Stack erfolgreich erstellt wurde, sind Ihre AWS Ressourcen betriebsbereit. Sie können den Stack genauso einfach löschen, womit dann auch alle Ressourcen im Stack gelöscht werden. Durch die Verwendung CloudFormation können Sie auf einfache Weise eine Sammlung von Ressourcen als eine einzige Einheit verwalten.

Schnelles Replizieren der Infrastruktur

Wenn Ihre Anwendung zusätzliche Verfügbarkeit erfordert, sollten sie in mehreren Regionen replizieren, damit die Benutzer Ihre Anwendung in anderen Regionen weiter nutzen können, wenn eine Region ausfällt. Die Herausforderung bei der Replikation einer Anwendung besteht darin, dass auch die Ressourcen repliziert werden müssen. Sie müssen nicht nur alle von der Anwendung benötigten Ressourcen verzeichnen, sondern diese Ressourcen auch in jeder Region bereitstellen und konfigurieren.

Verwenden Sie Ihre CloudFormation Vorlage erneut, um Ihre Ressourcen konsistent und wiederholbar zu erstellen. Um Ihre Vorlage erneut zu verwenden, beschreiben Sie Ihre Ressourcen einmal und stellen Sie die gleichen Ressourcen dann in mehreren Regionen immer wieder bereit.

Einfaches Kontrollieren und Nachverfolgen von Änderungen an Ihrer Infrastruktur

In einigen Fällen haben Sie möglicherweise zugrunde liegende Ressourcen, die Sie inkrementell aktualisieren möchten. Sie können beispielsweise in Ihrer Auto Scaling-Startkonfiguration zu einem Instance-Typ mit einer höheren Leistung wechseln, sodass Sie die maximale Anzahl von Instances in der Auto-Scaling-Gruppe verringern können. Wenn Probleme nach Abschluss der Aktualisierung auftreten, müssen Sie möglicherweise Ihre Infrastruktur durch einen ein Rollback-Vorgang auf die ursprünglichen Einstellungen zurücksetzen. Um dies manuell durchführen zu können, müssen Sie nicht nur wissen, welche Ressourcen geändert wurden, sondern Sie müssen auch deren Originaleinstellungen kennen.

Wenn Sie Ihre Infrastruktur bereitstellen CloudFormation, beschreibt die CloudFormation Vorlage genau, welche Ressourcen bereitgestellt werden und welche Einstellungen sie haben. Da es sich bei den Vorlagen um Textdateien handelt, müssen Sie zum Verfolgen der Infrastrukturänderungen lediglich die Unterschiede in den Vorlagen verfolgen, ähnlich wie Entwickler durch die Versionskontrolle Änderungen am Quellcode verfolgen. Sie können beispielsweise ein Versionskontrollsystem für Ihre Vorlagen verwenden, damit Sie genau wissen, wer wann welche Änderungen daran vorgenommen hat. Sollten Sie zu irgend einem Zeitpunkt Änderungen an Ihrer Infrastruktur rückgängig machen müssen, können Sie eine ältere Version der Vorlage verwenden.

Erste Schritte mit CloudFormation

CloudFormation ist über die CloudFormation Konsole, API, AWS CLIAWS SDKs, und über verschiedene Integrationen verfügbar.

Eine Einführung zu finden Sie CloudFormation unterWie CloudFormation funktioniert.

Informationen zum Einstieg in die Nutzung CloudFormation finden Sie unterErstelle deinen ersten Stack.

Ähnliche Informationen

CloudFormation In diesem Benutzerhandbuch sowie in den folgenden Ressourcen erfahren Sie mehr darüber:

- Produktdetails und FAQs weitere Informationen finden Sie auf der AWS CloudFormation Produktseite.
- Preisinformationen finden Sie unter AWS CloudFormation Preise.

Erste Schritte mit CloudFormation

Sie können mit der Nutzung des beginnen, AWS Management Console indem Sie CloudFormation anhand einer Beispielvorlage einen Stack erstellen, der Ihnen hilft, die Grundlagen der Stack-Erstellung zu erlernen. Eine Vorlage ist eine Textdatei, die alle Ressourcen innerhalb eines Stacks definiert. Ein Stack ist die Bereitstellung einer CloudFormation Vorlage. Aus einer einzigen Vorlage können Sie mehrere Stapel erstellen. Jeder Stapel enthält eine Sammlung von AWS Ressourcen, die als eine einzige Einheit verwaltet werden können.

CloudFormation ist ein kostenloser Service. Die AWS Ressourcen, die Sie in Ihren Stacks aufnehmen, werden Ihnen jedoch zu den jeweils aktuellen Preisen in Rechnung gestellt. Weitere Informationen zur AWS Preisgestaltung finden Sie auf der Detailseite der einzelnen Produkte unter http://aws.amazon.com.

Themen

- Wie CloudFormation funktioniert
- Melden Sie sich an für ein AWS-Konto
- Erstelle deinen ersten Stack

Wie CloudFormation funktioniert

In diesem Thema wird beschrieben, wie es CloudFormation funktioniert, und es werden die wichtigsten Konzepte vorgestellt, die Sie bei der Verwendung kennen sollten.

Themen

- Die wichtigsten Konzepte
- Wie CloudFormation funktioniert
- Möglichkeiten für den Einstieg CloudFormation

Die wichtigsten Konzepte

Wenn Sie es verwenden CloudFormation, arbeiten Sie mit Vorlagen und Stacks. Sie erstellen Vorlagen, um Ihre AWS Ressourcen und ihre Eigenschaften zu beschreiben. Jedes Mal, wenn Sie einen Stapel erstellen CloudFormation , werden die Ressourcen bereitgestellt, die in Ihrer Vorlage beschrieben sind.

Themen

- Vorlagen
- Stacks
- Änderungssätze

Vorlagen

Eine CloudFormation Vorlage ist eine Textdatei im YAML- oder JSON-Format. Sie können diese Dateien mit einer beliebigen Erweiterung speichern, z. B.`.yaml`, `.json.template`, oder. `.txt` CloudFormation verwendet diese Vorlagen als Vorlagen für den Aufbau Ihrer AWS Ressourcen. In einer Vorlage können Sie beispielsweise eine EC2 Amazon-Instance beschreiben, z. B. den Instance-Typ, die AMI-ID, Blockgerätezuordnungen und den Namen des EC2 Amazon-Schlüsselpaars. Jedes Mal, wenn Sie einen Stack erstellen, geben Sie auch eine Vorlage an, die CloudFormation verwendet, um das zu erstellen, was Sie in der Vorlage beschrieben haben.

Wenn Sie beispielsweise einen Stack mit der folgenden Vorlage erstellt haben, CloudFormation wird eine Instance mit einer `ami-0ff8a91507f77f867` AMI-ID, einem `t2.micro` Instance-Typ, einem `testkey` Schlüsselpaarnamen und einem Amazon EBS-Volume bereitgestellt.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: A sample template
Resources:
  MyEC2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
```

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "A sample template",
    "Resources": {
        "MyEC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId": "ami-0ff8a91507f77f867",
                "InstanceType": "t2.micro",
                "KeyName": "testkey",
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sdm",
                        "Ebs": {
                            "VolumeType": "io1",
                            "Iops": 200,
                            "DeleteOnTermination": false,
                            "VolumeSize": 20
                        }
                    }
                ]
            }
        }
    }
}
```

Sie können auch mehrere Ressourcen in einer einzigen Vorlage angeben und diese Ressourcen so konfigurieren, dass sie zusammenarbeiten. Sie können beispielsweise die vorherige Vorlage so ändern, dass sie eine Elastic IP-Adresse (EIP) enthält, und sie der EC2 Amazon-Instance zuordnen, wie im folgenden Beispiel gezeigt:

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: A sample template
Resources:
  MyEC2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
```

```
      KeyName: testkey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
  MyEIP:
    Type: 'AWS::EC2::EIP'
    Properties:
      InstanceId: !Ref MyEC2Instance
```

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "A sample template",
    "Resources": {
        "MyEC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId": "ami-0ff8a91507f77f867",
                "InstanceType": "t2.micro",
                "KeyName": "testkey",
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sdm",
                        "Ebs": {
                            "VolumeType": "io1",
                            "Iops": 200,
                            "DeleteOnTermination": false,
                            "VolumeSize": 20
                        }
                    }
                ]
            }
        },
        "MyEIP": {
            "Type": "AWS::EC2::EIP",
            "Properties": {
                "InstanceId": {
                    "Ref": "MyEC2Instance"
```

```
                }
            }
        }
    }
}
```

Die vorherigen Vorlagen konzentrieren sich auf eine einzelne EC2 Amazon-Instance. CloudFormation Vorlagen verfügen jedoch über zusätzliche Funktionen, mit denen Sie komplexe Ressourcensätze erstellen und diese Vorlagen in mehreren Kontexten wiederverwenden können. Sie können beispielsweise Eingabeparameter hinzufügen, deren Werte bei der Erstellung eines CloudFormation Stacks angegeben werden. Mit anderen Worten: Sie können einen Wert wie den Instance-Typ beim Erstellen eines Stacks und nicht beim Erstellen der Vorlage angeben, sodass die Vorlage in anderen Situationen einfacher wiederverwendet werden kann.

Stacks

Wenn Sie verwenden CloudFormation, verwalten Sie zugehörige Ressourcen als eine einzige Einheit, die als Stapel bezeichnet wird. Sie können eine Sammlung von Ressourcen durch Erstellen, Aktualisieren und Löschen von Stacks erstellen und löschen. Alle Ressourcen in einem Stapel werden durch die CloudFormation Vorlage des Stacks definiert. Angenommen, Sie erstellen eine Vorlage, die eine Auto-Scaling-Gruppe, einen Elastic Load Balancing Load Balancer und eine Amazon Relational Database Service (Amazon RDS)-Datenbank-Instance enthält. Um diese Ressourcen zu erstellen, erstellen Sie einen Stapel, indem Sie die von Ihnen erstellte Vorlage einreichen CloudFormation und all diese Ressourcen für Sie bereitstellen.

Änderungssätze

Wenn Sie Änderungen an den ausgeführten Ressourcen in einem Stack vornehmen müssen, aktualisieren Sie den Stack. Bevor Sie Änderungen an Ihren Ressourcen vornehmen, können Sie einen Änderungssatz generieren, eine Zusammenfassung der von Ihnen vorgeschlagenen Änderungen. Anhand von Änderungssätzen können Sie sehen, wie sich Ihre Änderungen möglicherweise auf die ausgeführten Ressourcen auswirken, bevor Sie sie implementieren, insbesondere bei kritischen Ressourcen.

Wenn Sie beispielsweise den Namen einer Amazon RDS-Datenbank-Instance ändern, CloudFormation wird eine neue Datenbank erstellt und die alte gelöscht. Sofern Sie die Daten in der alten Datenbank nicht gesichert haben, gehen diese verloren. Wenn Sie einen Änderungssatz generieren, sehen Sie, dass die Änderung dazu führt, dass die Datenbank ersetzt wird. So können Sie entsprechend planen, bevor Sie den Stack aktualisieren.

Wie CloudFormation funktioniert

Wenn Sie CloudFormation Ihren Stack erstellen, CloudFormation ruft die zugrunde liegenden Dienste auf, AWS um die in Ihrer Vorlage beschriebenen Ressourcen bereitzustellen und zu konfigurieren. Sie benötigen eine Genehmigung, um diese Ressourcen zu erstellen. Um beispielsweise EC2 Instanzen mithilfe von zu erstellen CloudFormation, benötigen Sie Berechtigungen zum Erstellen von Instanzen. Sie verwalten diese Berechtigungen mit AWS Identity and Access Management(IAM).

Die Aufrufe, die getätigt CloudFormation werden, werden alle von Ihrer Vorlage deklariert. Nehmen wir zum Beispiel an, Sie haben eine Vorlage, die eine EC2 Instanz mit einem `t2.micro` Instanztyp beschreibt. Wenn Sie diese Vorlage verwenden, um einen Stack zu erstellen, CloudFormation ruft sie die Amazon EC2 Create Instance API auf und gibt den Instance-Typ als an`t2.micro`. Das folgende Diagramm bietet eine Übersicht über die CloudFormation -Workflows für die Erstellung von Stacks.

So erstellen Sie einen -Stack

1. Verwenden Sie einen Texteditor, um eine CloudFormation Vorlage im YAML- oder JSON-Format zu erstellen. Die CloudFormation-Vorlage beschreibt die Ressourcen, die sie möchten und deren Einstellungen. Verwenden Sie ihnInfrastructure Composer, um Ihre Vorlage zu visualisieren und zu validieren. Auf diese Weise können Sie sicherstellen, dass Ihre Vorlage ordnungsgemäß strukturiert und frei von Syntaxfehlern ist. Weitere Informationen finden Sie unter Mit CloudFormation Vorlagen arbeiten.
2. Speichern Sie die Vorlage lokal oder in einem Amazon S3 S3-Bucket.
3. Erstellen Sie einen CloudFormation Stapel, indem Sie den Speicherort Ihrer Vorlagendatei angeben, z. B. einen Pfad auf Ihrem lokalen Computer oder eine Amazon S3 S3-URL. Wenn die Vorlage Parameter enthält, können Sie beim Erstellen des Stacks Eingabewerte angeben. Parameter ermöglichen Ihnen Werte in Ihre Vorlage zu übergeben, sodass Sie Ihre Ressourcen jedes Mal anpassen können, wenn Sie einen Stack erstellen.

> **Note**
>
> Wenn Sie eine lokal gespeicherte Vorlagendatei angeben, CloudFormation lädt sie in einen S3-Bucket in Ihrem AWS Konto hoch. CloudFormation erstellt einen Bucket für jede Region, in die Sie eine Vorlagendatei hochladen. Auf die Buckets kann jeder zugreifen, der über Amazon Simple Storage Service (Amazon S3) -Berechtigungen in Ihrem AWS Konto verfügt. Wenn ein von erstellter Bucket bereits vorhanden CloudFormation ist, wird die Vorlage diesem Bucket hinzugefügt.

Sie können Ihren eigenen Bucket verwenden und dessen Berechtigungen durch manuelles Hochladen von Vorlagen in Amazon S3 verwalten. Anschließend geben Sie immer, wenn Sie einen Stack erstellen oder aktualisieren, die Amazon S3-URL einer Vorlagendatei an.

Nachdem alle Ressourcen erstellt wurden, CloudFormation meldet es, dass Ihr Stack erstellt wurde. Sie können dann die Ressourcen in Ihrem Stack verwenden. Wenn das Erstellen des Stacks misslingt, setzt CloudFormation Ihre Änderungen durch Löschen der Ressourcen, die es erstellt hat, zurück.

Aktualisierung eines Stacks mit einem Änderungssatz

Wenn Sie die Ressourcen in Ihrem Stack aktualisieren müssen, können Sie die Stack-Vorlage ändern. Sie müssen keinen neuen Stack erstellen und den alten löschen. Um einen Stack zu aktualisieren, erstellen Sie einen Änderungssatz, indem Sie eine modifizierte Version der ursprünglichen Stack-Vorlage, andere Eingabeparameterwerte oder beides einreichen. CloudFormation vergleicht die geänderte Vorlage mit der Originalvorlage und generiert einen Änderungssatz. Der Änderungssatz listet die vorgeschlagenen Änderungen auf. Nach dem Überprüfen der Änderungen können Sie den Änderungssatz starten um Ihren Stack zu aktualisieren oder Sie können einen neuen Änderungssatz erstellen. Das folgende Diagramm bietet eine Übersicht über den Workflow für die Aktualisierung eines Stacks.

Um einen Stack mit einem Änderungssatz zu aktualisieren

1. Sie können eine CloudFormation Stapelvorlage mithilfe Infrastructure Composer eines Texteditors ändern. Weitere Informationen finden Sie unter Aktualisiere deine Stack-Vorlage.

   Denken Sie beim Aktualisieren Ihrer Vorlage daran, dass Aktualisierungen zu Unterbrechungen führen können. Abhängig von der Ressource und den Eigenschaften, die Sie aktualisieren, kann eine Aktualisierung unterbrechen oder sogar eine vorhandene Ressource ersetzen. Weitere Informationen finden Sie unter Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen.
2. Speichern Sie die CloudFormation Vorlage lokal oder in einem S3-Bucket.
3. Erstellen Sie einen Änderungssatz, indem Sie den Stack, den Sie aktualisieren möchten angeben und den Ort der geänderten Vorlage wie z.B. ein Pfad auf Ihrem lokalen Computer oder

einem Amazon S3-URL. Weitere Informationen zum Erstellen von Änderungssätzen finden Sie unter CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren.

> **Note**
>
> Wenn Sie eine Vorlage angeben, die auf Ihrem lokalen Computer gespeichert ist, CloudFormation wird Ihre Vorlage automatisch in einen S3-Bucket in Ihrem AWS-Konto.

4. Sehen Sie sich den Änderungssatz an, um zu überprüfen, CloudFormation ob er die erwarteten Änderungen vornimmt. Prüfen Sie beispielsweise, ob kritische Stack-Ressourcen ersetzt CloudFormation werden. Sie können so viele Änderungssätze erstellen wie sie benötigen, bis Sie die gewünschten Änderungen eingebunden haben.

> **Important**
>
> Änderungssätze geben nicht an, ob Ihr Stack erfolgreich aktualisiert wird. Ein Änderungssatz prüft beispielsweise nicht, ob Sie ein Kontokontingent überschreiten, ob Sie eine Ressource aktualisieren, die keine Updates unterstützt, oder ob Sie nicht über ausreichende Berechtigungen zum Ändern einer Ressource verfügen, was dazu führen kann, dass ein Stack-Update fehlschlägt.

5. Initiieren Sie den Änderungssatz, den Sie auf Ihren Stack anwenden möchten. CloudFormation aktualisiert Ihren Stack, indem es nur die Ressourcen aktualisiert, die Sie geändert haben, und signalisiert, dass Ihr Stack erfolgreich aktualisiert wurde. Wenn die Aktualisierung des Stacks CloudFormation fehlschlägt, werden die Änderungen rückgängig gemacht, um den Stack auf den letzten bekannten Betriebszustand zurückzusetzen.

Möglichkeiten für den Einstieg CloudFormation

Informationen zum Erstellen eines Hello CloudFormation World-Stacks mit der Konsole finden Sie unterErstelle deinen ersten Stack.

Geführtes Lernen finden Sie im AWS CloudFormation Workshop „Erste Schritte mit“, der praktische Erfahrungen mit der Vorlagenentwicklung bietet.

Melden Sie sich an für ein AWS-Konto

Wenn Sie sich für anmelden AWS, AWS-Konto ist Ihr automatisch für alle Dienste in angemeldet AWS, einschließlich AWS CloudFormation. Wenn Sie AWS-Konto bereits einen haben, fahren Sie mit dem nächsten Thema fort. Wenn Sie noch keine haben AWS-Konto, gehen Sie wie folgt vor, um eine zu erstellen.

Melden Sie sich für eine an 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/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen 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. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers 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 eines Benutzers 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 Administratorbenutzer im IAM Identity Center Benutzerzugriff.

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

Anmelden als Administratorbenutzer

- 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 Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

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

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

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

> **Note**
>
> Weitere Informationen darüber, wie Sie verwalten können, wer Zugriff auf was hat, finden Sie unterSteuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management.

Erstelle deinen ersten Stack

In diesem Thema erfahren Sie, wie Sie Ihren ersten CloudFormation Stack mit dem erstellen AWS Management Console. In diesem Tutorial erfahren Sie, wie Sie grundlegende AWS Ressourcen bereitstellen, Stack-Ereignisse überwachen und Ausgaben generieren.

In diesem Beispiel ist die CloudFormation Vorlage in YAML geschrieben. YAML ist ein für Menschen lesbares Format, das häufig zur Definition von Infrastruktur als Code verwendet wird. Wenn Sie mehr darüber erfahren CloudFormation, werden Sie möglicherweise auch auf andere Vorlagen im JSON-Format stoßen, aber für dieses Tutorial wurde YAML aufgrund seiner Lesbarkeit ausgewählt.

> **Note**
>
> CloudFormation ist kostenlos, aber die von Ihnen erstellten Amazon EC2 - und Amazon S3 S3-Ressourcen werden Ihnen in Rechnung gestellt. Wenn Sie jedoch noch keine Erfahrung damit haben AWS, können Sie das kostenlose Kontingent nutzen, um die Kosten während dieses Lernprozesses zu minimieren oder ganz zu vermeiden.

Themen

- Voraussetzungen
- Erstellen Sie einen CloudFormation Stack mit der Konsole

- Überwachen Sie die Stapelerstellung
- Testen Sie den Webserver
- Fehlerbehebung
- Bereinigen
- Nächste Schritte

Voraussetzungen

- Sie benötigen Zugriff auf einen Benutzer oder eine Rolle AWS-Konto mit einem IAM-Status, der über Berechtigungen zur Nutzung von Amazon EC2, Amazon S3 und CloudFormation oder Administratorzugriff verfügt.
- Sie benötigen eine Virtual Private Cloud (VPC), die Zugriff auf das Internet hat. Für diese Walkthrough-Vorlage ist eine Standard-VPC erforderlich, die automatisch mit neueren Modellen geliefert wird. AWS-Konten Wenn Sie keine Standard-VPC haben oder diese gelöscht wurde, finden Sie im Abschnitt zur Fehlerbehebung in diesem Thema alternative Lösungen.

Erstellen Sie einen CloudFormation Stack mit der Konsole

Um einen Hello CloudFormation World-Stack mit der Konsole zu erstellen

1. Öffnen Sie die CloudFormation-Konsole.
2. Wählen Sie Stapel erstellen aus.
3. Wählen Sie auf der Seite Stack erstellen die Option Aus Infrastructure Composer erstellen und anschließend In Infrastructure Composer erstellen aus. Dadurch gelangen Sie im CloudFormation Konsolenmodus zu Infrastructure Composer, wo Sie die Beispielvorlage hochladen und validieren können.
4. Gehen Sie wie folgt vor, um die Beispielvorlage hochzuladen und zu validieren:

   a. Wählen Sie „Vorlage“. Kopieren Sie dann die folgende CloudFormation Vorlage und fügen Sie sie in den Vorlageneditor ein:

```
AWSTemplateFormatVersion: 2010-09-09
Description: CloudFormation Template for WebServer with Security Group and EC2
 Instance

Parameters:
```

```
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'

  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t3.micro
      - t2.micro
    ConstraintDescription: must be a valid EC2 instance type.

  MyIP:
    Description: Your IP address in CIDR format (e.g. 203.0.113.1/32).
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\/\d{1,2}$'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:
  WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow HTTP access via my IP address
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: !Ref MyIP

  WebServer:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      SecurityGroupIds:
        - !Ref WebServerSecurityGroup
      UserData: !Base64 |
        #!/bin/bash
        yum update -y
```

```
        yum install -y httpd
        systemctl start httpd
        systemctl enable httpd
        echo "<html><body><h1>Hello World!</h1></body></html>" > /var/www/html/
index.html

Outputs:
  WebsiteURL:
    Value: !Join
      - ''
      - - http://
        - !GetAtt WebServer.PublicDnsName
    Description: Website URL
```

Bevor Sie mit dem nächsten Schritt fortfahren, nehmen wir uns einen Moment Zeit, um einen Blick auf die Vorlage zu werfen und einige wichtige CloudFormation Konzepte zu verstehen.

- **`Parameters`**In diesem Abschnitt werden Werte deklariert, die bei der Erstellung des Stacks an die Vorlage übergeben werden können. Später in der Vorlage angegebene Ressourcen verweisen auf diese Werte und verwenden die Daten. Parameter sind eine effektive Methode, um Informationen anzugeben, die Sie nicht in der Vorlage selbst speichern möchten. Sie sind auch eine Möglichkeit, Informationen anzugeben, die für die spezifische Anwendung oder Konfiguration, die Sie bereitstellen, möglicherweise spezifisch sind.
- Die Vorlage definiert die folgenden Parameter:
  - **`LatestAmiId`**— Ruft die neueste Amazon Linux 2 AMI-ID aus dem AWS Systems Manager Parameter Store ab.
  - **`InstanceType`**— Ermöglicht die Auswahl des EC2 Instance-Typs (Standard:`t2.micro`, erlaubt:`t3.micro`,`t2.micro`).
  - **`MyIP`**— Gibt den IP-Adressbereich für den HTTP-Zugriff an (Standard: 0.0.0.0/0, ermöglicht den Zugriff von jeder IP aus).
- Der **`Resources`**Abschnitt enthält die Definitionen der AWS Ressourcen, die Sie mit der Vorlage erstellen möchten. Resourcendeklarationen stellen eine effiziente Möglichkeit dar, alle diese Konfigurationseinstellungen gleichzeitig anzugeben. Wenn Sie Ressourcendeklarationen in eine Vorlage aufnehmen, können Sie alle deklarierten Ressourcen erstellen und konfigurieren, indem Sie diese Vorlage verwenden, um einen Stack zu erstellen. Sie können auch neue Stacks aus derselben Vorlage erstellen, um identische Ressourcenkonfigurationen zu starten.

- Diese-Vorlage erstellt die folgenden Ressourcen:
  - **`WebServerSecurityGroup`**— Eine EC2 Sicherheitsgruppe, die eingehenden HTTP-Verkehr auf Port 80 aus dem angegebenen IP-Bereich zulässt.
  - **`WebServer`**— Eine EC2 Instanz mit der folgenden Konfiguration:
    - Verwendet das neueste Amazon Linux 2 AMI
    - Wendet den ausgewählten Instance-Typ an
    - Fügt `WebServerSecurityGroup` der `SecurityGroupIds` Eigenschaft das hinzu
    - Beinhaltet ein Benutzerdatenskript zur Installation von Apache HTTP Server
- Am Anfang jeder Ressourcen- und Parameterdeklaration wird ein logischer Name angegeben. Beispielsweise `WebServerSecurityGroup` ist der logische Name der EC2 Sicherheitsgruppenressource zugewiesen. Die `Ref` Funktion wird dann verwendet, um Ressourcen und Parameter anhand ihrer logischen Namen in anderen Teilen der Vorlage zu referenzieren. Wenn eine Ressource auf eine andere Ressource verweist, entsteht eine Abhängigkeit zwischen ihnen.
- Der **`Outputs`**Abschnitt definiert benutzerdefinierte Werte, die nach der Stack-Erstellung zurückgegeben werden. Sie können die Ausgabewerte verwenden, um Informationen aus den Ressourcen im Stapel zurückzugeben, z. B. Ressourcen-Identifikatoren oder URLs.
- Die Vorlage definiert eine Ausgabe:
  - **`WebsiteURL`**— Die URL des bereitgestellten Webservers, die unter Verwendung des öffentlichen DNS-Namens der EC2 Instanz erstellt wurde. Die `Join` Funktion hilft dabei, das Feste `http://` mit der Variablen zu einer `PublicDnsName` einzigen Zeichenfolge zu kombinieren, sodass die vollständige URL des Webservers einfach ausgegeben werden kann.

b. Wählen Sie Validieren, um sicherzustellen, dass der YAML-Code gültig ist, bevor Sie die Vorlage hochladen.

c. Wählen Sie als Nächstes Vorlage erstellen, um die Vorlage zu erstellen und sie einem S3-Bucket hinzuzufügen.

d. Notieren Sie sich in dem sich öffnenden Dialogfeld den Namen des S3-Buckets, damit Sie ihn später löschen können. Wählen Sie dann Bestätigen und fahren Sie fort CloudFormation. Dadurch gelangen Sie zur CloudFormation Konsole, in der jetzt der S3-Pfad zu Ihrer Vorlage angegeben ist.

5. Wählen Sie auf der Seite „Stack erstellen“ die Option Weiter aus.

6. Geben Sie auf der Seite „Stack-Details angeben“ einen Namen in das Feld Stackname ein. Der Stack-Name darf keine Leerzeichen enthalten. Wählen Sie für dieses Beispiel **`MyTestStack`**.
7. Geben Sie unter Parameter die Parameterwerte wie folgt an:
   - LatestAmiId: Dies ist standardmäßig auf das neueste Amazon Linux 2-AMI eingestellt.
   - InstanceType: Wählen Sie entweder t2.micro oder t3.micro als Instance-Typ. EC2

     > **Note**
     >
     > Wenn Sie noch nicht damit vertraut sind AWS, können Sie das kostenlose Kontingent nutzen, um eine `t2.micro` Instance zu starten und 12 Monate lang kostenlos zu nutzen (in Regionen, in denen die Instance nicht verfügbar `t2.micro` ist, können Sie eine `t3.micro` Instance im Rahmen des kostenlosen Kontingents nutzen).

   - MyIP: Geben Sie Ihre tatsächliche öffentliche IP-Adresse mit einem `/32` Suffix an. Das `/32` Suffix wird in der CIDR-Notation verwendet, um anzugeben, dass eine einzelne IP-Adresse zulässig ist. Es bedeutet im Wesentlichen, Datenverkehr zu und von dieser bestimmten IP-Adresse und nicht zu anderen IP-Adressen zuzulassen.
8. Wählen Sie zweimal Weiter, um zur Seite Überprüfen und Erstellen zu gelangen. Für dieses Tutorial können Sie die Standardeinstellungen auf der Seite „Stack-Optionen konfigurieren“ unverändert lassen.
9. Überprüfen Sie die Informationen für den Stack. Wenn Sie mit den Einstellungen zufrieden sind, klicken Sie auf Submit (Absenden).

Überwachen Sie die Stapelerstellung

Nachdem Sie Submit ausgewählt haben, CloudFormation beginnt die Erstellung der Ressourcen, die in der Vorlage angegeben sind. Ihr neuer Stack**`MyTestStack`**,, wird in der Liste im oberen Bereich der CloudFormationKonsole angezeigt. Sein Status sollte sein`CREATE_IN_PROGRESS`. Sie können den detaillierten Status für einen Stack anzeigen, indem Sie die entsprechenden Ereignisse anzeigen.

Anzeigen der Ereignisse für den Stack

1. Wählen Sie auf der CloudFormation Konsole den Stack **`MyTestStack`** in der Liste aus.
2. Wählen Sie im Detailbereich des Stacks die Registerkarte Events aus.

Die Konsole aktualisiert die Ereignisliste alle 60 Sekunden automatisch mit den neuesten Ereignissen.

Auf der Registerkarte Events wird jeder wichtige Schritt bei der Erstellung des Stacks sortiert nach dem Zeitpunkt jedes Ereignisses angezeigt, wobei die neuesten Ereignisse oben angezeigt werden.

Das erste Ereignis (unten in der Ereignisliste) ist der Beginn des Prozesses zum Erstellen des Stacks:

```
2024-12-23 18:54 UTC-7 MyTestStack CREATE_IN_PROGRESS User initiated
```

Es folgen Ereignisse, die den Beginn und den Abschluss der Erstellung jeder Ressource darstellen. Die Erstellung der EC2 Instanz führt beispielsweise zu den folgenden Einträgen:

```
2024-12-23 18:59 UTC-7 WebServer CREATE_COMPLETE
```

```
2024-12-23 18:54 UTC-7 WebServer CREATE_IN_PROGRESS Resource creation
initiated
```

Das `CREATE_IN_PROGRESS` Ereignis wird protokolliert, wenn CloudFormation gemeldet wird, dass mit der Erstellung der Ressource begonnen wurde. Das `CREATE_COMPLETE`-Ereignis wird protokolliert, wenn die Ressource erfolgreich erstellt wurde.

Wenn CloudFormation der Stack erfolgreich erstellt wurde, wird oben auf der Registerkarte Ereignisse das folgende Ereignis angezeigt:

```
2024-12-23 19:17 UTC-7 MyTestStack CREATE_COMPLETE
```

Wenn eine Ressource nicht erstellt werden CloudFormation kann, meldet sie ein `CREATE_FAILED` Ereignis und setzt standardmäßig den Stapel zurück und löscht alle Ressourcen, die erstellt wurden. In der Spalte Status Reason wird die Ursache des Fehlers angezeigt.

Nachdem der Stack erstellt wurde, können Sie zur Registerkarte Ressourcen wechseln, um die EC2 Instanz und Sicherheitsgruppe anzuzeigen, die Sie erstellt haben.

Testen Sie den Webserver

Nachdem der Stack erfolgreich erstellt wurde, navigieren Sie in der CloudFormation Konsole zur Registerkarte Ausgaben. Suchen Sie nach dem Feld WebsiteURL. Es wird die öffentliche URL Ihrer EC2 Instanz enthalten.

Öffnen Sie einen Browser und rufen Sie die unter WebsiteURL aufgeführte URL auf. Im Browser sollte eine einfache "Hello World!" Meldung angezeigt werden.

Dadurch wird bestätigt, dass auf Ihrer EC2 Instance Apache HTTP Server ausgeführt wird und eine einfache Webseite bereitgestellt wird.

Fehlerbehebung

Wenn bei der Stack-Erstellung ein Rollback auftritt, kann dies an einer fehlenden VPC liegen. Gehen Sie wie folgt vor, um dieses Problem zu lösen.

Keine Standard-VPC verfügbar

Die Vorlage in dieser exemplarischen Vorgehensweise erfordert eine Standard-VPC. Wenn Ihre Stack-Erstellung aufgrund von VPC- oder Subnetzverfügbarkeitsfehlern fehlschlägt, haben Sie möglicherweise keine Standard-VPC in Ihrem Konto. Ihnen stehen folgende Optionen zur Verfügung:

- Eine neue Standard-VPC erstellen — Sie können eine neue Standard-VPC über die Amazon VPC-Konsole erstellen. Anweisungen finden Sie unter Erstellen einer Standard-VPC im Amazon VPC-Benutzerhandbuch.
- Ändern Sie die Vorlage, um ein Subnetz anzugeben — Wenn Sie eine nicht standardmäßige VPC haben, können Sie die Vorlage ändern, um die VPC und das Subnetz explizit anzugeben. IDs Fügen Sie der Vorlage den folgenden Parameter hinzu:

  ```
  SubnetId:
    Description: The subnet ID to launch the instance into
    Type: AWS::EC2::Subnet::Id
  ```

  Aktualisieren Sie dann die `WebServer` Ressource so, dass sie die Subnetz-ID enthält:

  ```
  WebServer:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      SecurityGroupIds:
        - !Ref WebServerSecurityGroup
      SubnetId: !Ref SubnetId
      UserData: !Base64 |
        #!/bin/bash
        yum update -y
  ```

```
        yum install -y httpd
        systemctl start httpd
        systemctl enable httpd
        echo "<html><body><h1>Hello World!</h1></body></html>" > /var/www/html/
index.html
```

Bei der Erstellung des Stacks müssen Sie ein Subnetz mit Internetzugang angeben, damit der Webserver erreichbar ist.

Bereinigen

Um sicherzustellen, dass Ihnen keine Gebühren für nicht gewünschte Dienste berechnet werden, können Sie den Stack und die entsprechenden Ressourcen löschen. Sie können auch den S3-Bucket löschen, in dem die Vorlage des Stacks gespeichert ist.

Löschen des Stacks und der entsprechenden Ressourcen

1. Öffnen Sie die CloudFormation-Konsole.
2. Wählen Sie auf der Seite Stacks die Option neben dem Namen des von Ihnen erstellten Stacks (**`MyTestStack`**) aus und wählen Sie dann Löschen aus.
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).
4. Überwachen Sie den Fortschritt des Stack-Löschvorgangs auf der Registerkarte Ereignis. Der Status für **`MyTestStack`** wechselt zu `DELETE_IN_PROGRESS`. Wenn das Löschen des Stacks CloudFormation abgeschlossen ist, wird der Stapel aus der Liste entfernt.

Wenn Sie mit der Beispielvorlage fertig sind und Ihren Amazon S3 S3-Bucket nicht mehr benötigen, löschen Sie ihn. Bevor Sie einen Bucket löschen können, müssen Sie ihn zuerst leeren. Beim Leeren eines Buckets werden alle darin enthaltenen Objekte gelöscht.

So leeren und löschen Sie den Amazon S3 S3-Bucket

1. Öffnen Sie die Amazon S3-Konsole.
2. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Buckets.
3. Wählen Sie in der Buckets-Liste die Option neben dem Namen des Buckets aus, den Sie für dieses Tutorial erstellt haben, und wählen Sie dann Leer aus.
4. Bestätigen Sie auf der Seite Bucket leeren, dass Sie den Bucket leeren möchten, indem Sie **`permanently delete`** in das Textfeld eingeben, und wählen Sie dann Leeren.

5. Überwachen Sie den Fortschritt des Vorgangs zum Entleeren von Buckets auf der Seite Bucket entleeren: Status).
6. Um zur Bucket-Liste zurückzukehren, wählen Sie Exit (Beenden) aus.
7. Wählen Sie die Option neben dem Namen des Buckets aus und klicken Sie dann auf Löschen.
8. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den Namen des Buckets ein und wählen Sie dann Bucket löschen.
9. Überwachen Sie den Fortschritt des Bucket-Löschvorgangs in der Buckets-Liste. Wenn Amazon S3 das Löschen des Buckets abgeschlossen hat, entfernt es den Bucket aus der Liste.

Nächste Schritte

Herzlichen Glückwunsch! Sie haben erfolgreich einen Stack erstellt, seine Erstellung überwacht und seine Ausgabe verwendet.

Um weiter zu lernen:

- Erfahren Sie mehr über Vorlagen, damit Sie Ihre eigenen erstellen können. Weitere Informationen finden Sie unter Mit CloudFormation Vorlagen arbeiten.
- Probieren Sie den AWS CloudFormation Workshop „Erste Schritte mit“ aus, um mehr praktische Übungen zur Erstellung von Vorlagen zu erhalten.
- Eine verkürzte Version von Erste Schritte mit finden Sie AWS CloudFormation unterStellen Sie Anwendungen auf Amazon bereit EC2. In diesem Thema wird dasselbe Szenario beschrieben, in dem ein CloudFormation Hilfsskript verwendet wird`cfn-init`, um eine EC2 Amazon-Instance zu booten.

CloudFormation bewährte Verfahren

Bewährte Methoden sind Empfehlungen, die Ihnen helfen können, den gesamten Workflow CloudFormation effektiver und sicherer zu nutzen. Erfahren Sie mehr dazu, wie Sie Stacks planen und organisieren, Vorlagen zum Beschreiben Ihrer Ressourcen und die Software-Anwendungen, die auf ihnen laufen, erstellen und Ihre Stacks und ihre Ressourcen verwalten. Die folgenden bewährten Methoden basieren auf realen Erfahrungen aktueller CloudFormation Kunden.

Planung und Organisation

- Verkürzen der Feedbackschleife zur Verbesserung der Liefergeschwindigkeit
- Stacks nach Lebenszyklus und Besitz organisieren
- Verwenden von Stack-übergreifenden Verweisen zum Export von gemeinsam genutzten Ressourcen
- Verwendung von IAM für die Zugriffskontrolle
- Verwenden von Vorlagen zum Replizieren von Stacks in mehreren Umgebungen
- Prüfen der Kontingente für alle Ressourcentypen
- Verwenden von Modulen, um Ressourcenkonfigurationen wiederzuverwenden

Erstellung von Vorlagen

- Keine Anmeldeinformationen in Vorlagen einbetten
- Verwenden Sie AWS-spezifische Parametertypen
- Verwenden von Parameter-Einschränkungen
- Verwenden Sie Pseudoparameter, um die Portabilität zu fördern.
- Wird `AWS::CloudFormation::Init` zur Bereitstellung von Softwareanwendungen auf EC2 Amazon-Instances verwendet
- Verwenden der neuesten Hilfsskripts
- Validieren von Vorlagen vor ihrer Verwendung

Verwaltung von Stacks

- Verwalte alle Stack-Ressourcen über CloudFormation
- Erstellen von Änderungssätzen vor der Aktualisierung von Stacks
- Verwenden von Stack-Richtlinien
- Wird verwendet AWS CloudTrail , um CloudFormation Anrufe zu protokollieren
- Verwenden von Code-Bewertungen und Revisionskontrollen für die Verwaltung Ihrer Vorlagen

- Aktualisieren Sie Ihre EC2 Amazon-Instances regelmäßig

Verkürzen der Feedbackschleife zur Verbesserung der Liefergeschwindigkeit

Verwenden Sie Methoden und Tools, mit denen Sie die Feedback-Schleife für Ihre Infrastruktur, die Sie mit CloudFormation Vorlagen beschreiben, verkürzen können. Dazu gehört das frühzeitige Linting und Testen Ihrer Vorlagen auf Ihrer Workstation. Wenn Sie dies tun, haben Sie die Möglichkeit, potenzielle Syntax- und Konfigurationsprobleme zu entdecken, noch bevor Sie Ihre Beiträge an ein Quellcode-Repository übermitteln. Die frühzeitige Erkennung solcher Probleme hilft dabei, zu verhindern, dass sie formale Lebenszyklusumgebungen wie Entwicklung, Qualitätssicherung und Produktion erreichen. Dieser schnell scheiternde Ansatz mit frühzeitigem Testen bietet Ihnen die Vorteile, die Wartezeiten für Nacharbeiten zu verkürzen, potenzielle Auswirkungen zu reduzieren und Ihr Vertrauen in erfolgreiche Bereitstellungsvorgänge zu erhöhen.

Zu den Tools, mit denen Sie schnelle Abläufe umsetzen können, gehören AWS CloudFormation Linter (`cfn-lint`) und TaskCatBefehlszeilentools. Das `cfn-lint` Tool bietet Ihnen die Möglichkeit, Ihre CloudFormation Vorlagen anhand der Ressourcenspezifikation zu validieren.AWS CloudFormation Dazu gehört das Überprüfen gültiger Werte für Ressourceneigenschaften, sowie bewährte Methoden. Plug-ins für `cfn-lint` sind für eine Reihe von Code-Editoren verfügbar. Dies gibt Ihnen die Möglichkeit, Probleme in Ihrem Editor zu visualisieren und direktes Linter-Feedback zu erhalten. Sie können sich auch dafür entscheiden, sie `cfn-lint` in die Konfiguration Ihres Quellcode-Repositorys zu integrieren, sodass Sie beim Commit Ihrer Beiträge die Vorlage validieren können. Weitere Informationen finden Sie unter Git Pre-Commit-Validierung von AWS CloudFormation Vorlagen mit `cfn-lint`. Sobald du dein erstes Linting durchgeführt und alle `cfn-lint` eventuell aufgetretenen Probleme behoben hast, kannst du deine Vorlagen testen, indem du programmatisch Stapel in TaskCat den von dir ausgewählten Dateien erstellst. AWS-Regionen TaskCat generiert außerdem einen Bericht mit Noten für jede Region, die Sie ausgewählt haben. pass/fail

Eine praktische Anleitung step-by-step, wie Sie beide Tools verwenden können, um die Feedback-Schleife zu verkürzen, finden Sie im Linting and Testing Lab des Workshops.AWS CloudFormation

Stacks nach Lebenszyklus und Besitz organisieren

Verwenden Sie den Lebenszyklus und die Eigentümerschaft Ihrer AWS Ressourcen, um zu entscheiden, welche Ressourcen in den einzelnen Stapel aufgenommen werden sollen. Anfänglich

können Sie alle Ihre Ressourcen einem Stack zuweisen, aber da Ihr Stack wächst und umfangreicher wird, kann die Verwaltung eines einzelnen Stack mühsam und zeitaufwändig sein. Durch die Gruppierung von Ressourcen mit gemeinsamen Lebenszyklen und Besitzeigenschaften können Eigentümer Änderungen an ihren Ressourcensätzen unter Anwendung ihrer eigenen Prozesse und Zeitpläne vornehmen, ohne andere Ressourcen zu beeinträchtigen.

Stellen Sie sich zum Beispiel ein Team von Entwicklern und Technikern vor, das eine Website besitzt, die auf Amazon EC2 Auto Scaling Scaling-Instances hinter einem Load Balancer gehostet wird. Da die Website einen eigenen Lebenszyklus hat und vom Website-Team gepflegt wird, können Sie einen Stack für die Website und ihre Ressourcen erstellen. Nehmen wir einmal an, dass die Website auch Back-End-Datenbanken verwendet, bei denen sich die Datenbanken in einem separaten Stack befinden, die Eigentum von Datenbankadministratoren sind und von diesen gepflegt werden. Wenn das Website-Team oder das Datenbank-Team seine Ressourcen aktualisieren muss, so können sie dies tun, ohne dass dies Auswirkungen auf die Stacks der anderen hat. Wenn sich alle Ressourcen in einem einzigen Stack befinden, kann sich die Koordination und Übermittlung von Updates schwierig gestalten.

Für weitere Informationen über das Organisieren Ihrer Stacks können Sie zwei übliche Frameworks verwenden: eine mehrschichtige Architektur und eine Service-orientierte Architektur (SOA).

Eine geschichtete Architektur organisiert Stacks in mehrere horizontale Layers, die aufeinander aufbauen, wobei jede Layer über eine Abhängigkeit zur Layer direkt unter ihr verfügt. Sie können also einen oder mehrere Stacks auf jeder Layer haben, aber innerhalb jeder Layer sollten Ihre Stacks über AWS -Ressourcen mit ähnlichem Lebenszyklus und ähnlichen Besitzeigenschaften verfügen.

Bei einer serviceorientierten Architektur können Sie große Business-Probleme in überschaubare Teile organisieren. Jeder dieser Teile ist ein Service mit einem klar definierten Zweck und stellt eine eigenständige Funktionseinheit dar. Sie können diese Services einem Stack mit jeweils eigenem Lebenszyklus und eigene Besitzeigenschaften zuweisen. Diese Services (Stacks) können untereinander verbunden werden, sodass sie miteinander interagieren können.

Verwenden von Stack-übergreifenden Verweisen zum Export von gemeinsam genutzten Ressourcen

Wenn Sie Ihre AWS Ressourcen nach Lebenszyklus und Eigentum organisieren, möchten Sie vielleicht einen Stack erstellen, der Ressourcen verwendet, die sich in einem anderen Stapel befinden. Sie können Werte fest codieren oder Eingabeparameter verwenden, um Ressourcennamen

und IDs zu übergeben. Durch diese Methoden sind Vorlagen jedoch möglicherweise nur schwer erneut zu verwenden oder können den Overhead zum Ausführen des Stack vergrößern. Verwenden Sie stattdessen Stack-übergreifende Verweise, um Ressourcen aus einem Stack zu exportieren und in einem anderen Stack zu verwenden. Stacks können die exportierten Ressourcen verwenden, indem sie sie mithilfe der `Fn::ImportValue`-Funktion aufrufen.

Angenommen, Sie verfügen über ein Netzwerk-Stack mit VPC, Sicherheitsgruppe und Subnetz. Sie möchten, dass alle öffentliche Webanwendungen diese Ressourcen verwenden. Durch den Export der Ressourcen ermöglichen Sie allen Stacks mit öffentlichen Webanwendungen die Nutzung. Weitere Informationen finden Sie unter Holen Sie sich exportierte Ausgaben aus einem bereitgestellten CloudFormation Stack.

Prüfen der Kontingente für alle Ressourcentypen

Stellen Sie vor dem Start eines Stacks sicher, dass Sie alle gewünschten Ressourcen erstellen können, ohne Ihre AWS Kontolimits zu überschreiten. Wenn Sie ein Limit erreichen, CloudFormation wird Ihr Stack erst erfolgreich erstellt, wenn Sie Ihr Kontingent erhöhen oder zusätzliche Ressourcen löschen. Jeder Service kann verschiedene Grenzwerte haben, die Sie vor dem Starten eines Stacks kennen sollten. Beispielsweise können Sie standardmäßig nur 2000 CloudFormation Stacks pro Region in Ihrer AWS-Konto starten. Weitere Informationen zu den Grenzwerten und wie Standard-Grenzwerte erhöht werden können, finden Sie unter AWS -Service-Quotas im Allgemeine AWS-Referenz.

Verwenden von Vorlagen zum Replizieren von Stacks in mehreren Umgebungen

Nachdem Sie Ihre Stacks und Ressourcen eingerichtet haben, können Sie Ihre Vorlagen für die Replikation Ihrer Infrastruktur in mehreren Umgebungen erneut verwenden. Sie können beispielsweise Umgebungen für Entwicklung, Tests und Produktion erstellen, sodass Sie Änderungen testen können, bevor Sie sie in Produktion geben. Um Vorlagen wiederverwendbar zu machen, verwenden Sie die Abschnitte für Parameter, Zuweisungen und Bedingungen, sodass Sie Ihre Stacks bei der Erstellung individuell anpassen können. Beispielsweise können Sie für Ihre Entwicklungsumgebungen einen Instance-Typ für niedrigere Kosten als für Ihre Produktionsumgebung festlegen. Aber alle weiteren Konfigurationen und Einstellungen bleiben gleich. Weitere Informationen zu Parameter, Zuweisungen und Bedingungen finden Sie unter CloudFormation Vorlagenabschnitte.

Verwenden von Modulen, um Ressourcenkonfigurationen wiederzuverwenden

Wenn Ihre Infrastruktur wächst, können häufig wiederkehrende Muster entstehen, in denen Sie dieselben Komponenten in jeder Ihrer Vorlagen deklarieren. Module sind eine Möglichkeit für Sie, Ressourcenkonfigurationen für die Aufnahme in Stack-Vorlagen auf transparente, verwaltbare und wiederholbare Weise zu erstellen. Module können gängige Servicekonfigurationen und Best Practices als modulare, anpassbare Bausteine zusammenfassen, die Sie in Ihre Stack-Vorlagen aufnehmen können.

Diese Bausteine können für eine einzelne Ressource verwendet werden, wie z. B. Best Practices für die Definition einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance, oder sie können für mehrere Ressourcen verwendet werden, um gemeinsame Muster der Anwendungsarchitektur zu definieren. Diese Bausteine können in anderen Modulen verschachtelt werden, sodass Sie Ihre Best Practices zu übergeordneten Bausteinen zusammenfassen können. CloudFormation Module sind in der CloudFormation Registrierung verfügbar, sodass Sie sie wie eine native Ressource verwenden können. Wenn Sie ein CloudFormation Modul verwenden, wird die Modulvorlage zur verbrauchenden Vorlage erweitert, sodass Sie mit einem Ref oder Fn:: auf die Ressourcen innerhalb des Moduls zugreifen könnenGetAtt. Weitere Informationen finden Sie unter Erstellen Sie wiederverwendbare Ressourcenkonfigurationen, die in allen Vorlagen mit CloudFormation Modulen enthalten sein können.

Verwenden Sie AWS-spezifische Parametertypen

Wenn Ihre Vorlage Eingaben für bestehende AWS-spezifische Werte erfordert, wie z. B. eine bestehende Amazon Virtual Private Cloud IDs oder einen EC2 Amazon-Schlüsselpaarnamen, verwenden Sie AWS-spezifische Parametertypen. Sie können beispielsweise einen Parameter als Typ angeben`AWS::EC2::KeyPair::KeyName`, der einen vorhandenen Schlüsselpaarnamen verwendet, der sich in Ihrer AWS-Konto und in der Region befindet, in der Sie den Stack erstellen. CloudFormation kann schnell Werte für AWS-spezifische Parametertypen validieren, bevor Sie Ihren Stack erstellen. Wenn Sie die CloudFormation Konsole verwenden, wird außerdem eine Drop-down-Liste mit gültigen Werten CloudFormation angezeigt, sodass Sie nicht die richtigen VPC- oder Schlüsselpaarnamen nachschlagen IDs oder sich diese merken müssen. Weitere Informationen finden Sie unter Geben Sie vorhandene Ressourcen zur Laufzeit mit von CloudFormation -bereitgestellten Parametertypen an.

Verwenden von Parameter-Einschränkungen

Mithilfe von Einschränkungen können Sie zulässige Eingabewerte so beschreiben, dass alle ungültigen Werte CloudFormation erfasst werden, bevor ein Stack erstellt wird. Sie können z. B. Mindestlänge, Maximallänge und zulässige Muster festlegen. Sie können beispielsweise für einen Datenbank-Benutzername-Wert festlegen Einschränkungen, sodass dieser mindestens aus acht Zeichen bestehen und alphanumerische Zeichen enthalten muss. Weitere Informationen finden Sie unter CloudFormation ParametersVorlagensyntax.

Verwenden Sie Pseudoparameter, um die Portabilität zu fördern.

Verwenden Sie Pseudoparameter in Ihren Vorlagen als Argumente für intrinsische Funktionen wie `Ref` und `Fn::Sub` verwenden. Pseudo-Parameter sind Parameter, die von vordefiniert sind. CloudFormation Sie deklarieren sie nicht in Ihrer Vorlage. Die Verwendung von Pseudo-Parametern in systemeigenen Funktionen erhöht die Portabilität Ihrer Stack-Vorlagen über Regionen und Konten hinweg.

Stellen Sie sich beispielsweise vor, Sie möchten eine Vorlage erstellen, in der Sie für eine bestimmte Ressourceneigenschaft den Amazon-Ressourcennamen (ARN) einer anderen vorhandenen Ressource angeben müssen. In diesem Fall ist die vorhandene Ressource vom Typ AWS Systems Manager Parameter Store mit dem folgenden ARN: `arn:aws:ssm:us-east-1:123456789012:parameter/MySampleParameter`. Sie müssen das ARN-Format an Ihre AWS Zielpartition, Region und Konto-ID anpassen. Anstatt diese Werte fest einzuprogrammieren, können Sie die Pseudoparameter `AWS::Partition`, `AWS::Region`, und `AWS::AccountId` verwenden, um Ihre Vorlage besser portabel zu gestalten. In diesem Fall zeigt Ihnen das folgende Beispiel, wie Sie Elemente in einem ARN verketten mit:. CloudFormation `!Sub 'arn:${AWS::Partition}:ssm:${AWS::Region}:${AWS::AccountId}:parameter/MySampleParameter`

Nehmen wir als weiteres Beispiel an, Sie möchten Ressourcen oder Konfigurationen für mehrere Stacks gemeinsam nutzen. Gehen Sie in diesem Beispiel davon aus, dass Sie ein Subnetz für Ihre VPC erstellt und dann dessen ID zur Verwendung mit anderen Stacks in derselben AWS-Konto Region exportiert haben. In einem anderen Stack verweisen Sie bei der Beschreibung einer EC2 Amazon-Instance auf den exportierten Wert der Subnetz-ID. Ein ausführliches Beispiel für die Verwendung des Ausgabefeldes `Export` und der intrinsischen Funktion `Fn::ImportValue` finden Sie unter Verweisen Sie auf Ressourcenausgaben in einem anderen CloudFormation Stack.

Stack-Exporte müssen pro Konto und Region eindeutig sein. In diesem Fall können Sie also den Pseudoparameter `AWS::StackName` verwenden, um ein Präfix für Ihren Export zu erstellen. Da Stacknamen auch pro Konto und Region eindeutig sein müssen, erhöht die Verwendung dieses Pseudo-Parameters als Präfix die Möglichkeit, einen eindeutigen Exportnamen zu haben, und fördert gleichzeitig einen wiederverwendbaren Ansatz für alle Stacks, aus denen Sie Werte exportieren. Alternativ können Sie ein Präfix Ihrer Wahl verwenden.

Wird **`AWS::CloudFormation::Init`** zur Bereitstellung von Softwareanwendungen auf EC2 Amazon-Instances verwendet

Wenn Sie Stacks starten, können Sie Softwareanwendungen auf EC2 Amazon-Instances mithilfe des `cfn-init` Hilfsskripts und der `AWS::CloudFormation::Init` Ressource installieren und konfigurieren. Wenn Sie `AWS::CloudFormation::Init` verwenden, können Sie die Konfigurationen, die Sie wünschen, beschreiben statt die einzelnen Verfahrensschritte. Sie können Konfigurationen auch aktualisieren, ohne Instances neu zu erstellen. Und falls bei Ihrer Konfiguration etwas schief geht, CloudFormation generiert es Protokolle, anhand derer Sie Probleme untersuchen können.

Legen Sie in Ihrer Vorlage den Installations- und Konfigurationsstatus in der Ressource `AWS::CloudFormation::Init` fest. Eine exemplarische Vorgehensweise zur Verwendung von `cfn-init` und finden Sie `AWS::CloudFormation::Init` unterStellen Sie Anwendungen auf Amazon bereit EC2.

Verwenden der neuesten Hilfsskripts

Die CloudFormation Hilfsskripte werden regelmäßig aktualisiert. Stellen Sie sicher, dass Sie den folgenden Befehl in die `UserData`-Eigenschaft Ihrer Vorlage aufnehmen, bevor Sie die Hilfsskripte aufrufen, um sicherzustellen, dass Ihre gestarteten Instances die neuesten Hilfsskripte erhalten:

```
yum install -y aws-cfn-bootstrap
```

Weitere Informationen zum Herunterladen der neuesten Hilfsskripte finden Sie in der Referenz zu CloudFormation Hilfsskripten im AWS CloudFormation Vorlagenreferenzhandbuch.

Validieren von Vorlagen vor ihrer Verwendung

Bevor Sie eine Vorlage verwenden, um einen Stack zu erstellen oder zu aktualisieren, können Sie ihn CloudFormation zur Validierung verwenden. Die Validierung einer Vorlage kann Ihnen helfen, Syntax catch und einige semantische Fehler, wie z. B. zirkuläre Abhängigkeiten, zu erkennen, bevor Ressourcen CloudFormation erstellt werden. Wenn Sie die CloudFormation Konsole verwenden, validiert die Konsole die Vorlage automatisch, nachdem Sie die Eingabeparameter angegeben haben. Verwenden Sie für die AWS CLI CloudFormation API den validate-templateCLI-Befehl oder die ValidateTemplateAPI-Operation.

Prüft bei der Validierung CloudFormation zunächst, ob es sich bei der Vorlage um ein gültiges JSON handelt. Wenn dies nicht der Fall ist, wird von CloudFormation geprüft, ob die Vorlage gültiger YAML-Code ist. Wenn beide Prüfungen fehlschlagen, wird ein Fehler bei der Vorlagenvalidierung CloudFormation zurückgegeben.

Prüfen von Vorlagen auf die Einhaltung von Organisationsrichtlinien

Sie können auch überprüfen, ob Ihre Vorlage den Richtlinien der Organisation entspricht. AWS CloudFormation Guard (`cfn-guard`) ist ein Open-Source-Tool für die Befehlszeilenschnittstelle (CLI), das eine policy-as-code Sprache zur Definition von Regeln bereitstellt, mit denen sowohl nach erforderlichen als auch nach verbotenen Ressourcenkonfigurationen gesucht werden kann. Anschließend können Sie Ihre Vorlagen anhand dieser Regeln validieren. Beispielsweise können Administratoren Regeln erstellen, um sicherzustellen, dass Benutzer immer verschlüsselte Amazon S3-Buckets erstellen.

Sie können es `cfn-guard` entweder lokal bei der Bearbeitung von Vorlagen oder automatisch als Teil einer CI/CD Pipeline verwenden, um die Bereitstellung nicht richtlinienkonformer Ressourcen zu unterbinden.

`cfn-guard`Enthält außerdem eine Funktion`rulegen`, mit der Sie Regeln aus vorhandenen kompatiblen CloudFormation Vorlagen extrahieren können.

Weitere Informationen finden Sie im CFN-Guard-Repository unter. GitHub

Verwalte alle Stack-Ressourcen über CloudFormation

Nachdem Sie einen Stack gestartet haben, verwenden Sie die CloudFormation Konsole, API oder AWS CLI, um die Ressourcen in Ihrem Stack zu aktualisieren. Nehmen Sie keine Änderungen an

Stack-Ressourcen außerhalb von vor CloudFormation. Andernfalls stimmen Ihre Stack-Vorlage und der aktuelle Status Ihrer Stack-Ressourcen möglicherweise nicht mehr überein, was zu Fehlern führen kann, wenn Sie den Stack aktualisieren oder löschen. Dies wird als Drift bezeichnet. Wenn eine Änderung an einer Ressource außerhalb der CloudFormation Vorlage vorgenommen wird und Sie den Stack aktualisieren, werden die direkt an der Ressource vorgenommenen Änderungen verworfen und die Ressourcenkonfiguration wird auf die Konfiguration in der Vorlage zurückgesetzt.

Weitere Informationen zu Drift finden Sie unter. Was ist Abweichung?

Weitere Informationen zum Aktualisieren von Stacks finden Sie unterAktualisieren eines Stacks.

Erstellen von Änderungssätzen vor der Aktualisierung von Stacks

Mithilfe von Änderungssätzen können Sie sich vor der Implementierung ein Bild davon machen, wie sich vorgeschlagene Änderungen an einem Stack auf Ihre laufenden Ressourcen auswirken könnten. CloudFormation nimmt keine Änderungen an Ihrem Stack vor, bis Sie den Änderungssatz ausgeführt haben, sodass Sie entscheiden können, ob Sie mit Ihren vorgeschlagenen Änderungen fortfahren oder einen weiteren Änderungssatz erstellen möchten.

Verwenden Sie Änderungssätze, um zu prüfen, wie Ihre Änderungen Ihre laufenden Ressourcen – insbesondere kritische Ressourcen – beeinflussen können. Wenn Sie beispielsweise den Namen einer Amazon RDS-Datenbank-Instance ändern, CloudFormation wird eine neue Datenbank erstellt und die alte gelöscht. Sie verlieren die Daten in der alten Datenbank, sofern Sie sie nicht bereits gesichert haben. Wenn Sie einen Änderungssatz erstellen, werden Sie sehen, dass Ihre Änderung Ihrer Datenbank ersetzt. Auf diese Weise können Sie planen, bevor Sie Ihren Stack aktualisieren. Weitere Informationen finden Sie unter CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren.

Verwenden von Stack-Richtlinien

Stack-Richtlinien helfen Ihnen dabei, kritische Stack-Ressourcen vor unerwünschten Aktualisierungen zu schützen, durch die Ressourcen unterbrochen oder sogar ersetzt werden können. Eine Stack-Richtlinie ist ein JSON-Dokument, in dem dargelegt wird, welche Aktionen auf designierten Ressourcen ausgeführt werden können. Geben Sie bei der Erstellung eines Stack mit kritischen Ressourcen immer auch eine Stack-Richtlinie an.

Bei der Aktualisierung eines Stack müssen Sie die geschützten Ressourcen, die Sie aktualisieren möchten, explizit angeben. Andernfalls werden keine Änderungen an geschützten Ressourcen

vorgenommen. Weitere Informationen finden Sie unter Verhindern von Aktualisierungen der Stack-Ressourcen.

Verwenden von Code-Bewertungen und Revisionskontrollen für die Verwaltung Ihrer Vorlagen

Ihre Stack-Vorlagen beschreiben die Konfiguration Ihrer AWS Ressourcen, z. B. deren Eigenschaftswerte. Um Änderungen zu prüfen und den genauen Verlauf Ihrer Ressourcen zu pflegen, verwenden Sie Prüfungen und Revisionskontrollen. Diese Methoden können Ihnen dabei helfen, Änderungen zwischen verschiedenen Versionen Ihrer Vorlagen zu verfolgen, die Ihnen wiederum dabei helfen, Änderungen an Ihren Stack-Ressourcen zu verfolgen. Außerdem können Sie durch die Pflege des Verlaufs Ihren Stack stets auf eine bestimmte Version Ihrer Vorlage zurücksetzen.

Aktualisieren Sie Ihre EC2 Amazon-Instances regelmäßig

Führen Sie auf all Ihren Amazon EC2 Windows-Instances und Amazon EC2 Linux-Instances CloudFormation, die mit erstellt wurden, regelmäßig den `yum update` Befehl zum Aktualisieren des RPM-Pakets aus. Auf diese Weise wird sichergestellt, dass Sie die neuesten Fixes und Sicherheits-Updates erhalten.

Mit CloudFormation Vorlagen arbeiten

Eine AWS CloudFormation Vorlage definiert die AWS Ressourcen, die Sie als Teil eines Stacks erstellen, aktualisieren oder löschen möchten. Sie besteht aus mehreren Abschnitten, aber der einzige erforderliche Abschnitt ist der Resources Abschnitt, der mindestens eine Ressource deklarieren muss.

Sie können Vorlagen mit den folgenden Methoden erstellen:

- AWS-Infrastruktur-Composer— Eine visuelle Oberfläche zum Entwerfen von Vorlagen.
- Texteditor — Schreiben Sie Vorlagen direkt in JSON- oder YAML-Syntax.
- IaC-Generator — Generieren Sie Vorlagen aus Ressourcen, die in Ihrem Konto bereitgestellt werden und derzeit nicht von verwaltet werden. CloudFormation Der IaC-Generator funktioniert mit einer Vielzahl von Ressourcentypen, die von der Cloud Control API in Ihrer Region unterstützt werden.

Dieser Abschnitt enthält eine umfassende Anleitung zur Verwendung der verschiedenen Abschnitte einer CloudFormation Vorlage und zum Erstellen von Stack-Vorlagen. Es werden folgende Themen behandelt:

Themen

- Wo Vorlagen gespeichert werden
- Vorlagen werden validiert
- Erste Schritte mit Vorlagen
- Mustervorlagen
- CloudFormation Vorlagenformat
- CloudFormation Vorlagenabschnitte
- Visuelles Erstellen von Vorlagen mit Infrastructure Composer
- Generieren Sie Vorlagen aus vorhandenen Ressourcen mit dem IaC-Generator
- Ruft mithilfe dynamischer Verweise Werte ab, die in anderen Diensten gespeichert sind
- AWS Werte mithilfe von Pseudo-Parametern abrufen
- Holen Sie sich exportierte Ausgaben aus einem bereitgestellten CloudFormation Stack
- Geben Sie vorhandene Ressourcen zur Laufzeit mit von CloudFormation -bereitgestellten Parametertypen an

- CloudFormation Komplettlösungen
- CloudFormation Schnipsel aus Vorlagen
- Bereitstellen von Windows basierten Stacks mit CloudFormation
- Erweitern Sie die Funktionen Ihrer Vorlage mit CloudFormation bereitgestellten Ressourcentypen
- Erstellen Sie wiederverwendbare Ressourcenkonfigurationen, die in allen Vorlagen mit CloudFormation Modulen enthalten sein können

Wo Vorlagen gespeichert werden

Amazon-S3-Bucket

Sie können CloudFormation Vorlagen in einem Amazon S3 S3-Bucket speichern. Beim Erstellen oder Aktualisieren eines Stacks können Sie die S3-URL der Vorlage angeben, anstatt sie direkt hochzuladen.

Wenn Sie Vorlagen direkt über das AWS Management Console oder hochladen AWS CLI, wird automatisch ein S3-Bucket für Sie erstellt. Weitere Informationen finden Sie unter Einen Stack von der CloudFormation Konsole aus erstellen.

Git-Repository

Mit Git Sync können Sie Vorlagen in einem Git-Repository speichern. Wenn Sie einen Stack erstellen oder aktualisieren, können Sie den Speicherort und den Branch des Git-Repositorys angeben, der die Vorlage enthält, anstatt sie direkt hochzuladen oder auf eine S3-URL zu verweisen. CloudFormation überwacht das angegebene Repository und den angegebenen Branch automatisch auf Vorlagenänderungen. Weitere Informationen finden Sie unter Erstellen Sie mit Git Sync einen Stack aus dem Repository-Quellcode.

Vorlagen werden validiert

Überprüfung der Syntax

Sie können die JSON- oder YAML-Syntax Ihrer Vorlage überprüfen, indem Sie den validate-templateCLI-Befehl verwenden oder Ihre Vorlage auf der Konsole angeben. Die Konsole führt die Validierung automatisch durch. Weitere Informationen finden Sie unter Einen Stack von der CloudFormation Konsole aus erstellen.

Diese Methoden überprüfen jedoch nur die Syntax Ihrer Vorlage und nicht die Eigenschaftswerte, die Sie für eine Ressource angegeben haben.

Zusätzliche Validierungstools

Für komplexere Validierungen und Best-Practice-Prüfungen können Sie zusätzliche Tools verwenden, z. B.:

- CloudFormation Linter (cfn-lint) — Überprüfen Sie Vorlagen anhand der Schemas des Ressourcenanbieters. CloudFormation Beinhaltet die Überprüfung gültiger Werte auf Ressourceneigenschaften und bewährte Methoden.
- CloudFormation Rain (rain fmt) — Formatieren Sie Ihre CloudFormation Vorlagen nach einem konsistenten Standard oder formatieren Sie eine Vorlage von JSON zu YAML (oder YAML zu JSON) neu. Es behält Kommentare bei der Verwendung von YAML bei und stellt die Verwendung systemeigener Funktionen nach Möglichkeit auf die Kurzsyntax um.

Erste Schritte mit Vorlagen

Gehen Sie wie folgt vor, um mit der Erstellung einer CloudFormation Vorlage zu beginnen:

1. Ressourcen auswählen — Identifizieren Sie die AWS Ressourcen, die Sie in Ihren Stack aufnehmen möchten, z. B. EC2 Instanzen VPCs, Sicherheitsgruppen und mehr.
2. Schreiben Sie die Vorlage — Schreiben Sie die Vorlage im JSON- oder YAML-Format und definieren Sie die Ressourcen und ihre Eigenschaften.
3. Speichern Sie die Vorlage — Speichern Sie die Vorlage lokal mit einer Dateierweiterung wie: `.json`, `.yaml`, oder `.txt`.
4. Vorlage validieren — Validieren Sie die Vorlage mit den im Vorlagen werden validiert Abschnitt beschriebenen Methoden.
5. Stapel erstellen — Erstellen Sie einen Stack mit der validierten Vorlage.

Planen Sie, die CloudFormation Vorlagenreferenz zu verwenden

Während Sie Ihre Vorlagen schreiben, finden Sie in der Referenz zu Ressourcen- und Eigenschaftstypen eine Dokumentation zur detaillierten Syntax für verschiedene AWS Ressourcentypen.

Oft benötigen Ihre Stack-Vorlagen systeminterne Funktionen, um Eigenschaftswerte zuzuweisen, die erst zur Laufzeit verfügbar sind, und spezielle Attribute, um das Verhalten von Ressourcen zu steuern. Wenn Sie Ihre Vorlage schreiben, sollten Sie sich an den folgenden Ressourcen orientieren:

- Referenz zu systeminternen Funktionen — Zu den häufig verwendeten systeminternen Funktionen gehören:
  - `Ref`— Ruft den Wert eines Parameters oder die physische ID einer Ressource ab.
  - `Sub`— Ersetzt Platzhalter in Zeichenketten durch tatsächliche Werte.
  - `GetAtt`— Gibt den Wert eines Attributs aus einer Ressource in der Vorlage zurück.
  - `Join`— Verbindet eine Reihe von Werten zu einer einzigen Zeichenfolge.
- Referenz zu Ressourcenattributen — Zu den häufig verwendeten Spezialattributen gehören:
  - `DependsOn`— Verwenden Sie dieses Attribut, um anzugeben, dass eine Ressource nach der anderen erstellt werden muss.
  - `DeletionPolicy`— Verwenden Sie dieses Attribut, um anzugeben, wie mit dem Löschen einer Ressource umgegangen werden CloudFormation soll.

Mustervorlagen

CloudFormation stellt Open-Source-Stack-Vorlagen bereit, die Sie für den Einstieg verwenden können. Weitere Informationen finden Sie unter AWS CloudFormation Beispielvorlagen auf der GitHub Website.

Beachten Sie, dass diese Vorlagen nicht für die Produktion vorgesehen sind. Sie sollten sich die Zeit nehmen, um zu lernen, wie sie funktionieren, sie an Ihre Bedürfnisse anpassen und sicherstellen, dass sie den Compliance-Standards Ihres Unternehmens entsprechen.

Jede Vorlage in diesem Repository besteht die CloudFormation Linter-Prüfungen (cfn-lint) und auch ein grundlegendes AWS CloudFormation Guard Regelwerk, das auf den Center for Internet Security (CIS) Top 20 basiert, mit Ausnahmen für einige Regeln, bei denen es sinnvoll war, das Beispiel auf einen einzigen Anwendungsfall zu konzentrieren.

CloudFormation Vorlagenformat

Sie können CloudFormation Vorlagen in den Formaten JSON oder YAML erstellen. Beide Formate dienen demselben Zweck, bieten jedoch deutliche Vorteile in Bezug auf Lesbarkeit und Komplexität.

- JSON — JSON ist ein leichtes Datenaustauschformat, das für Maschinen einfach zu analysieren und zu generieren ist. Das Lesen und Schreiben kann für Menschen jedoch umständlich werden, insbesondere bei komplexen Konfigurationen. In JSON ist die Vorlage mithilfe verschachtelter Klammern `{}` und Klammern strukturiert, `[]` um Ressourcen, Parameter und andere Komponenten zu definieren. Die Syntax erfordert die explizite Deklaration jedes Elements, wodurch die Vorlage ausführlich werden kann, aber die strikte Einhaltung eines strukturierten Formats gewährleistet wird.
- YAML — YAML ist so konzipiert, dass es für Menschen besser lesbar und weniger ausführlich ist als JSON. Es verwendet Einrückungen anstelle von geschweiften und eckigen Klammern, um die Verschachtelung zu kennzeichnen, was die Visualisierung der Hierarchie von Ressourcen und Parametern erleichtern kann. YAML wird oft wegen seiner Übersichtlichkeit und Benutzerfreundlichkeit bevorzugt, insbesondere wenn es sich um komplexere Vorlagen handelt. Die Tatsache, dass YAML auf Einrückungen setzt, kann jedoch zu Fehlern führen, wenn die Abstände nicht einheitlich sind. Daher muss sorgfältig darauf geachtet werden, dass die Genauigkeit gewahrt bleibt.

Struktur der Vorlage

CloudFormation Vorlagen sind in verschiedene Abschnitte unterteilt, und jeder Abschnitt ist so konzipiert, dass er eine bestimmte Art von Informationen enthält. Einige Abschnitte müssen in einer bestimmten Reihenfolge deklariert werden, und für andere spielt die Reihenfolge keine Rolle. Beim Erstellen Ihrer Vorlage kann es jedoch hilfreich sein, die in den folgenden Beispielen gezeigte logische Reihenfolge zu verwenden, da sich Werte in einem Abschnitt möglicherweise auf Werte aus einem vorherigen Abschnitt beziehen.

Verwenden Sie beim Erstellen von Vorlagen keine doppelten Hauptabschnitte, wie z. B. den `Resources` Abschnitt. Die Vorlage akzeptiert zwar CloudFormation möglicherweise, verhält sich aber bei der Verarbeitung der Vorlage undefiniert und kann Ressourcen falsch bereitstellen oder unerklärliche Fehler zurückgeben.

JSON

Das folgende Beispiel zeigt die Struktur einer Vorlage im JSON-Format mit allen verfügbaren Abschnitten.

```
{
  "AWSTemplateFormatVersion" : "version date",
```

```
  "Description" : "JSON string",

  "Metadata" : {
    template metadata
  },

  "Parameters" : {
    set of parameters
  },

  "Rules" : {
    set of rules
  },

  "Mappings" : {
    set of mappings
  },

  "Conditions" : {
    set of conditions
  },

  "Transform" : {
    set of transforms
  },

  "Resources" : {
    set of resources
  },

  "Outputs" : {
    set of outputs
  }
}
```

YAML

Das folgende Beispiel zeigt die Struktur einer Vorlage im YAML-Format mit allen verfügbaren Abschnitten.

```
---
AWSTemplateFormatVersion: version date
```

```
Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

Kommentare

In Vorlagen im JSON-Format werden Kommentare nicht unterstützt. JSON enthält standardmäßig keine Syntax für Kommentare, was bedeutet, dass Sie Kommentare nicht direkt innerhalb der JSON-Struktur hinzufügen können. Wenn Sie jedoch erläuternde Hinweise oder Dokumentation hinzufügen müssen, können Sie das Hinzufügen von Metadaten in Betracht ziehen. Weitere Informationen finden Sie unter Metadata-Attribut.

In Vorlagen im YAML-Format können Sie mithilfe des Symbols Inline-Kommentare einfügen. #

Das folgende Beispiel zeigt eine YAML-Vorlage mit eingebundenen Kommentaren.

```
AWSTemplateFormatVersion: 2010-09-09
Description: A sample CloudFormation template with YAML comments.
# Resources section
```

```
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      # Linux AMI
      ImageId: ami-1234567890abcdef0
      InstanceType: t2.micro
      KeyName: MyKey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
```

Technische Daten

CloudFormation unterstützt die folgenden JSON- und YAML-Spezifikationen:

JSON

CloudFormation folgt dem ECMA-404 JSON-Standard. Weitere Informationen zum JSON-Format finden Sie unter http://www.json.org.

YAML

CloudFormation unterstützt die YAML-Spezifikation Version 1.1 mit wenigen Ausnahmen. CloudFormation unterstützt die folgenden Funktionen nicht:

- Die Tags `binaryomap`, `pairs`, `set` und `timestamp`
- Aliasnamen
- Hash-Zusammenführungen

Weitere Informationen zu YAML finden Sie unter https://yaml.org/.

Weitere Informationen

Für jede Ressource, die Sie in Ihrer Vorlage angeben, definieren Sie ihre Eigenschaften und Werte mithilfe der spezifischen Syntaxregeln von JSON oder YAML. Weitere Informationen zur Vorlagensyntax der einzelnen Formate finden Sie unter CloudFormation Vorlagenabschnitte.

Reguläre Ausdrücke in CloudFormation Vorlagen verwenden

Sie können reguläre Ausdrücke (allgemein als reguläre Ausdrücke bezeichnet) an einer Reihe von Stellen in Ihren CloudFormation Vorlagen verwenden, z. B. für die `AllowedPattern` Eigenschaft, wenn Sie einen Vorlagenparameter erstellen.

Alle regulären Ausdrücke CloudFormation entsprechen der Java-Regex-Syntax. Eine umfassende Beschreibung der Java-Regex-Syntax und ihrer Konstrukte finden Sie unter java.util.Regex.Pattern.

Wenn Sie Ihre CloudFormation Vorlage in JSON-Syntax schreiben, müssen Sie alle Backslash-Zeichen (\) in Ihrem regulären Ausdruck maskieren, indem Sie einen zusätzlichen Backslash hinzufügen. Das liegt daran, dass JSON Backslashes als Escape-Zeichen interpretiert und Sie sie maskieren müssen, um sicherzustellen, dass sie im regulären Ausdruck als wörtliche Backslashes behandelt werden.

Wenn Sie beispielsweise in Ihrem regulären Ausdruck ein `\d` für eine Ziffer einschließen, müssen Sie in Ihrer JSON-Vorlage `\\d` verwenden.

Im folgenden Beispiel gibt die `AllowedPattern` Eigenschaft einen regulären Ausdruck an, der vier aufeinanderfolgenden Ziffernzeichen () entspricht. `\d{4}` Da der reguläre Ausdruck jedoch in einer JSON-Vorlage definiert ist, muss der umgekehrte Schrägstrich mit einem zusätzlichen Backslash () maskiert werden. `\\d`

```
{
  "Parameters": {
    "MyParameter": {
      "Type": "String",
      "AllowedPattern": "\\d{4}"
    }
  }
}
```

Wenn Sie Ihre CloudFormation Vorlage in YAML-Syntax schreiben, müssen Sie den regulären Ausdruck in einfache Anführungszeichen („) setzen. Es ist kein zusätzliches Escaping erforderlich.

```
Parameters:
  MyParameter:
    Type: String
    AllowedPattern: '\d{4}'
```

> **Note**
>
> Reguläre Ausdrücke in CloudFormation werden nur zu Validierungszwecken in bestimmten Kontexten unterstützt, wie z. `AllowedPattern` Sie werden nicht als Mustervergleichsoperationen CloudFormation in systemeigenen Funktionen unterstützt, z. B.`Fn::Equals`, die nur einen exakten Zeichenkettenvergleich und keinen Mustervergleich durchführen.

CloudFormation Vorlagenabschnitte

Jede CloudFormation Vorlage besteht aus einem oder mehreren Abschnitten, die jeweils einem bestimmten Zweck dienen.

Der Abschnitt Ressourcen ist in jeder CloudFormation Vorlage erforderlich und bildet den Kern der Vorlage. In diesem Abschnitt werden die Stack-Ressourcen und ihre Eigenschaften angegeben, z. B. eine EC2 Amazon-Instance oder ein Amazon S3-Bucket. Jede Ressource ist mit einer eindeutigen logischen ID, einem Typ und spezifischen Konfigurationsdetails definiert.

Der Abschnitt Parameter ist zwar optional, spielt aber eine wichtige Rolle bei der Flexibilisierung von Vorlagen. Er ermöglicht es Benutzern, Werte zur Laufzeit zu übergeben, wenn sie einen Stack erstellen oder aktualisieren. Auf diese Parameter kann in den `Outputs` Abschnitten `Resources` und verwiesen werden, sodass Anpassungen möglich sind, ohne die Vorlage selbst zu ändern. Sie können beispielsweise Parameter verwenden, um Instanztypen oder Umgebungseinstellungen anzugeben, die je nach Bereitstellung variieren.

Der Abschnitt Ausgaben, ebenfalls optional, definiert die Werte, die zurückgegeben werden, wenn die Eigenschaften eines Stacks angezeigt werden. Die Ausgaben enthalten nützliche Informationen wie Ressourcen-Identifikatoren oder URLs, die für betriebliche Zwecke oder zur Integration mit anderen Stacks genutzt werden können. In diesem Abschnitt können Benutzer wichtige Informationen zu den mit der Vorlage erstellten Ressourcen abrufen und verwenden.

Zu den weiteren optionalen Abschnitten gehören Zuordnungen, die wie Nachschlagetabellen zur Verwaltung bedingter Werte funktionieren. Mit Mappings definieren Sie Schlüssel-Wert-Paare und verwenden sie zusammen mit der `Fn::FindInMap` systemeigenen Funktion in den Abschnitten und. `Resources` `Outputs` Dies ist nützlich für Szenarien, in denen Sie Konfigurationen auf der Grundlage von Bedingungen wie der Umgebung anpassen müssen. AWS-Region

Die Abschnitte Metadaten und Regeln werden zwar weniger häufig verwendet, bieten aber zusätzliche Funktionen. `Metadata`kann zusätzliche Informationen über die Vorlage enthalten und `Rules` überprüft gleichzeitig einen Parameter oder eine Kombination von Parametern während der Stack-Erstellung oder Aktualisierung, um sicherzustellen, dass sie bestimmten Kriterien entsprechen. Der Abschnitt „Bedingungen“ erhöht die Flexibilität weiter, indem gesteuert wird, ob bestimmte Ressourcen erstellt werden oder Eigenschaften anhand von Bedingungen wie dem Umgebungstyp ein Wert zugewiesen wird.

Schließlich wird der Abschnitt Transformieren verwendet, um Makros während der Bearbeitung der Vorlage anzuwenden. Für serverlose Anwendungen (auch als Lambda-Anwendungen bezeichnet) gibt es die Version des AWS Serverless Application Model (AWS SAM) an, die verwendet werden soll. Wenn Sie eine Transformation angeben, können Sie die AWS SAM Syntax verwenden, um Ressourcen in Ihrer Vorlage zu deklarieren. Das Modell definiert, welche Syntax verwendet werden darf und wie diese Syntax verarbeitet wird. Sie können die `AWS::Include` Transformation auch verwenden, um Vorlagenausschnitte einzubeziehen, die getrennt von der CloudFormation Hauptvorlage gespeichert werden.

Die folgenden Themen enthalten weitere Informationen und Beispiele für die Verwendung der einzelnen Abschnitte.

Themen

- CloudFormation ResourcesVorlagensyntax
- CloudFormation ParametersVorlagensyntax
- CloudFormation OutputsVorlagensyntax
- CloudFormation MappingsVorlagensyntax
- CloudFormation MetadataVorlagensyntax
- CloudFormation RulesVorlagensyntax
- CloudFormation ConditionsVorlagensyntax
- CloudFormation TransformAbschnitt „Vorlage“
- CloudFormation Versionssyntax für das Vorlagenformat
- CloudFormation DescriptionVorlagensyntax

CloudFormation ResourcesVorlagensyntax

Der `Resources` Abschnitt ist ein erforderlicher Abschnitt auf oberster Ebene in einer CloudFormation Vorlage. Er deklariert die AWS Ressourcen, die Sie als Teil Ihres Stacks bereitstellen und konfigurieren möchten CloudFormation .

Syntax

Der `Resources` Abschnitt verwendet die folgende Syntax:

JSON

```
"Resources" : {
    "LogicalResourceName1" : {
        "Type" : "AWS::ServiceName::ResourceType",
        "Properties" : {
            "PropertyName1" : "PropertyValue1",
            ...
        }
    },

    "LogicalResourceName2" : {
        "Type" : "AWS::ServiceName::ResourceType",
        "Properties" : {
            "PropertyName1" : "PropertyValue1",
            ...
        }
    }
}
```

YAML

```
Resources:
  LogicalResourceName1:
    Type: AWS::ServiceName::ResourceType
    Properties:
      PropertyName1: PropertyValue1
      ...

  LogicalResourceName2:
    Type: AWS::ServiceName::ResourceType
    Properties:
      PropertyName1: PropertyValue1
```

```
    ...
```

Logische ID (auch logischer Name genannt)

Innerhalb einer CloudFormation Vorlage werden Ressourcen anhand ihrer logischen Ressourcennamen identifiziert. Diese Namen müssen alphanumerisch (a-zA-Z0-9) und innerhalb der Vorlage eindeutig sein. Logische Namen werden verwendet, um auf Ressourcen aus anderen Abschnitten der Vorlage zu verweisen.

Ressourcentyp

Jede Ressource muss über ein `Type` Attribut verfügen, das definiert, um welche Art von AWS Ressource es sich handelt. Das `Type` Attribut hat das Format`AWS::ServiceName::ResourceType`. Das `Type` Attribut für einen Amazon S3 S3-Bucket lautet beispielsweise`AWS::S3::Bucket`.

Die vollständige Liste der unterstützten Ressourcentypen finden Sie in der Referenz zu AWS Ressourcen- und Eigenschaftstypen.

ResourceProperties

Ressourceneigenschaften sind zusätzliche Optionen, die Sie angeben können, um Konfigurationsdetails für den jeweiligen Ressourcentyp zu definieren. Einige Eigenschaften sind erforderlich, während andere optional sind. Einige Eigenschaften haben Standardwerte, daher ist die Angabe dieser Eigenschaften optional.

Einzelheiten zu den Eigenschaften, die für die einzelnen Ressourcentypen unterstützt werden, finden Sie in den Themen in der Referenz zu AWS Ressourcen- und Eigenschaftstypen.

Eigenschaftswerte können Literalzeichenfolgen, Zeichenfolgenlisten, Boolesche Werte, dynamische Verweise, Parameterverweise, Pseudoverweise oder der von einer Funktion zurückgegebene Wert sein. Die folgenden Beispiele zeigen, wie Sie verschiedene Eigenschaftswerttypen deklarieren:

JSON

```
"Properties" : {
    "String" : "A string value",
    "Number" : 123,
    "LiteralList" : [ "first-value", "second-value" ],
    "Boolean" : true
}
```

YAML

```
Properties:
  String: A string value
  Number: 123
  LiteralList:
    - first-value
    - second-value
  Boolean: true
```

Physische ID

Zusätzlich zur logischen ID haben bestimmte Ressourcen auch eine physische ID, bei der es sich um den tatsächlich zugewiesenen Namen für diese Ressource handelt, z. B. eine EC2 Instanz-ID oder einen S3-Bucket-Namen. Verwenden Sie die physische IDs , um Ressourcen außerhalb von CloudFormation Vorlagen zu identifizieren, aber erst, nachdem die Ressourcen erstellt wurden. Nehmen wir beispielsweise an, Sie geben einer EC2 Instanzressource die logische ID von`MyEC2Instance`. Beim CloudFormation Erstellen der Instanz CloudFormation wird automatisch eine physische ID (z. B.`i-1234567890abcdef0`) generiert und ihr zugewiesen. Sie können diese physische ID verwenden, um die Instance zu identifizieren und ihre Eigenschaften (wie den DNS-Namen) mithilfe der EC2 Amazon-Konsole einzusehen.

Generiert für Amazon S3 S3-Buckets und viele andere Ressourcen CloudFormation automatisch einen eindeutigen physischen Namen für die Ressource, wenn Sie keinen explizit angeben. Dieser physische Name basiert auf einer Kombination aus dem Namen des CloudFormation Stacks, dem in der CloudFormation Vorlage angegebenen logischen Namen der Ressource und einer eindeutigen ID. Wenn Sie beispielsweise einen Amazon S3 S3-Bucket mit dem logischen Namen `MyBucket` in einem Stack namens haben`MyStack`, CloudFormation könnten Sie den Bucket mit dem folgenden physischen Namen benennen`MyStack-MyBucket-abcdefghijk1`.

Für Ressourcen, die benutzerdefinierte Namen unterstützen, können Sie Ihre eigenen physischen Namen zuweisen, um Ressourcen schnell identifizieren zu können. So können Sie einem S3-Bucket, in dem Protokolle gespeichert werden, zum Beispiel den Namen `MyPerformanceLogs` geben. Weitere Informationen finden Sie unter Namenstyp.

Ressourcen referenzieren

Oft müssen Sie Eigenschaften für eine Ressource festlegen, die auf dem Namen oder der Eigenschaft einer anderen Ressource basieren. Sie können beispielsweise eine EC2 Instanz

erstellen, die EC2 Sicherheitsgruppen verwendet, oder eine CloudFront Distribution, die von einem S3-Bucket unterstützt wird. All diese Ressourcen können in derselben CloudFormation Vorlage erstellt werden.

CloudFormation stellt systemeigene Funktionen bereit, mit denen Sie auf andere Ressourcen und deren Eigenschaften verweisen können. Mit diesen Funktionen können Sie Abhängigkeiten zwischen Ressourcen erstellen und Werte von einer Ressource an eine andere übergeben.

Die Funktion **`Ref`**

Die `Ref` Funktion wird häufig verwendet, um eine identifizierende Eigenschaft von Ressourcen abzurufen, die in derselben CloudFormation Vorlage definiert sind. Was sie zurückgibt, hängt von der Art der Ressource ab. Bei den meisten Ressourcen wird der physische Name der Ressource zurückgegeben. Bei einigen Ressourcentypen kann es jedoch einen anderen Wert zurückgeben, z. B. eine IP-Adresse für eine `AWS::EC2::EIP` Ressource oder einen Amazon-Ressourcennamen (ARN) für ein Amazon SNS-Thema.

Die folgenden Beispiele zeigen, wie die `Ref` Funktion in Eigenschaften verwendet wird. In jedem dieser Beispiele gibt die `Ref` Funktion den tatsächlichen Namen der `LogicalResourceName` Ressource zurück, die an anderer Stelle in der Vorlage deklariert wurde. Das `!Ref` Syntaxbeispiel im YAML-Beispiel ist nur eine kürzere Art, die `Ref` Funktion zu schreiben.

JSON

```
"Properties" : {
    "PropertyName" : { "Ref" : "LogicalResourceName" }
}
```

YAML

```
Properties:
  PropertyName1:
    Ref: LogicalResourceName
  PropertyName2: !Ref LogicalResourceName
```

Ausführlichere Informationen zu der `Ref` Funktion finden Sie unter Ref.

Die Funktion **`Fn::GetAtt`**

Die `Ref` Funktion ist hilfreich, wenn der Parameter oder der zurückgegebene Wert für eine Ressource genau Ihren Wünschen entspricht. Möglicherweise benötigen Sie jedoch andere Attribute

einer Ressource. Wenn Sie beispielsweise eine CloudFront Distribution mit einem S3-Ursprung erstellen möchten, müssen Sie den Bucket-Standort mithilfe einer Adresse im DNS-Stil angeben. Eine Reihe von Ressourcen haben zusätzliche Attribute, deren Werte Sie in Ihrer Vorlage verwenden können. Um diese Attribute zu erhalten, verwenden Sie die `Fn::GetAtt` Funktion.

Die folgenden Beispiele zeigen, wie die `GetAtt` Funktion in Eigenschaften verwendet wird. Die `Fn::GetAtt` Funktion benötigt zwei Parameter, den logischen Namen der Ressource und den Namen des abzurufenden Attributs. Das `!GetAtt` Syntaxbeispiel im YAML-Beispiel ist nur eine kürzere Art, die `GetAtt` Funktion zu schreiben.

JSON

```
"Properties" : {
    "PropertyName" : {
        "Fn::GetAtt" : [ "LogicalResourceName", "AttributeName" ]
    }
}
```

YAML

```
Properties:
  PropertyName1:
    Fn::GetAtt:
      - LogicalResourceName
      - AttributeName
  PropertyName2: !GetAtt LogicalResourceName.AttributeName
```

Ausführlichere Informationen zu der `GetAtt` Funktion finden Sie unter Fn::GetAtt.

Beispiele

Die folgenden Beispiele veranschaulichen, wie Ressourcen deklariert werden und wie CloudFormation Vorlagen auf andere Ressourcen verweisen können, die in derselben Vorlage definiert sind, sowie auf vorhandene AWS Ressourcen.

Themen

- Eine einzelne Ressource mit einem benutzerdefinierten Namen deklarieren
- Referenzieren anderer Ressourcen mit der Funktion Ref
- Referenzieren von Ressourcenattributen mithilfe der Funktion Fn::GetAtt

Eine einzelne Ressource mit einem benutzerdefinierten Namen deklarieren

In den folgenden Beispielen wird eine einzelne Ressource vom Typ `AWS::S3::Bucket` mit dem logischen Namen `MyBucket` deklariert. Die `BucketName` Eigenschaft ist auf gesetzt *`amzn-s3-demo-bucket`* und sollte durch den gewünschten Namen für Ihren S3-Bucket ersetzt werden.

Wenn Sie diese Ressourcendeklaration verwenden, um einen Stack zu erstellen, CloudFormation wird ein Amazon S3 S3-Bucket mit Standardeinstellungen erstellt. Für andere Ressourcen, z. B. eine EC2 Amazon-Instance oder eine Auto Scaling Scaling-Gruppe, CloudFormation sind weitere Informationen erforderlich.

JSON

```
{
    "Resources": {
        "MyBucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "BucketName": "amzn-s3-demo-bucket"
            }
        }
    }
}
```

YAML

```
Resources:
  MyBucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: amzn-s3-demo-bucket
```

Referenzieren anderer Ressourcen mit der Funktion **`Ref`**

Die folgenden Beispiele zeigen eine Ressourcendeklaration, die eine EC2 Instanz und eine Sicherheitsgruppe definiert. Die `Ec2Instance` Ressource verweist mithilfe der `Ref` Funktion als Teil ihrer `SecurityGroupIds` Eigenschaft auf die `InstanceSecurityGroup` Ressource. Sie enthält auch eine vorhandene Sicherheitsgruppe (`sg-12a4c434`), die nicht in der Vorlage deklariert ist. Verwenden Sie Literalzeichenfolgen, um auf eine vorhandene AWS -Ressource zu verweisen.

JSON

```
{
    "Resources": {
        "Ec2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "SecurityGroupIds": [
                    {
                        "Ref": "InstanceSecurityGroup"
                    },
                    "sg-12a4c434"
                ],
                "KeyName": "MyKey",
                "ImageId": "ami-1234567890abcdef0"
            }
        },
        "InstanceSecurityGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "Enable SSH access via port 22",
                "SecurityGroupIngress": [
                    {
                        "IpProtocol": "tcp",
                        "FromPort": 22,
                        "ToPort": 22,
                        "CidrIp": "0.0.0.0/0"
                    }
                ]
            }
        }
    }
}
```

YAML

```
Resources:
  Ec2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      SecurityGroupIds:
        - !Ref InstanceSecurityGroup
        - sg-12a4c434
```

```
      KeyName: MyKey
      ImageId: ami-1234567890abcdef0
  InstanceSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Enable SSH access via port 22
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
```

Referenzieren von Ressourcenattributen mithilfe der Funktion **`Fn::GetAtt`**

Die folgenden Beispiele zeigen eine Ressourcendeklaration, die eine CloudFront Verteilungsressource und einen S3-Bucket definiert. Die `MyDistribution` Ressource gibt den DNS-Namen der `MyBucket` Ressource an, die die `Fn::GetAtt` Funktion verwendet, um das `DomainName` Bucket-Attribut abzurufen. Sie werden feststellen, dass die `Fn::GetAtt` Funktion ihre beiden Parameter in einem Array auflistet. Bei Funktionen mit mehreren Parametern werden diese in einem Array angegeben.

JSON

```
{
  "Resources": {
    "MyBucket": {
      "Type": "AWS::S3::Bucket"
    },
    "MyDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
        "DistributionConfig": {
          "Origins": [
            {
              "DomainName": {
                "Fn::GetAtt": [
                  "MyBucket",
                  "DomainName"
                ]
              },
              "Id": "MyS3Origin",
              "S3OriginConfig": {}
```

```
            }
          ],
          "Enabled": "true",
          "DefaultCacheBehavior": {
            "TargetOriginId": "MyS3Origin",
            "ForwardedValues": {
              "QueryString": "false"
            },
            "ViewerProtocolPolicy": "allow-all"
          }
        }
      }
    }
  }
}
```

YAML

```
Resources:
  MyBucket:
    Type: 'AWS::S3::Bucket'
  MyDistribution:
    Type: 'AWS::CloudFront::Distribution'
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: !GetAtt
              - MyBucket
              - DomainName
            Id: MyS3Origin
            S3OriginConfig: {}
        Enabled: 'true'
        DefaultCacheBehavior:
          TargetOriginId: MyS3Origin
          ForwardedValues:
            QueryString: 'false'
          ViewerProtocolPolicy: allow-all
```

CloudFormation ParametersVorlagensyntax

Mit dem optionalen Abschnitt `Parameters` können Sie Ihre Vorlagen anpassen. Mithilfe von Parametern können Sie bei jeder Erstellung oder Aktualisierung eines Stacks benutzerdefinierte

Werte in Ihre Vorlage eingeben. Durch die Verwendung von Parametern in Ihren Vorlagen können Sie wiederverwendbare und flexible Vorlagen erstellen, die auf bestimmte Szenarien zugeschnitten werden können.

Indem Sie Parameter des entsprechenden Typs definieren, können Sie aus einer Liste mit Kennungen vorhandener Ressourcen auswählen, wenn Sie die Konsole zum Erstellen Ihres Stacks verwenden. Weitere Informationen finden Sie unter Geben Sie vorhandene Ressourcen zur Laufzeit mit von CloudFormation -bereitgestellten Parametertypen an.

Parameter sind eine beliebte Methode zur Angabe von Eigenschaftswerten von Stack-Ressourcen. Es kann jedoch Einstellungen geben, die regional oder aufgrund anderer Bedingungen oder Abhängigkeiten für Benutzer etwas komplex zu ermitteln sind. In diesen Fällen sollten Sie der Vorlage selbst etwas Logik hinzufügen, sodass Benutzer einfachere Werte (oder gar keine) angeben können, um die gewünschten Ergebnisse zu erzielen, z. B. mithilfe eines Mappings. Weitere Informationen finden Sie unter CloudFormation MappingsVorlagensyntax.

Syntax

Sie deklarieren Parameter im `Parameters` Abschnitt einer Vorlage, der die folgende allgemeine Syntax verwendet:

JSON

```
"Parameters" : {
  "ParameterLogicalID" : {
    "Description": "Information about the parameter",
    "Type" : "DataType",
    "Default" : "value",
    "AllowedValues" : ["value1", "value2"]
  }
}
```

YAML

```
Parameters:
  ParameterLogicalID:
    Description: Information about the parameter
    Type: DataType
    Default: value
    AllowedValues:
```

```
      - value1
      - value2
```

Ein Parameter enthält eine Liste von Attributen, die seinen Wert sowie Einschränkungen für diesen Wert definieren. Das einzige erforderliche Attribut ist`Type`, bei dem es sich um einen `StringNumber`, oder CloudFormation angegebenen Parametertyp handeln kann. Sie können auch ein `Description` Attribut hinzufügen, das beschreibt, welche Art von Wert Sie angeben sollten. Der Name und die Beschreibung des Parameters werden auf der Seite „Parameter angeben“ angezeigt, wenn Sie die Vorlage im Assistenten zum Erstellen von Stacks verwenden.

> **Note**
>
> Standardmäßig listet die CloudFormation Konsole Eingabeparameter alphabetisch nach ihrer logischen ID auf. Um diese Standardreihenfolge zu überschreiben und verwandte Parameter zu gruppieren, können Sie den `AWS::CloudFormation::Interface` Metadatenschlüssel in Ihrer Vorlage verwenden. Weitere Informationen finden Sie unter Organisieren von CloudFormation Parametern mit `AWS::CloudFormation::Interface` Metadaten.

CloudFormation Verwendet für Parameter mit Standardwerten die Standardwerte, sofern die Benutzer keinen anderen Wert angeben. Wenn Sie das Standardattribut weglassen, müssen Benutzer einen Wert für diesen Parameter angeben. Wenn der Benutzer jedoch einen Wert eingeben muss, wird nicht sichergestellt, dass der Wert gültig ist. Um den Wert eines Parameters zu überprüfen, können Sie Einschränkungen deklarieren oder einen AWS-spezifischen Parametertyp angeben.

Für Parameter ohne Standardwerte müssen Benutzer bei der Stack-Erstellung einen Schlüsselnamenwert angeben. Wenn nicht, CloudFormation schlägt die Erstellung des Stacks fehl und es wird eine Ausnahme ausgelöst:

```
Parameters: [KeyName] must have values
```

Eigenschaften

`AllowedPattern`

Gibt einen regulären Ausdruck an, der die zulässigen Muster für `String`- oder `CommaDelimitedList`-Typen beschreibt. Bei der Anwendung auf einen Parameter des Typs

`String` muss das Muster mit dem gesamten angegebenen Parameterwert übereinstimmen. Wenn das Muster auf einen Parameter des Typs `CommaDelimitedList` angewendet wird, muss es mit jedem Wert in der Liste übereinstimmen.

Required: No

`AllowedValues`

Gibt ein Array mit einer Liste der für den Parameter zulässigen Werte an. Wenn der Parameterwert auf einen Parameter des Typs `String` angewendet wird, muss er einer der zulässigen Werte sein. Wenn der Parameterwert auf einen Parameter des Typs `CommaDelimitedList` angewendet wird, muss jeder Wert in der Liste einer der ausgewählten zulässigen Werte sein.

Required: No

> **Note**
>
> Wenn Sie YAML verwenden und `Yes`- und `No`-Zeichenfolgen für `AllowedValues` nutzen möchten, geben Sie einfache Anführungszeichen ein, um zu verhindern, dass diese booleschen Werte vom YAML-Parser berücksichtigt werden.

`ConstraintDescription`

Gibt eine Zeichenfolge als Beschreibung für eine Einschränkung an. Diese Beschreibung wird bei Verstößen gegen die Einschränkung angezeigt. Für einen Parameter mit einem zulässigen Muster `[A-Za-z0-9]+` beispielsweise wird ohne Einschränkungsbeschreibung die folgende Fehlermeldung angezeigt, wenn der Benutzer einen ungültigen Wert angibt:

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

Durch das Hinzufügen der Beschreibung einer Einschränkung, wie z. B. dürfen nur Groß- und Kleinbuchstaben und Zahlen enthalten, können Sie die folgende benutzerdefinierte Fehlermeldung anzeigen:

```
Malformed input-Parameter MyParameter must only contain uppercase and
lowercase letters and numbers
```

Required: No

`Default`

Gibt einen Wert des entsprechenden Typs an, den die Vorlage nutzen soll, wenn zum Zeitpunkt der Stack-Erstellung kein Wert angegeben wird. Wenn Sie Einschränkungen für den Parameter definieren, müssen Sie einen Wert angeben, der diese Einschränkungen einhält.

Required: No

`Description`

Gibt eine Zeichenfolge mit maximal 4 000 Zeichen an, die den Parameter beschreibt.

Required: No

`MaxLength`

Gibt einen Ganzzahlwert an, der festlegt, wie viele Zeichen maximal für `String`-Typen zulässig sind.

Required: No

`MaxValue`

Gibt einen numerischen Wert an, der den zulässigen numerischen Maximalwert für `Number`-Typen definiert.

Required: No

`MinLength`

Gibt einen Ganzzahlwert an, der die zulässige Mindestzeichenzahl für `String`-Typen definiert.

Required: No

`MinValue`

Gibt einen numerischen Wert an, der den zulässigen numerischen Mindestwert für `Number`-Typen definiert.

Required: No

`NoEcho`

Gibt an, ob der Parameterwert maskiert werden soll, damit er in der Konsole, in Befehlszeilen-Tools und in der API nicht angezeigt wird. Wenn Sie das `NoEcho` Attribut auf setzen`true`,

wird der als Sternchen (*****) maskierte Parameterwert für alle Aufrufe CloudFormation zurückgegeben, die den Stack oder die Stack-Ereignisse beschreiben, mit Ausnahme von Informationen, die an den unten angegebenen Speicherorten gespeichert sind.

Required: No

> **Important**
>
> Durch die Verwendung des `NoEcho`-Attributs werden keine Informationen maskiert, die im Folgenden gespeichert sind:
>
> - Der Vorlagenbereich. `Metadata` CloudFormation transformiert, modifiziert oder redigiert keine Informationen, die Sie in den `Metadata` Abschnitt aufnehmen. Weitere Informationen finden Sie unter Metadaten.
> - Der `Outputs`-Vorlagenabschnitt. Weitere Informationen finden Sie unter Ausgaben:
> - Das `Metadata`-Attribut einer Ressourcendefinition. Weitere Informationen finden Sie unter `Metadata`-Attribut.
>
> Es wird dringend empfohlen, diese Mechanismen nicht zu verwenden, um vertrauliche Informationen wie Passwörter oder Geheimnisse einzugeben.

> **Important**
>
> Anstatt vertrauliche Informationen direkt in Ihre CloudFormation Vorlagen einzubetten, empfehlen wir Ihnen, dynamische Parameter in der Stack-Vorlage zu verwenden, um auf vertrauliche Informationen zu verweisen, die außerhalb von gespeichert und verwaltet werden CloudFormation, z. B. im AWS Systems Manager Parameterspeicher oder. AWS Secrets Manager
>
> Weitere Informationen finden Sie in den bewährten Methoden zu Keine Anmeldeinformationen in Ihre Vorlagen einbetten.

> **Important**
>
> Wir empfehlen dringend davon ab, `NoEcho`-Parameter oder vertrauliche Daten in Ressourceneigenschaften, die Teil der primären ID einer Ressource sind.
>
> Wenn ein `NoEcho` Parameter in einer Eigenschaft enthalten ist, die eine primäre Ressourcen-ID bildet, CloudFormation kann der tatsächliche Klartext-Wert in der primären

Ressourcen-ID verwendet werden. Diese Ressourcen-ID kann in allen abgeleiteten Ausgaben oder Zielen vorkommen.
Informationen dazu, welche Ressourceneigenschaften die primäre ID eines Ressourcentyps bilden, finden Sie in der Ressourcenreferenzdokumentation für diese Ressource in der Referenz zu AWS Ressourcen- und Eigenschaftstypen. Im Abschnitt Return values (Rückgbewerte) stellt der Rückgabewert der `Ref`-Funktion die Ressourceneigenschaften dar, welche die primäre Kennung des Ressourcentyps bilden.

`Type`

Gibt den Datentyp des Parameters an (`DataType`).

Erforderlich: Ja

CloudFormation unterstützt die folgenden Parametertypen:

`String`

Gibt eine Literalzeichenfolge an. Sie können die folgenden Attribute verwenden, um Einschränkungen zu deklarieren: `MinLengthMaxLength`,`Default`,`AllowedValues`, und`AllowedPattern`.

Zum Beispiel könnten Benutzer `"MyUserName"` angeben.

`Number`

Eine Ganzzahl oder eine Gleitkommazahl. CloudFormation validiert den Parameterwert als Zahl. Wenn Sie den Parameter jedoch an einer anderen Stelle in Ihrer Vorlage verwenden (z. B. mithilfe der `Ref` systeminternen Funktion), wird der Parameterwert zu einer Zeichenfolge.

Sie können die folgenden Attribute verwenden, um Einschränkungen zu deklarieren:`MinValue`,`MaxValue`, `Default` und. `AllowedValues`

Zum Beispiel könnten Benutzer `"8888"` angeben.

`List<Number>`

Ein Array aus Ganzzahlen oder Gleitkommazahlen, die durch Kommas getrennt sind. CloudFormation validiert den Parameterwert als Zahlen; wenn Sie den Parameter jedoch an anderer Stelle in Ihrer Vorlage verwenden (z. B. mit der intrinsischen `Ref`-Funktion), wird der Parameterwert zu einer Liste von Zeichenfolgen.

Zum Beispiel könnten Benutzer `"80,20"` angeben. `Ref` würde dann `["80","20"]` zurückgeben.

`CommaDelimitedList`

Gibt ein Array von Literalzeichenfolgen an, jeweils durch Komma getrennt. Die Gesamtanzahl von Zeichenfolgen sollte um 1 höher sein als die Gesamtanzahl von Kommas. Jede Mitgliedszeichenfolge wird zudem gekürzt.

Zum Beispiel könnten Benutzer `"test,dev,prod"` angeben. `Ref` würde dann `["test","dev","prod"]` zurückgeben.

AWS-spezifische Parametertypen

AWS Werte wie EC2 Amazon-Schlüsselpaarnamen und VPC IDs. Weitere Informationen finden Sie unter Geben Sie vorhandene Ressourcen zur Laufzeit an.

Systems Manager Manager-Parametertypen

Parameter, die vorhandenen Parametern im Systems Manager Parameter Store entsprechen. Sie geben einen Systems Manager Manager-Parameterschlüssel als Wert des Systems Manager Manager-Parametertyps an und CloudFormation rufen den neuesten Wert aus dem Parameterspeicher ab, um ihn für den Stack zu verwenden. Weitere Informationen finden Sie unter Geben Sie vorhandene Ressourcen zur Laufzeit an.

Allgemeine Anforderungen für Parameter

Die folgenden Anforderungen gelten bei der Verwendung von Parametern:

- Eine CloudFormation Vorlage kann maximal 200 Parameter enthalten.
- Jedem Parameter muss ein logischer Name (auch logische ID genannt) zugewiesen werden, der alphanumerisch sein und unter allen logischen Namen innerhalb der Vorlage eindeutig sein muss.
- Jedem Parameter muss ein Parametertyp zugewiesen werden, der von CloudFormation unterstützt wird. Weitere Informationen finden Sie unter Type.
- Jedem Parameter muss zur Laufzeit ein Wert zugewiesen werden CloudFormation , damit der Stack erfolgreich bereitgestellt werden kann. Sie können optional einen Standardwert angeben, der verwendet werden CloudFormation soll, sofern kein anderer Wert angegeben wird.
- Die Parameter innerhalb derselben Vorlage deklariert und referenziert werden. Sie können Parameter aus den Vorlagenabschnitten `Resources` und `Outputs` der Vorlage referenzieren.

Beispiele

Themen

- Einfacher Zeichenkettenparameter
- Passwort-Parameter
- Parameter referenzieren
- Durch Kommas getrennter Listenparameter
- Zurückgeben eines Werts aus einem Parameter für Komma-getrennte Listen

Einfacher Zeichenkettenparameter

Im folgenden Beispiel wird ein Parameter mit dem Namen `InstanceTypeParameter` type `String` deklariert. Mit diesem Parameter können Sie den EC2 Amazon-Instance-Typ für den Stack angeben. Wenn bei der Erstellung oder Aktualisierung des Stacks kein Wert angegeben wird, wird der Standardwert von CloudFormation verwendett`2.micro`.

JSON

```
"Parameters" : {
  "InstanceTypeParameter" : {
    "Description" : "Enter t2.micro, m1.small, or m1.large. Default is t2.micro.",
    "Type" : "String",
    "Default" : "t2.micro",
    "AllowedValues" : ["t2.micro", "m1.small", "m1.large"]
  }
}
```

YAML

```
Parameters:
  InstanceTypeParameter:
    Description: Enter t2.micro, m1.small, or m1.large. Default is t2.micro.
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - m1.small
      - m1.large
```

Passwort-Parameter

Im folgenden Beispiel wird ein Parameter `String` mit dem Namen `DBPwd` Typ ohne Standardwert deklariert. Die `NoEcho` Eigenschaft ist auf gesetzt, um `true` zu verhindern, dass der Parameterwert in Stackbeschreibungen angezeigt wird. Die zulässige Mindestlänge ist `1`, die zulässige Maximallänge `41`. Das Muster erlaubt die Verwendung von Klein- und Großbuchstaben sowie Ziffern. Dieses Beispiel veranschaulicht auch die Verwendung eines regulären Ausdrucks für die `AllowedPattern` Eigenschaft.

JSON

```
"Parameters" : {
  "DBPwd" : {
    "NoEcho" : "true",
    "Description" : "The database admin account password",
    "Type" : "String",
    "MinLength" : "1",
    "MaxLength" : "41",
    "AllowedPattern" : "^[a-zA-Z0-9]*$"
  }
}
```

YAML

```
Parameters:
  DBPwd:
    NoEcho: true
    Description: The database admin account password
    Type: String
    MinLength: 1
    MaxLength: 41
    AllowedPattern: ^[a-zA-Z0-9]*$
```

Parameter referenzieren

Sie verwenden die intrinsische Funktion `Ref`, um auf einen Parameter zu verweisen, und CloudFormation verwendet den Wert des Parameters für die Bereitstellung des Stacks. Sie können Parameter aus den Vorlagenabschnitten `Resources` und `Outputs` derselben Vorlage referenzieren.

Im folgenden Beispiel verweist die `InstanceType` Eigenschaft der EC2 Instanzressource auf den `InstanceTypeParameter` Parameterwert:

JSON

```
"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "InstanceType" : { "Ref" : "InstanceTypeParameter" },
    "ImageId" : "ami-0ff8a91507f77f867"
  }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType:
      Ref: InstanceTypeParameter
    ImageId: ami-0ff8a91507f77f867
```

Durch Kommas getrennter Listenparameter

Der `CommaDelimitedList` Parametertyp kann nützlich sein, wenn Sie mehrere Werte für eine einzelne Eigenschaft angeben müssen. Im folgenden Beispiel wird ein Parameter deklariert, der `DbSubnetIpBlocks` mit einem Standardwert aus drei durch Kommas getrennten CIDR-Blöcken benannt ist.

JSON

```
"Parameters" : {
  "DbSubnetIpBlocks": {
    "Description": "Comma-delimited list of three CIDR blocks",
    "Type": "CommaDelimitedList",
    "Default": "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
  }
}
```

YAML

```
Parameters:
  DbSubnetIpBlocks:
    Description: "Comma-delimited list of three CIDR blocks"
    Type: CommaDelimitedList
```

```
    Default: "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
```

Zurückgeben eines Werts aus einem Parameter für Komma-getrennte Listen

Um auf einen bestimmten Wert in der kommagetrennten Liste eines Parameters zu verweisen, verwenden Sie die `Fn::Select` systeminterne Funktion im Abschnitt Ihrer Vorlage. `Resources` Übergeben Sie den Indexwert des gewünschten Objekts und eine Liste von Objekten, wie im folgenden Beispiel gezeigt.

JSON

```
{
    "Parameters": {
        "VPC": {
            "Type": "String",
            "Default": "vpc-123456"
        },
        "VpcAzs": {
            "Type": "CommaDelimitedList",
            "Default": "us-west-2a, us-west-2b, us-west-2c"
        },
        "DbSubnetIpBlocks": {
            "Type": "CommaDelimitedList",
            "Default": "172.16.0.0/26, 172.16.0.64/26, 172.16.0.128/26"
        }
    },
    "Resources": {
        "DbSubnet1": {
            "Type": "AWS::EC2::Subnet",
            "Properties": {
                "AvailabilityZone": {
                    "Fn::Sub": [
                        "${AWS::Region}${AZ}",
                        {
                            "AZ": {
                                "Fn::Select": [
                                    0,
                                    { "Ref": "VpcAzs" }
                                ]
                            }
                        }
                    ]
                },
```

```
                "VpcId": {
                    "Ref": "VPC"
                },
                "CidrBlock": {
                    "Fn::Select": [
                        0,
                        { "Ref": "DbSubnetIpBlocks" }
                    ]
                }
            }
        },
        "DbSubnet2": {
            "Type": "AWS::EC2::Subnet",
            "Properties": {
                "AvailabilityZone": {
                    "Fn::Sub": [
                        "${AWS::Region}${AZ}",
                        {
                            "AZ": {
                                "Fn::Select": [
                                    1,
                                    { "Ref": "VpcAzs" }
                                ]
                            }
                        }
                    ]
                },
                "VpcId": {
                    "Ref": "VPC"
                },
                "CidrBlock": {
                    "Fn::Select": [
                        1,
                        { "Ref": "DbSubnetIpBlocks" }
                    ]
                }
            }
        },
        "DbSubnet3": {
            "Type": "AWS::EC2::Subnet",
            "Properties": {
                "AvailabilityZone": {
                    "Fn::Sub": [
                        "${AWS::Region}${AZ}",
```

```
                        {
                            "AZ": {
                                "Fn::Select": [
                                    2,
                                    { "Ref": "VpcAzs" }
                                ]
                            }
                        }
                    ]
                },
                "VpcId": {
                    "Ref": "VPC"
                },
                "CidrBlock": {
                    "Fn::Select": [
                        2,
                        { "Ref": "DbSubnetIpBlocks" }
                    ]
                }
            }
        }
    }
}
```

YAML

```
Parameters:
  VPC:
    Type: String
    Default: vpc-123456
  VpcAzs:
    Type: CommaDelimitedList
    Default: us-west-2a, us-west-2b, us-west-2c
  DbSubnetIpBlocks:
    Type: CommaDelimitedList
    Default: 172.16.0.0/26, 172.16.0.64/26, 172.16.0.128/26
Resources:
  DbSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Sub
        - ${AWS::Region}${AZ}
        - AZ: !Select
```

```
            - 0
            - !Ref VpcAzs
      VpcId: !Ref VPC
      CidrBlock: !Select
        - 0
        - !Ref DbSubnetIpBlocks
  DbSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Sub
        - ${AWS::Region}${AZ}
        - AZ: !Select
            - 1
            - !Ref VpcAzs
      VpcId: !Ref VPC
      CidrBlock: !Select
        - 1
        - !Ref DbSubnetIpBlocks
  DbSubnet3:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Sub
        - ${AWS::Region}${AZ}
        - AZ: !Select
            - 2
            - !Ref VpcAzs
      VpcId: !Ref VPC
      CidrBlock: !Select
        - 2
        - !Ref DbSubnetIpBlocks
```

Zugehörige Ressourcen

CloudFormation unterstützt auch die Verwendung dynamischer Verweise zur dynamischen Angabe von Eigenschaftswerten. Beispielsweise müssen Sie möglicherweise auf sichere Zeichenfolgen verweisen, die im Systems Manager Parameter Store gespeichert sind. Weitere Informationen finden Sie unter Ruft mithilfe dynamischer Verweise Werte ab, die in anderen Diensten gespeichert sind.

Sie können auch Pseudo-Parameter innerhalb einer `Ref` oder einer `Sub` Funktion verwenden, um Werte dynamisch aufzufüllen. Weitere Informationen finden Sie unter AWS Werte mithilfe von Pseudo-Parametern abrufen.

CloudFormation OutputsVorlagensyntax

Der optionale `Outputs` Abschnitt deklariert Ausgabewerte für den Stack. Diese Ausgabewerte können auf verschiedene Arten verwendet werden:

- Erfassen Sie wichtige Details zu Ihren Ressourcen — Eine Ausgabe ist eine bequeme Möglichkeit, wichtige Informationen über Ihre Ressourcen zu erfassen. Beispielsweise können Sie den Namen des S3-Buckets eines Stacks ausgeben lassen, um den Bucket einfacher zu finden. Sie können Ausgabewerte auf der Registerkarte Ausgaben der CloudFormation Konsole oder mithilfe des describe-stacksCLI-Befehls anzeigen.
- Stapelübergreifende Referenzen — Sie können Ausgabewerte in andere Stapel importieren, um Verweise zwischen Stapeln zu erstellen. Dies ist hilfreich, wenn Sie Ressourcen oder Konfigurationen für mehrere Stapel gemeinsam nutzen müssen.

> **Important**
>
> CloudFormation redigiert oder verschleiert keine Informationen, die Sie in den Abschnitt aufnehmen. `Outputs` Es wird dringend empfohlen, diesen Abschnitt nicht für die Ausgabe vertraulicher Informationen wie Passwörter oder Secrets zu verwenden.
> Ausgabewerte sind verfügbar, nachdem der Stack-Vorgang abgeschlossen ist. Stack-Ausgabewerte sind nicht verfügbar, wenn sich ein Stack-Status in einem der folgenden Status befindet. `IN_PROGRESS` Wir empfehlen nicht, Abhängigkeiten zwischen einer Servicelaufzeit und dem Stack-Ausgabewert festzulegen, da Ausgabewerte möglicherweise nicht immer verfügbar sind.

Syntax

Der Abschnitt `Outputs` besteht aus dem Schlüsselnamen `Outputs`. Sie können maximal 200 Ausgaben in einer Vorlage deklarieren.

Das folgende Beispiel veranschaulicht die Struktur des Abschnitts `Outputs`.

JSON

Schließen Sie alle Ausgabedeklarationen in Klammern ein. Trennen Sie bei mehreren Ausgaben die einzelnen Ausgaben jeweils durch ein Komma voneinander.

```
"Outputs" : {
  "OutputLogicalID" : {
    "Description" : "Information about the value",
    "Value" : "Value to return",
    "Export" : {
      "Name" : "Name of resource to export"
    }
  }
}
```

YAML

```
Outputs:
  OutputLogicalID:
    Description: Information about the value
    Value: Value to return
    Export:
      Name: Name of resource to export
```

Ausgabefelder

Der Abschnitt `Outputs` kann die folgenden Felder enthalten:

Logische ID (auch logischer Name genannt)

Gibt den Bezeichner der aktuellen Ausgabe an. Die logische ID muss alphanumerisch (`a-z`, `A-Z`, `0-9`) und innerhalb der Vorlage eindeutig sein.

`Description` (optional)

Gibt einen `String`-Typ an, der den Ausgabewert beschreibt. Der Wert für die Deklaration der Beschreibung muss eine Literalzeichenfolge mit einer Länge zwischen 0 und 1 024 Byte sein. Parameter oder Funktionen dürfen nicht zur Angabe der Beschreibung verwendet werden.

`Value` (Erforderlich)

Gibt den Wert der Eigenschaft an, der vom Befehl describe-stacks zurückgegeben wird. Der Wert einer Ausgabe kann Literale, Parameterreferenzen, Pseudoparameter, einen Zuordnungswert oder systeminterne Funktionen enthalten.

`Export` (optional)

Gibt den Namen der Ressourcenausgabe an, wie er für eine Stack-übergreifende Referenzierung exportiert werden soll.

Sie können den `Name`-Wert eines Exports mithilfe intrinsischer Funktionen individuell anpassen.

Weitere Informationen finden Sie unter Holen Sie sich exportierte Ausgaben aus einem bereitgestellten CloudFormation Stack.

Wenn Sie eine Bedingung mit einer Ausgabe verknüpfen möchten, müssen Sie die Bedingung im Abschnitt Conditions der Vorlage definieren.

Beispiele

Die folgenden Beispiele veranschaulichen die Funktionsweise von Stack-Ausgaben.

Themen

- Stack-Ausgabe
- Passen Sie den Exportnamen an mit Fn::Sub
- Passen Sie den Exportnamen an mit Fn::Join
- Gibt eine URL zurück, die erstellt wurde mit Fn::Join

Stack-Ausgabe

Im folgenden Beispiel gibt die Ausgabe `BackupLoadBalancerDNSName` den DNS-Namen der Ressource mit der logischen ID `BackupLoadBalancer` zurück, und zwar nur, wenn die Bedingung `CreateProdResources` als wahr identifiziert wird. Die benannte Ausgabe `InstanceID` gibt die ID der EC2 Instanz mit der logischen ID zurück. `EC2Instance`

JSON

```
"Outputs" : {
  "BackupLoadBalancerDNSName" : {
    "Description": "The DNSName of the backup load balancer",
    "Value" : { "Fn::GetAtt" : [ "BackupLoadBalancer", "DNSName" ]},
    "Condition" : "CreateProdResources"
  },
  "InstanceID" : {
    "Description": "The Instance ID",
    "Value" : { "Ref" : "EC2Instance" }
  }
}
```

YAML

```
Outputs:
  BackupLoadBalancerDNSName:
    Description: The DNSName of the backup load balancer
    Value: !GetAtt BackupLoadBalancer.DNSName
    Condition: CreateProdResources
  InstanceID:
    Description: The Instance ID
    Value: !Ref EC2Instance
```

Passen Sie den Exportnamen an mit **`Fn::Sub`**

In den folgenden Beispielen gibt die Ausgabe mit dem Namen `StackVPC` die ID einer VPC zurück. Anschließend exportiert sie den Wert für Stack-übergreifende Referenzierungen, wobei sie dem Stack-Namen den Namen `VPCID` anfügt.

JSON

```
"Outputs" : {
  "StackVPC" : {
    "Description" : "The ID of the VPC",
    "Value" : { "Ref" : "MyVPC" },
    "Export" : {
      "Name" : {"Fn::Sub": "${AWS::StackName}-VPCID" }
    }
  }
}
```

YAML

```
Outputs:
  StackVPC:
    Description: The ID of the VPC
    Value: !Ref MyVPC
    Export:
      Name: !Sub "${AWS::StackName}-VPCID"
```

Weitere Informationen zur Funktion `Fn::Sub` finden Sie unter Fn::Sub.

Passen Sie den Exportnamen an mit **`Fn::Join`**

Sie können die `Fn::Join` Funktion auch verwenden, um Werte auf der Grundlage von Parametern, Ressourcenattributen und anderen Zeichenketten zu erstellen.

In den folgenden Beispielen wird anstelle der `Fn::Join` Funktion die `Fn::Sub` Funktion verwendet, um den Exportnamen anzupassen. Die `Fn::Join` Beispielfunktion verkettet den Stacknamen mit dem Namen, wobei ein Doppelpunkt als `VPCID` Trennzeichen verwendet wird.

JSON

```
"Outputs" : {
  "StackVPC" : {
    "Description" : "The ID of the VPC",
    "Value" : { "Ref" : "MyVPC" },
    "Export" : {
      "Name" : { "Fn::Join" : [ ":", [ { "Ref" : "AWS::StackName" }, "VPCID" ] ] }
    }
  }
}
```

YAML

```
Outputs:
  StackVPC:
    Description: The ID of the VPC
    Value: !Ref MyVPC
    Export:
      Name: !Join [ ":", [ !Ref "AWS::StackName", VPCID ] ]
```

Weitere Informationen zur Funktion `Fn::Join` finden Sie unter Fn::Join.

Gibt eine URL zurück, die erstellt wurde mit **`Fn::Join`**

Im folgenden Beispiel für eine Vorlage, die eine WordPress Site erstellt, `InstallURL` ist die Zeichenfolge, die von einem `Fn::Join` Funktionsaufruf zurückgegeben wird`http://`, der den DNS-Namen der Ressource `ElasticLoadBalancer` verknüpft, und. `/wp-admin/install.php` Der Ausgabewert sollte folgendermaßen oder ähnlich aussehen:

```
http://mywptests-elasticl-1gb51l6sl8y5v-206169572.aws-region.elb.amazonaws.com/wp-
admin/install.php
```

JSON

```
{
    "Outputs": {
        "InstallURL": {
            "Value": {
                "Fn::Join": [
                    "",
                    [
                        "http://",
                        {
                            "Fn::GetAtt": [
                                "ElasticLoadBalancer",
                                "DNSName"
                            ]
                        },
                        "/wp-admin/install.php"
                    ]
                ]
            },
            "Description": "Installation URL of the WordPress website"
        }
    }
}
```

YAML

```
Outputs:
  InstallURL:
    Value: !Join
      - ''
      - - 'http://'
        - !GetAtt
          - ElasticLoadBalancer
          - DNSName
        - /wp-admin/install.php
    Description: Installation URL of the WordPress website
```

Weitere Informationen zur Funktion `Fn::Join` finden Sie unter Fn::Join.

CloudFormation MappingsVorlagensyntax

Der optionale `Mappings` Abschnitt hilft Ihnen bei der Erstellung von Schlüssel-Wert-Paaren, mit denen Sie Werte angeben können, die auf bestimmten Bedingungen oder Abhängigkeiten basieren.

Ein häufiger Anwendungsfall für den `Mappings` Abschnitt besteht darin, Werte basierend darauf festzulegen, AWS-Region wo der Stack bereitgestellt wird. Dies kann durch die Verwendung des `AWS::Region` Pseudo-Parameters erreicht werden. Der `AWS::Region` Pseudo-Parameter ist ein Wert, der in die Region CloudFormation aufgelöst wird, in der der Stapel erstellt wird. Pseudo-Parameter werden aufgelöst, CloudFormation wenn Sie den Stack erstellen.

Um Werte in einer Map abzurufen, können Sie die `Fn::FindInMap` systeminterne Funktion im `Resources` Abschnitt Ihrer Vorlage verwenden.

Syntax

Der `Mappings` Abschnitt verwendet die folgende Syntax:

JSON

```
"Mappings" : {
  "MappingLogicalName" : {
    "Key1" : {
      "Name" : "Value1"
    },
    "Key2" : {
      "Name" : "Value2"
    },
    "Key3" : {
      "Name" : "Value3"
    }
  }
}
```

YAML

```
Mappings:
  MappingLogicalName:
    Key1:
      Name: Value1
    Key2:
      Name: Value2
```

```
    Key3:
      Name: Value3
```

- `MappingLogicalName`ist der logische Name für das Mapping.
- Innerhalb des Mappings ist jede Map ein Schlüssel, gefolgt von einem anderen Mapping.
- Der Schlüssel muss eine Zuordnung von Name-Wert-Paaren darstellen und innerhalb der Zuordnung eindeutig sein.
- Das Name-Wert-Paar ist ein Etikett und der Zuweisungswert. Durch die Benennung der Werte können Sie einem Schlüssel mehr als einen Satz Werte zuweisen.
- Die Schlüssel in Zuweisungen müssen Literalzeichenfolgen sein.
- Die Werte können vom Typ `String` oder sein`List`.

> **Note**
>
> Im Abschnitt `Mappings` dürfen keine Parameter, Pseudoparameter oder intrinsischen Funktionen verwendet werden.
> Wenn die Werte in einem Mapping derzeit nicht von Ihrem Stack verwendet werden, können Sie das Mapping nicht alleine aktualisieren. Sie müssen Änderungen einbeziehen, die Ressourcen hinzufügen, ändern oder löschen.

Beispiele

Grundlegende Zuweisungen

Das folgende Beispiel zeigt einen `Mappings`-Abschnitt mit einer Map `RegionMap`, die fünf Schlüssel enthält. Diesen Schlüsseln sind Name-Wert-Paare mit jeweils einem einzigen Zeichenfolgewert zugewiesen. Die Schlüssel sind Namen von Regionen. Jedes Name-Wert-Paar ist ein Instance-Typ aus der T-Familie, der in der Region verfügbar ist, die durch den Schlüssel repräsentiert wird. Die Name-Wert-Paare haben einen Namen (`InstanceType`im Beispiel) und einen Wert.

JSON

```
"Mappings" : {
  "RegionMap" : {
    "us-east-1"      : {"InstanceType": "t2.micro"},
    "us-west-1"      : {"InstanceType": "t2.micro"},
    "eu-west-1"      : {"InstanceType": "t2.micro"},
```

```
    "eu-north-1"     : {"InstanceType": "t3.micro"},
    "me-south-1"     : {"InstanceType": "t3.micro"}
  }
}
```

YAML

```
Mappings:
  RegionMap:
    us-east-1:
      InstanceType: t2.micro
    us-west-1:
      InstanceType: t2.micro
    eu-west-1:
      InstanceType: t2.micro
    eu-north-1:
      InstanceType: t3.micro
    me-south-1:
      InstanceType: t3.micro
```

Zuweisung mit mehreren Werten

Das folgende Beispiel enthält Regionsschlüssel, die zwei Wertesätzen zugeordnet sind: einem benannten `MyAMI1` und einem anderen. `MyAMI2`

JSON

```
    "AMIIDMap" : {
      "us-east-1"        : {"MyAMI1" : "ami-0ff8a91507f77f867", "MyAMI2" :
 "ami-0a584ac55a7631c0c"},
      "us-west-1"        : {"MyAMI1" : "ami-0bdb828fd58c52235", "MyAMI2" :
 "ami-066ee5fd4a9ef77f1"},
      "eu-west-1"        : {"MyAMI1" : "ami-047bb4163c506cd98", "MyAMI2" :
 "ami-0a7c483d527806435"},
      "ap-southeast-1"   : {"MyAMI1" : "ami-08569b978cc4dfa10", "MyAMI2" :
 "ami-0be9df32ae9f92309"},
      "ap-northeast-1"   : {"MyAMI1" : "ami-06cd52961ce9f0d85", "MyAMI2" :
 "ami-053cdd503598e4a9d"}
    }
```

YAML

```
AMIIDMap:
```

```
    us-east-1:
      MyAMI1: ami-0ff8a91507f77f867
      MyAMI2: ami-0a584ac55a7631c0c
    us-west-1:
      MyAMI1: ami-0bdb828fd58c52235
      MyAMI2: ami-066ee5fd4a9ef77f1
    eu-west-1:
      MyAMI1: ami-047bb4163c506cd98
      MyAMI2: ami-0a7c483d527806435
    ap-southeast-1:
      MyAMI1: ami-08569b978cc4dfa10
      MyAMI2: ami-0be9df32ae9f92309
    ap-northeast-1:
      MyAMI1: ami-06cd52961ce9f0d85
      MyAMI2: ami-053cdd503598e4a9d
```

Zurückgeben eines Werts aus einer Zuweisung

Sie können die Funktion `Fn::FindInMap` verwenden, um einen benannten Wert auf Basis eines angegebenen Schlüssels zurückzugeben. Die folgende Beispielvorlage enthält eine EC2 Amazon-Ressource, deren `ImageId` Eigenschaft von der `FindInMap` Funktion zugewiesen wird. Die `FindInMap` Funktion gibt Schlüssel als die Region an, in der der Stapel erstellt wird (unter Verwendung des `AWS::Region` Pseudo-Parameters), und `MyAMI1` als Namen des Werts, dem die Zuordnung erfolgen soll. Weitere Informationen zu Pseudoparametern finden Sie unter AWS Werte mithilfe von Pseudo-Parametern abrufen.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
    "AMIIDMap" : {
      "us-east-1"        : {"MyAMI1" : "ami-0ff8a91507f77f867", "MyAMI2" :
 "ami-0a584ac55a7631c0c"},
      "us-west-1"        : {"MyAMI1" : "ami-0bdb828fd58c52235", "MyAMI2" :
 "ami-066ee5fd4a9ef77f1"},
      "eu-west-1"        : {"MyAMI1" : "ami-047bb4163c506cd98", "MyAMI2" :
 "ami-0a7c483d527806435"},
      "ap-northeast-1"   : {"MyAMI1" : "ami-06cd52961ce9f0d85", "MyAMI2" :
 "ami-053cdd503598e4a9d"},
      "ap-southeast-1"   : {"MyAMI1" : "ami-08569b978cc4dfa10", "MyAMI2" :
 "ami-0be9df32ae9f92309"}
```

```
    }
  },
  "Resources" : {
    "myEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : { "Fn::FindInMap" : [ "AMIIDMap", { "Ref" : "AWS::Region" },
 "MyAMI1" ]},
        "InstanceType" : "t2.micro"
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Mappings:
  AMIIDMap:
    us-east-1:
      MyAMI1: ami-0ff8a91507f77f867
      MyAMI2: ami-0a584ac55a7631c0c
    us-west-1:
      MyAMI1: ami-0bdb828fd58c52235
      MyAMI2: ami-066ee5fd4a9ef77f1
    eu-west-1:
      MyAMI1: ami-047bb4163c506cd98
      MyAMI2: ami-0a7c483d527806435
    ap-northeast-1:
      MyAMI1: ami-06cd52961ce9f0d85
      MyAMI2: ami-053cdd503598e4a9d
    ap-southeast-1:
      MyAMI1: ami-08569b978cc4dfa10
      MyAMI2: ami-0be9df32ae9f92309
Resources:
  myEC2Instance:
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: !FindInMap [AMIIDMap, !Ref "AWS::Region", MyAMI1]
      InstanceType: t2.micro
```

Eingabeparameter und **`Fn::FindInMap`**

Sie können einen Eingabeparameter mit der Funktion `Fn::FindInMap` verwenden, um einen bestimmten Wert in einer Map zu referenzieren. Nehmen wir beispielsweise an, Sie haben eine Liste von Regionen und Umgebungstypen, die einer bestimmten Sicherheitsgruppen-ID zugeordnet sind. Sie können die Sicherheitsgruppen-ID, die Ihr Stack verwendet, mithilfe eines Eingabeparameters (`EnvironmentType`) auswählen. Um die Region zu bestimmen, verwenden Sie den `AWS::Region` Pseudo-Parameter, der den Wert abruft, AWS-Region in dem Sie den Stack erstellen.

In diesem Beispiel wird auch ein Systems Manager Manager-Parametertyp deklariert, der einen Systems Manager Manager-Parameteralias (`/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2`) als Standardwert für die `ImageId` Eigenschaft der EC2 Instanz bereitstellt. Dieser Wert wird als AMI-ID-Wert für das neueste Amazon Linux 2-AMI in der Region, in der der Stack erstellt wurde, CloudFormation aufgelöst.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "LatestAmiId" : {
      "Description" : "The latest Amazon Linux 2 AMI from the Parameter Store",
      "Type" : "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default" : "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    },
    "EnvironmentType" : {
      "Description" : "The environment type (Dev or Prod)",
      "Type" : "String",
      "Default" : "Dev",
      "AllowedValues" : [
        "Dev",
        "Prod"
      ]
    }
  },
  "Mappings" : {
    "SecurityGroupMap" : {
      "us-east-1" : {
        "Dev" : "sg-12345678",
        "Prod" : "sg-abcdef01"
      },
      "us-west-2" : {
```

```
        "Dev" : "sg-ghijkl23",
        "Prod" : "sg-45678abc"
      }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : {
          "Ref" : "LatestAmiId"
        },
        "InstanceType" : "t2.micro",
        "SecurityGroupIds" : [{ "Fn::FindInMap" : [ "SecurityGroupMap", { "Ref" :
 "AWS::Region" }, { "Ref" : "EnvironmentType" } ]}]
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
  EnvironmentType:
    Description: The environment type (Dev or Prod)
    Type: String
    Default: Dev
    AllowedValues:
      - Dev
      - Prod
Mappings:
  SecurityGroupMap:
    us-east-1:
      Dev: "sg-12345678"
      Prod: "sg-abcdef01"
    us-west-2:
      Dev: "sg-ghijkl23"
      Prod: "sg-45678abc"
```

```
Resources:
  Ec2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: t2.micro
      SecurityGroupIds:
        - !FindInMap [SecurityGroupMap, !Ref "AWS::Region", !Ref EnvironmentType]
```

Zugehörige Ressourcen

Diese verwandten Themen können bei der Entwicklung von Vorlagen, die diese `Fn::FindInMap` Funktion verwenden, hilfreich sein.

- Fn::FindInMap
- Fn::FindInMap-Verbesserungen
- Fn::Sub

CloudFormation MetadataVorlagensyntax

`Metadata`speichert zusätzliche Informationen mithilfe von JSON- oder YAML-Objekten. Zu den Arten von Metadaten auf Vorlagenebene, die Sie in Ihrer Vorlage verwenden können, gehören:

Benutzerdefinierte Metadaten

Speichert benutzerdefinierte Schlüssel-Wert-Paare. Sie können beispielsweise zusätzliche Informationen angeben, die sich nicht auf die Ressourcenerstellung auswirken, aber zusätzlichen Kontext zu den Infrastruktur-, Team- oder Bereitstellungsspezifikationen bieten.

`AWS::CloudFormation::Interface`

Definiert die Gruppierung und Reihenfolge von Eingabeparametern, wenn sie in der CloudFormation Konsole angezeigt werden. Standardmäßig sortiert die CloudFormation Konsole Parameter alphabetisch nach ihrer logischen ID.

`AWS::CloudFormation::Designer`

AWS CloudFormation Designer (Designer) hat am 5. Februar 2025 das Ende seiner Nutzungsdauer erreicht.

> **Important**
>
> Der Abschnitt `Metadata` kann während einer Stack-Aktualisierung nicht allein aktualisiert werden. Sie können ihn nur aktualisieren, wenn Sie Änderungen einschließen, die Ressourcen hinzufügen, ändern oder löschen.
> CloudFormation transformiert, modifiziert oder redigiert keine Informationen, die Sie in den `Metadata` Abschnitt aufnehmen. Aus diesem Grund wird dringend empfohlen, diesen Abschnitt nicht zu verwenden, um vertrauliche Informationen wie Passwörter oder Secrets zu speichern.

Syntax

Verwenden Sie die folgende Syntax, um benutzerdefinierte Metadaten in Ihrer CloudFormation Vorlage zu deklarieren:

JSON

```
"Metadata" : {
  "Instances" : {"Description" : "Information about the instances"},
  "Databases" : {"Description" : "Information about the databases"}
}
```

YAML

```
Metadata:
  Instances:
    Description: "Information about the instances"
  Databases:
    Description: "Information about the databases"
```

Die Syntax für die `AWS::CloudFormation::Interface` finden Sie unterOrganisieren von CloudFormation Parametern mit `AWS::CloudFormation::Interface` Metadaten.

Organisieren von CloudFormation Parametern mit **`AWS::CloudFormation::Interface`** Metadaten

`AWS::CloudFormation::Interface`ist ein Metadatenschlüssel, der definiert, wie Parameter in der CloudFormation Konsole gruppiert und sortiert werden. Standardmäßig listet die Konsole

Eingabeparameter in alphabetischer Reihenfolge nach ihrer logischen Reihenfolge auf, IDs wenn Sie Stacks in der Konsole erstellen oder aktualisieren. Mit diesem Schlüssel können Sie Ihre eigenen Parameter zum Gruppieren und Sortieren definieren, damit Benutzer effizient Parameterwerte angeben können. Sie könnten beispielsweise alle zugehörigen Parameter in einer Gruppe und alle EC2 VPC-bezogenen Parameter in einer anderen Gruppe gruppieren.

Im Metadatenschlüssel können Sie angeben, welche Gruppen erstellt werden sollen, welche Parameter in die jeweiligen Gruppen eingeschlossen werden sollen und in welcher Reihenfolge die Parameter innerhalb der einzelnen Gruppen in der Konsole angezeigt werden sollen.

Sie können auch Beschriftungen für Parameter definieren. Eine Bezeichnung ist ein Anzeigename oder eine Beschreibung, den die Konsole anstelle der logischen Parameter-ID anzeigt. Anhand von Bezeichnungen können Benutzer die Werte, die für Parameter angegeben werden, einfacher verstehen. Sie können z. B. einen `KeyPair`-Parameter mit `Select an EC2 key pair` bezeichnen.

Alle Parameter, die Sie im Metadatenschlüssel referenzieren, müssen im Abschnitt `Parameters` der Vorlage deklariert sein.

> **Note**
>
> Nur die CloudFormation Konsole verwendet den `AWS::CloudFormation::Interface` Metadatenschlüssel. AWS CLI und API-Aufrufe verwenden diesen Schlüssel nicht.

Syntax

Verwenden Sie die folgende Syntax, um diese Entität in Ihrer CloudFormation Vorlage zu deklarieren:

JSON

```
"Metadata" : {
  "AWS::CloudFormation::Interface" : {
    "ParameterGroups": [
      {
        "Label": {
          "default": "Group Label"
        },
        "Parameters": [
          "Parameter1",
          "Parameter2"
```

```
        ]
      }
    ],
    "ParameterLabels": {
      "Parameter1": {
        "default": "Friendly Name for Parameter1"
      }
    }
  }
}
```

YAML

```
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: Group Label
        Parameters:
          - Parameter1
          - Parameter2
    ParameterLabels:
      Parameter1:
        default: Friendly Name for Parameter1
```

Eigenschaften

`ParameterGroups`

Eine Liste der Parameter-Gruppentypen, für die Sie Gruppennamen, die Parameter in jeder Gruppe und die Reihenfolge, in der die Parameter angezeigt werden, angeben.

Required: No

`Label`

Ein Name für die Parametergruppe.

Required: No

`default`

Das Standardlabel, das die CloudFormation Konsole verwendet, um eine Parametergruppe zu benennen.

Required: No

Typ: Zeichenfolge

`Parameters`

Eine Liste von Parametern, bei denen Groß- und Kleinschreibung beachtet wird IDs , die logisch in die Gruppe aufgenommen werden sollen. Parameter müssen bereits im `Parameters`-Abschnitt der Vorlage definiert sein. Ein Parameter kann nur in eine Parametergruppe eingeschlossen werden.

In der Konsole werden die Parameter, die nicht mit einer Parametergruppe verknüpft werden, in alphabetischer Reihenfolge in der `Other parameters`-Gruppe aufgeführt.

Required: No

Typ: Liste von Zeichenfolgenwerten

`ParameterLabels`

Eine Zuordnung von Parametern und ihren benutzerfreundlichen Namen, die in der CloudFormation Konsole angezeigt wird, wenn ein Stack erstellt oder aktualisiert wird.

Required: No

Parameterbezeichnung

Eine Bezeichnung für einen Parameter. Das Label definiert einen benutzerfreundlichen Namen oder eine Beschreibung, die die CloudFormation Konsole auf der Seite „Parameter angeben“ anzeigt, wenn ein Stack erstellt oder aktualisiert wird. Bei der Parameterbezeichnung muss es sich um die logische ID eines gültigen Parameters handeln, der im `Parameters` Abschnitt der Vorlage deklariert wurde, wobei Groß- und Kleinschreibung beachtet werden muss.

Required: No

`default`

Die Standardbezeichnung, die die CloudFormation Konsole zur Benennung eines Parameters verwendet.

Required: No

Typ: Zeichenfolge

Beispiel

Im folgenden Beispiel werden zwei Parametergruppen definiert: `Network Configuration` und `Amazon EC2 Configuration`. Die `Network Configuration`-Gruppe enthält die Parameter `VPCIDSubnetId` und `SecurityGroupID`, die im Bereich `Parameters` der Vorlage (nicht dargestellt) definiert werden. Die Reihenfolge, in der die Konsole diese Parameter anzeigt, wird durch die Reihenfolge bestimmt, in der die Parameter beginnend mit dem `VPCID`-Parameter aufgeführt sind. Im Beispiel werden die `Amazon EC2 Configuration`-Parameter auf ähnliche Weise gruppiert und sortiert.

Außerdem wird eine Bezeichnung für den `VPCID`-Parameter definiert. Die Konsole zeigt Welchen VPCs sollten diese bereitgestellt werden? statt der logischen ID des Parameters (`VPCID`).

JSON

```
"Metadata" : {
  "AWS::CloudFormation::Interface" : {
    "ParameterGroups" : [
      {
        "Label" : { "default" : "Network Configuration" },
        "Parameters" : [ "VPCID", "SubnetId", "SecurityGroupID" ]
      },
      {
        "Label" : { "default":"Amazon EC2 Configuration" },
        "Parameters" : [ "InstanceType", "KeyName" ]
      }
    ],
    "ParameterLabels" : {
      "VPCID" : { "default" : "Which VPC should this be deployed to?" }
    }
  }
}
```

YAML

```
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: "Network Configuration"
        Parameters:
```

```
          - VPCID
          - SubnetId
          - SecurityGroupID
      - Label:
          default: "Amazon EC2 Configuration"
        Parameters:
          - InstanceType
          - KeyName
    ParameterLabels:
      VPCID:
        default: "Which VPC should this be deployed to?"
```

Parametergruppen in der Konsole

In der folgenden Abbildung wird dargestellt, wie die Konsole die Parametergruppen unter Verwendung des Metadatenschlüssels anzeigt, wenn ein Stack erstellt oder aktualisiert wird: Parametergruppen in der Konsole

CloudFormation RulesVorlagensyntax

Der `Rules` Abschnitt ist ein optionaler Teil einer CloudFormation Vorlage, der eine benutzerdefinierte Validierungslogik ermöglicht. Sofern enthalten, enthält dieser Abschnitt Regelfunktionen, die Parameterwerte validieren, bevor Ressourcen CloudFormation erstellt oder aktualisiert werden.

Regeln sind nützlich, wenn Standardparametereinschränkungen nicht ausreichen. Wenn beispielsweise SSL aktiviert ist, müssen sowohl ein Zertifikat als auch ein Domainname bereitgestellt werden. Eine Regel kann sicherstellen, dass diese Abhängigkeiten erfüllt werden.

Syntax

In `Rules` diesem Abschnitt wird die folgende Syntax verwendet:

JSON

Der Abschnitt `Rules` einer Vorlage besteht aus dem Schlüsselnamen `Rules`, gefolgt von einem einzigen Doppelpunkt. Sie müssen Klammern verwenden, um alle Regeldeklarationen einzuschließen. Wenn Sie mehrere Regeln deklarieren, werden sie durch Kommas getrennt. Für jede Regel deklarieren Sie einen logischen Namen in Anführungszeichen gefolgt von einem Doppelpunkt und Klammern, die die Regelbedingung und Assertionen umschließen.

```
{
    "Rules": {
        "LogicalRuleName1": {
            "RuleCondition": {
                "rule-specific intrinsic function": "Value"
            },
            "Assertions": [
                {
                    "Assert": {
                        "rule-specific intrinsic function": "Value"
                    },
                    "AssertDescription": "Information about this assert"
                },
                {
                    "Assert": {
                        "rule-specific intrinsic function": "Value"
                    },
                    "AssertDescription": "Information about this assert"
                }
            ]
        },
        "LogicalRuleName2": {
            "Assertions": [
                {
                    "Assert": {
                        "rule-specific intrinsic function": "Value"
                    },
                    "AssertDescription": "Information about this assert"
                }
            ]
        }
    }
}
```

YAML

```
Rules:
  LogicalRuleName1:
    RuleCondition:
      rule-specific intrinsic function: Value
    Assertions:
      - Assert:
          rule-specific intrinsic function: Value
```

```
        AssertDescription: Information about this assert
      - Assert:
          rule-specific intrinsic function: Value
        AssertDescription: Information about this assert
  LogicalRuleName2:
    Assertions:
      - Assert:
          rule-specific intrinsic function: Value
        AssertDescription: Information about this assert
```

Felder für Regeln

Der Abschnitt `Rules` kann die folgenden Felder enthalten:

Logische ID (auch logischer Name genannt)

Ein eindeutiger Bezeichner für jede Regel.

`RuleCondition` (optional)

Eine Eigenschaft, die bestimmt, wann eine Regel wirksam wird. Wenn Sie keine Regelbedingung definieren, werden die Assertions der Regel immer wirksam. Für jede Regel können Sie nur eine Regelbedingung definieren.

`Assertions` (Erforderlich)

Eine oder mehrere Anweisungen, die die akzeptablen Werte für einen bestimmten Parameter angeben.

`Assert`

Eine Bedingung, die als erfüllt werden muss`true`.

`AssertDescription`

Eine Meldung, die angezeigt wird, wenn eine Assertion fehlschlägt.

Regelspezifische intrinsische Funktionen

Um Ihre Regeln zu definieren, müssen Sie regelspezifische Funktionen verwenden, d. h. Funktionen, die nur im `Rules` Abschnitt einer Vorlage verwendet werden können. Diese Funktionen können zwar verschachtelt werden, aber das Endergebnis einer Regelbedingung oder Assertion muss entweder oder lauten. `true` `false`

Die folgenden Regelfunktionen sind verfügbar:

- Fn::And
- Fn::Contains
- Fn::EachMemberEquals
- Fn::EachMemberIn
- Fn::Equals
- Fn::If
- Fn::Not
- Fn::Or
- Fn::RefAll
- Fn::ValueOf
- Fn::ValueOfAll

Diese Funktionen werden in der Bedingung oder in den Assertionen einer Regel verwendet. Die Bedingungseigenschaft bestimmt, ob die CloudFormation Assertionen angewendet werden. Wenn die Bedingung zu ausgewertet wird`true`, werden die Assertionen CloudFormation ausgewertet, um zu überprüfen, ob ein Parameterwert gültig ist, wenn ein bereitgestelltes Produkt erstellt oder aktualisiert wird. Wenn ein Parameterwert ungültig ist, wird der Stack von CloudFormation nicht erstellt oder aktualisiert. Wenn die Bedingung als 0 ausgewertet wird`false`, wird der Parameterwert CloudFormation nicht überprüft und der Stack-Vorgang wird fortgesetzt.

Beispiele

Themen

- Bedingtes Überprüfen eines Parameterwerts
- Parameterübergreifende Validierung

Bedingtes Überprüfen eines Parameterwerts

Im folgenden Beispiel überprüfen die beiden Regeln den Wert des Parameters `InstanceType`. Abhängig vom Wert des Umgebungsparameters (`test` oder `prod`) muss der Benutzer `t3.medium` oder `t3.large` für den Parameter `InstanceType` angeben. Die Parameter `InstanceType` und `Environment` müssen im `Parameters`-Abschnitt derselben Vorlage deklariert sein.

JSON

```
{
  "Rules": {
    "testInstanceType": {
      "RuleCondition": {
        "Fn::Equals": [
          {"Ref": "Environment"},
          "test"
        ]
      },
      "Assertions": [
        {
          "Assert": {
            "Fn::Contains": [
              ["t3.medium"],
              {"Ref": "InstanceType"}
            ]
          },
          "AssertDescription": "For a test environment, the instance type must be
 t3.medium"
        }
      ]
    },
    "prodInstanceType": {
      "RuleCondition": {
        "Fn::Equals": [
          {"Ref": "Environment"},
          "prod"
        ]
      },
      "Assertions": [
        {
          "Assert": {
            "Fn::Contains": [
              ["t3.large"],
              {"Ref": "InstanceType"}
            ]
          },
          "AssertDescription": "For a production environment, the instance type must be
 t3.large"
        }
      ]
    }
```

```
  }
}
```

YAML

```
Rules:
  testInstanceType:
    RuleCondition: !Equals
      - !Ref Environment
      - test
    Assertions:
      - Assert:
          'Fn::Contains':
            - - t3.medium
            - !Ref InstanceType
        AssertDescription: 'For a test environment, the instance type must be
 t3.medium'
  prodInstanceType:
    RuleCondition: !Equals
      - !Ref Environment
      - prod
    Assertions:
      - Assert:
          'Fn::Contains':
            - - t3.large
            - !Ref InstanceType
        AssertDescription: 'For a production environment, the instance type must be
 t3.large'
```

Parameterübergreifende Validierung

Die folgenden Beispielvorlagen veranschaulichen die Verwendung von Regeln für parameterübergreifende Validierungen. Sie erstellen eine Beispielwebsite, die auf einer Auto Scaling Scaling-Gruppe hinter einem Load Balancer läuft. Die Website ist je nach Eingabeparametern auf Port 80 oder 443 verfügbar. Die Instances in der Auto Scaling Scaling-Gruppe können so konfiguriert werden, dass sie auf jedem Port lauschen (mit 8888 als Standard).

Die Regeln in dieser Vorlage validieren Eingabeparameter vor der Stack-Erstellung. Sie überprüfen, ob alle Subnetze zur angegebenen VPC gehören, und stellen sicher, dass, wenn der `UseSSL` Parameter auf gesetzt ist`Yes`, sowohl ein SSL-Zertifikat-ARN als auch ein Hosting-Zonenname bereitgestellt werden.

> **Note**
>
> Wenn Sie anhand dieser Vorlage einen Stack erstellen, werden Ihnen die verwendeten AWS Ressourcen in Rechnung gestellt.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "VpcId": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "VpcId of your existing Virtual Private Cloud (VPC)",
      "ConstraintDescription": "must be the VPC Id of an existing Virtual Private
 Cloud."
    },
    "Subnets": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "The list of SubnetIds in your Virtual Private Cloud (VPC)",
      "ConstraintDescription": "must be a list of at least two existing subnets
 associated with at least two different availability zones."
    },
    "InstanceType": {
      "Description": "WebServer EC2 instance type",
      "Type": "String",
      "Default": "t2.micro",
      "AllowedValues": ["t2.micro", "t3.micro"],
      "ConstraintDescription": "must be a valid EC2 instance type."
    },
    "KeyName": {
      "Description": "Name of an existing EC2 KeyPair to enable SSH access to the
 instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "SSHLocation": {
      "Description": "The IP address range that can be used to SSH to the EC2
 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
```

```
      "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    },
    "UseSSL": {
      "AllowedValues": ["Yes", "No"],
      "Default": "No",
      "Description": "Select \"Yes\" to implement SSL, \"No\" to skip (default).",
      "Type": "String"
    },
    "ALBSSLCertificateARN": {
      "Default": "",
      "Description": "[Optional] The ARN of the SSL certificate to be used for the
 Application Load Balancer",
      "Type": "String"
    },
    "HostedZoneName": {
      "AllowedPattern": "^$|(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\
\.)*([A-Za-z0-9]|[A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])$",
      "Default": "",
      "Description": "[Optional] The domain name of a valid Hosted Zone on AWS.",
      "Type": "String"
    }
  },
  "Conditions": {
    "UseALBSSL": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]}
  },
  "Rules": {
    "SubnetsInVPC": {
      "Assertions": [
        {
          "Assert": {"Fn::EachMemberEquals": [{"Fn::ValueOf": ["Subnets", "VpcId"]},
 {"Ref": "VpcId"}]},
          "AssertDescription": "All subnets must be in the VPC"
        }
      ]
    },
    "ValidateHostedZone": {
      "RuleCondition": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]},
      "Assertions": [
        {
          "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "ALBSSLCertificateARN"},
 ""]}]},
          "AssertDescription": "ACM Certificate value cannot be empty if SSL is
 required"
```

```
        },
        {
          "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "HostedZoneName"}, ""]}]},
          "AssertDescription": "Route53 Hosted Zone Name is mandatory when SSL is
 required"
        }
      ]
    }
  },
  "Resources": {
    "WebServerGroup": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "Properties": {
        "VPCZoneIdentifier": {"Ref": "Subnets"},
        "LaunchTemplate": {
          "LaunchTemplateId": {"Ref": "LaunchTemplate"},
          "Version": {"Fn::GetAtt": ["LaunchTemplate","LatestVersionNumber"]}
        },
        "MinSize": "2",
        "MaxSize": "2",
        "TargetGroupARNs": [{"Ref": "ALBTargetGroup"}]
      },
      "CreationPolicy": {
        "ResourceSignal": {"Timeout": "PT15M"}
      },
      "UpdatePolicy": {
        "AutoScalingRollingUpdate": {
          "MinInstancesInService": "1",
          "MaxBatchSize": "1",
          "PauseTime": "PT15M",
          "WaitOnResourceSignals": true
        }
      }
    },
    "LaunchTemplate": {
      "Type": "AWS::EC2::LaunchTemplate",
      "Metadata": {
        "Comment": "Install a simple application",
        "AWS::CloudFormation::Init": {
          "config": {
            "packages": {"yum": {"httpd": []}},
            "files": {
              "/var/www/html/index.html": {
```

```
                "content": {"Fn::Join": ["\n", ["<h1>Congratulations, you have
successfully launched the AWS CloudFormation sample.</h1>"]]},
                "mode": "000644",
                "owner": "root",
                "group": "root"
              },
              "/etc/cfn/cfn-hup.conf": {
                "content": {"Fn::Join": ["", [
                  "[main]\n",
                  "stack=", {"Ref": "AWS::StackId"}, "\n",
                  "region=", {"Ref": "AWS::Region"}, "\n"
                ]]},
                "mode": "000400",
                "owner": "root",
                "group": "root"
              },
              "/etc/cfn/hooks.d/cfn-auto-reloader.conf": {
                "content": {"Fn::Join": ["", [
                  "[cfn-auto-reloader-hook]\n",
                  "triggers=post.update\n",
                  "path=Resources.LaunchTemplate.Metadata.AWS::CloudFormation::Init\n",
                  "action=/opt/aws/bin/cfn-init -v ",
                  "         --stack ", {"Ref": "AWS::StackName"},
                  "         --resource LaunchTemplate ",
                  "         --region ", {"Ref": "AWS::Region"}, "\n",
                  "runas=root\n"
                ]]},
                "mode": "000400",
                "owner": "root",
                "group": "root"
              }
            },
            "services": {
              "sysvinit": {
                "httpd": {
                  "enabled": "true",
                  "ensureRunning": "true"
                },
                "cfn-hup": {
                  "enabled": "true",
                  "ensureRunning": "true",
                  "files": [
                    "/etc/cfn/cfn-hup.conf",
                    "/etc/cfn/hooks.d/cfn-auto-reloader.conf"
```

```
                  ]
                }
              }
            }
          }
        }
      },
      "Properties": {
        "LaunchTemplateName": {"Fn::Sub": "${AWS::StackName}-launch-template"},
        "LaunchTemplateData": {
          "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}",
          "SecurityGroupIds": [{"Ref": "InstanceSecurityGroup"}],
          "InstanceType": {"Ref": "InstanceType"},
          "KeyName": {"Ref": "KeyName"},
          "UserData": {
            "Fn::Base64": {"Fn::Join": ["", [
              "#!/bin/bash\n",
              "yum install -y aws-cfn-bootstrap\n",
              "/opt/aws/bin/cfn-init -v ",
              "         --stack ", {"Ref": "AWS::StackName"},
              "         --resource LaunchTemplate ",
              "         --region ", {"Ref": "AWS::Region"}, "\n",
              "/opt/aws/bin/cfn-signal -e $? ",
              "         --stack ", {"Ref": "AWS::StackName"},
              "         --resource WebServerGroup ",
              "         --region ", {"Ref": "AWS::Region"}, "\n"
            ]]}
          }
        }
      }
    },
    "ELBSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Allow access to the ELB",
        "VpcId": {"Ref": "VpcId"},
        "SecurityGroupIngress": [{
          "Fn::If": [
            "UseALBSSL",
            {
              "IpProtocol": "tcp",
              "FromPort": 443,
              "ToPort": 443,
```

```
              "CidrIp": "0.0.0.0/0"
            },
            {
              "IpProtocol": "tcp",
              "FromPort": 80,
              "ToPort": 80,
              "CidrIp": "0.0.0.0/0"
            }
          ]
        }]
      }
    },
    "ApplicationLoadBalancer": {
      "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
      "Properties": {
        "Subnets": {"Ref": "Subnets"},
        "SecurityGroups": [{"Ref": "ELBSecurityGroup"}]
      }
    },
    "ALBListener": {
      "Type": "AWS::ElasticLoadBalancingV2::Listener",
      "Properties": {
        "DefaultActions": [{
          "Type": "forward",
          "TargetGroupArn": {"Ref": "ALBTargetGroup"}
        }],
        "LoadBalancerArn": {"Ref": "ApplicationLoadBalancer"},
        "Port": {"Fn::If": ["UseALBSSL", 443, 80]},
        "Protocol": {"Fn::If": ["UseALBSSL", "HTTPS", "HTTP"]},
        "Certificates": [{
          "Fn::If": [
            "UseALBSSL",
            {"CertificateArn": {"Ref": "ALBSSLCertificateARN"}},
            {"Ref": "AWS::NoValue"}
          ]
        }]
      }
    },
    "ALBTargetGroup": {
      "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
      "Properties": {
        "HealthCheckIntervalSeconds": 30,
        "HealthCheckTimeoutSeconds": 5,
        "HealthyThresholdCount": 3,
```

```
        "Port": 80,
        "Protocol": "HTTP",
        "UnhealthyThresholdCount": 5,
        "VpcId": {"Ref": "VpcId"}
      }
    },
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Enable SSH access and HTTP access on the inbound port",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "SourceSecurityGroupId": {"Fn::Select": [0, {"Fn::GetAtt":
["ApplicationLoadBalancer", "SecurityGroups"]}]}
          },
          {
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIp": {"Ref": "SSHLocation"}
          }
        ],
        "VpcId": {"Ref": "VpcId"}
      }
    },
    "RecordSet": {
      "Type": "AWS::Route53::RecordSetGroup",
      "Condition": "UseALBSSL",
      "Properties": {
        "HostedZoneName": {"Fn::Join": ["", [{"Ref": "HostedZoneName"}, "."]]},
        "RecordSets": [{
          "Name": {"Fn::Join": ["", [
            {"Fn::Select": ["0", {"Fn::Split": [".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"]}]}]},
            ".",
            {"Ref": "HostedZoneName"},
            "."
          ]]},
          "Type": "A",
          "AliasTarget": {
            "DNSName": {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]},
```

```
            "EvaluateTargetHealth": true,
            "HostedZoneId": {"Fn::GetAtt": ["ApplicationLoadBalancer",
 "CanonicalHostedZoneID"]}
         }
       }]
     }
   }
 },
 "Outputs": {
   "URL": {
     "Description": "URL of the website",
     "Value": {"Fn::Join": ["", [
       {"Fn::If": [
         "UseALBSSL",
         {"Fn::Join": ["", [
           "https://",
           {"Fn::Join": ["", [
             {"Fn::Select": ["0", {"Fn::Split": [".", {"Fn::GetAtt":
 ["ApplicationLoadBalancer", "DNSName"]}]}]},
             ".",
             {"Ref": "HostedZoneName"},
             "."
           ]]}
         ]]},
         {"Fn::Join": ["", [
           "http://",
           {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]}
         ]]}
       ]}
     ]]}
   }
 }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VpcId of your existing Virtual Private Cloud (VPC)
    ConstraintDescription: must be the VPC Id of an existing Virtual Private Cloud.
  Subnets:
```

```
    Type: List<AWS::EC2::Subnet::Id>
    Description: The list of SubnetIds in your Virtual Private Cloud (VPC)
    ConstraintDescription: >-
      must be a list of at least two existing subnets associated with at least
      two different availability zones. They should be residing in the selected
      Virtual Private Cloud.
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t3.micro
    ConstraintDescription: must be a valid EC2 instance type.
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  UseSSL:
    AllowedValues:
      - 'Yes'
      - 'No'
    ConstraintDescription: Select Yes to create a HTTPS Listener
    Default: 'No'
    Description: 'Select "Yes" to implement SSL, "No" to skip (default).'
    Type: String
  ALBSSLCertificateARN:
    Default: ''
    Description: >-
      [Optional] The ARN of the SSL certificate to be used for the Application
      Load Balancer
    Type: String
  HostedZoneName:
    AllowedPattern: >-
      ^$|(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9]|[A-Za-
z0-9][A-Za-z0-9\-]*[A-Za-z0-9])$
```

```
    Default: ''
    Description: '[Optional] The domain name of a valid Hosted Zone on AWS.'
    Type: String
Conditions:
  UseALBSSL: !Equals
    - !Ref UseSSL
    - 'Yes'
Rules:
  SubnetsInVPC:
    Assertions:
      - Assert:
          'Fn::EachMemberEquals':
            - 'Fn::ValueOf':
                - Subnets
                - VpcId
            - Ref: VpcId
        AssertDescription: All subnets must be in the VPC
  ValidateHostedZone:
    RuleCondition: !Equals
      - !Ref UseSSL
      - 'Yes'
    Assertions:
      - Assert: !Not
          - !Equals
            - !Ref ALBSSLCertificateARN
            - ''
        AssertDescription: ACM Certificate value cannot be empty if SSL is required
      - Assert: !Not
          - !Equals
            - !Ref HostedZoneName
            - ''
        AssertDescription: Route53 Hosted Zone Name is mandatory when SSL is required
Resources:
  WebServerGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      VPCZoneIdentifier: !Ref Subnets
      LaunchTemplate:
        LaunchTemplateId: !Ref LaunchTemplate
        Version: !GetAtt LaunchTemplate.LatestVersionNumber
      MinSize: '2'
      MaxSize: '2'
      TargetGroupARNs:
        - !Ref ALBTargetGroup
```

```
    CreationPolicy:
      ResourceSignal:
        Timeout: PT15M
    UpdatePolicy:
      AutoScalingRollingUpdate:
        MinInstancesInService: '1'
        MaxBatchSize: '1'
        PauseTime: PT15M
        WaitOnResourceSignals: 'true'
  LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Metadata:
      Comment: Install a simple application
      AWS::CloudFormation::Init:
        config:
          packages:
            yum:
              httpd: []
          files:
            /var/www/html/index.html:
              content: !Join
                - |+
                - - >-
                    <h1>Congratulations, you have successfully launched the AWS
                    CloudFormation sample.</h1>
              mode: '000644'
              owner: root
              group: root
            /etc/cfn/cfn-hup.conf:
              content: !Sub |
                [main]
                stack=${AWS::StackId}
                region=${AWS::Region}
              mode: '000400'
              owner: root
              group: root
            /etc/cfn/hooks.d/cfn-auto-reloader.conf:
              content: !Sub |-
                [cfn-auto-reloader-hook]
                triggers=post.update
                path=Resources.LaunchTemplate.Metadata.AWS::CloudFormation::Init
                action=/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
  LaunchTemplate --region ${AWS::Region}
                runas=root
```

```
              mode: '000400'
              owner: root
              group: root
          services:
            sysvinit:
              httpd:
                enabled: 'true'
                ensureRunning: 'true'
              cfn-hup:
                enabled: 'true'
                ensureRunning: 'true'
                files:
                  - /etc/cfn/cfn-hup.conf
                  - /etc/cfn/hooks.d/cfn-auto-reloader.conf
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
      LaunchTemplateData:
        ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
        SecurityGroupIds:
        - !Ref InstanceSecurityGroup
        InstanceType: !Ref InstanceType
        KeyName: !Ref KeyName
        UserData: !Base64
          Fn::Sub: |
            #!/bin/bash
            yum install -y aws-cfn-bootstrap
            /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
 LaunchTemplate --region ${AWS::Region}
            /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
 WebServerGroup --region ${AWS::Region}
  ELBSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow access to the ELB
      VpcId: !Ref VpcId
      SecurityGroupIngress:
        - !If
          - UseALBSSL
          - IpProtocol: tcp
            FromPort: 443
            ToPort: 443
            CidrIp: 0.0.0.0/0
          - IpProtocol: tcp
```

```
            FromPort: 80
            ToPort: 80
            CidrIp: 0.0.0.0/0
  ApplicationLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Subnets: !Ref Subnets
      SecurityGroups:
        - !Ref ELBSecurityGroup
  ALBListener:
    Type: AWS::ElasticLoadBalancingV2::Listener
    Properties:
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref ALBTargetGroup
      LoadBalancerArn: !Ref ApplicationLoadBalancer
      Port: !If
        - UseALBSSL
        - 443
        - 80
      Protocol: !If
        - UseALBSSL
        - HTTPS
        - HTTP
      Certificates:
        - !If
          - UseALBSSL
          - CertificateArn: !Ref ALBSSLCertificateARN
          - !Ref 'AWS::NoValue'
  ALBTargetGroup:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
      HealthCheckIntervalSeconds: 30
      HealthCheckTimeoutSeconds: 5
      HealthyThresholdCount: 3
      Port: 80
      Protocol: HTTP
      UnhealthyThresholdCount: 5
      VpcId: !Ref VpcId
  InstanceSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable SSH access and HTTP access on the inbound port
      SecurityGroupIngress:
```

```
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          SourceSecurityGroupId: !Select
            - 0
            - !GetAtt
              - ApplicationLoadBalancer
              - SecurityGroups
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref SSHLocation
      VpcId: !Ref VpcId
  RecordSet:
    Type: AWS::Route53::RecordSetGroup
    Condition: UseALBSSL
    Properties:
      HostedZoneName: !Join
        - ''
        - - !Ref HostedZoneName
          - .
      RecordSets:
        - Name: !Join
            - ''
            - - !Select
                - '0'
                - !Split
                  - .
                  - !GetAtt
                    - ApplicationLoadBalancer
                    - DNSName
              - .
              - !Ref HostedZoneName
              - .
          Type: A
          AliasTarget:
            DNSName: !GetAtt
              - ApplicationLoadBalancer
              - DNSName
            EvaluateTargetHealth: true
            HostedZoneId: !GetAtt
              - ApplicationLoadBalancer
              - CanonicalHostedZoneID
Outputs:
```

```
  URL:
    Description: URL of the website
    Value: !Join
      - ''
      - - !If
          - UseALBSSL
          - !Join
            - ''
            - - 'https://'
              - !Join
                - ''
                - - !Select
                    - '0'
                    - !Split
                      - .
                      - !GetAtt
                        - ApplicationLoadBalancer
                        - DNSName
                  - .
                  - !Ref HostedZoneName
                  - .
          - !Join
            - ''
            - - 'http://'
              - !GetAtt
                - ApplicationLoadBalancer
                - DNSName
```

CloudFormation ConditionsVorlagensyntax

Der optionale Abschnitt `Conditions` enthält Anweisungen, die die Bedingungen definieren, unter denen Entitäten erstellt oder konfiguriert werden. Sie können beispielsweise eine Bedingung erstellen und sie einer Ressource oder Ausgabe zuordnen, sodass die Ressource oder Ausgabe nur CloudFormation erstellt wird, wenn die Bedingung erfüllt ist. Ebenso können Sie einer Eigenschaft eine Bedingung zuordnen, sodass die CloudFormation Eigenschaft nur dann auf einen bestimmten Wert gesetzt wird, wenn die Bedingung erfüllt ist. Wenn die Bedingung falsch ist, wird die CloudFormation Eigenschaft auf einen alternativen Wert gesetzt, den Sie angeben.

Sie können Bedingungen verwenden, wenn Sie eine Vorlage wiederverwenden möchten, um Ressourcen in unterschiedlichen Kontexten zu erstellen, z. B. in Test- und Produktionsumgebungen. Sie können Ihrer Vorlage beispielsweise einen `EnvironmentType` Eingabeparameter hinzufügen, der entweder `prod` oder `test` als Eingaben akzeptiert. Für die `prod` Umgebung

könnten Sie EC2 Instanzen mit bestimmten Funktionen einbeziehen, während Sie für die `test` Umgebung möglicherweise reduzierte Funktionen verwenden, um Geld zu sparen. Mit dieser Bedingungsdefinition können Sie für jeden Umgebungstyp definieren, welche Ressourcen erstellt und wie sie konfiguriert werden.

Syntax

Der Abschnitt `Conditions` besteht aus dem Schlüsselnamen `Conditions`. Jede Bedingungsdeklaration umfasst eine logische ID und eine oder mehrere systeminterne Funktionen.

JSON

```
"Conditions": {
  "LogicalConditionName1": {
    "Intrinsic function": ...[
  },

  "LogicalConditionName2": {
    "Intrinsic function": ...
  }
}
```

YAML

```
Conditions:
  LogicalConditionName1:
    Intrinsic function:
      ...

  LogicalConditionName2:
    Intrinsic function:
      ...
```

Wie funktionieren Bedingungen

Gehen Sie wie folgt vor, um Bedingungen zu verwenden:

1. Fügen Sie eine Parameterdefinition hinzu — Definieren Sie im `Parameters` Abschnitt Ihrer Vorlage die Eingaben, die Ihre Bedingungen auswerten werden. Die Bedingungen werden auf der Grundlage dieser Eingabeparameterwerte als wahr oder falsch bewertet. Beachten Sie, dass Pseudo-Parameter automatisch verfügbar sind und keine ausdrückliche Definition in `Parameters`

diesem Abschnitt erfordern. Weitere Informationen zu Pseudoparametern finden Sie unter AWS Werte mithilfe von Pseudo-Parametern abrufen.

2. Fügen Sie eine Bedingungsdefinition hinzu — Definieren Sie Bedingungen im `Conditions` Abschnitt mithilfe systemeigener Funktionen wie oder. `Fn::If` `Fn::Equals` Diese Bedingungen bestimmen, wann die zugehörigen CloudFormation Ressourcen erstellt werden. Die Bedingungen können auf folgenden Faktoren basieren:
   - Eingabe- oder Pseudo-Parameterwerte
   - Andere Bedingungen
   - Werte zuordnen

   Sie können jedoch in Bedingungen nicht auf logische Ressourcen IDs oder deren Attribute verweisen.
3. Zuordnen von Bedingungen zu Ressourcen oder Ausgaben — Referenzieren Sie Bedingungen in Ressourcen oder Ausgaben mithilfe des `Condition` Schlüssels und der logischen ID einer Bedingung. Verwenden Sie `Fn::If` optional auch andere Teile der Vorlage (z. B. Eigenschaftswerte), um Werte auf der Grundlage einer Bedingung festzulegen. Weitere Informationen finden Sie unter Verwenden des `Condition` Schlüssels.

CloudFormation wertet Bedingungen aus, wenn ein Stapel erstellt oder aktualisiert wird. CloudFormation erstellt Entitäten, die mit einer wahren Bedingung verknüpft sind, und ignoriert Entitäten, die einer falschen Bedingung zugeordnet sind. CloudFormation bewertet diese Bedingungen außerdem bei jedem Stack-Update neu, bevor Ressourcen geändert werden. Entitäten, die weiterhin mit einer wahren Bedingung verknüpft sind, werden aktualisiert, während Entitäten, denen eine falsche Bedingung zugewiesen wird, gelöscht werden.

> **Important**
>
> Bedingungen können während einer Stack-Aktualisierung nicht allein aktualisiert werden. Sie können Bedingungen nur aktualisieren, wenn Sie Änderungen einschließen, die Ressourcen hinzufügen, ändern oder löschen.

Intrinsische Bedingungsfunktionen

Sie können die folgenden intrinsischen Funktionen verwenden, um Bedingungen zu definieren:

- Fn::And

- Fn::Equals
- Fn::ForEach
- Fn::If
- Fn::Not
- Fn::Or

> **Note**
>
> `Fn::If` wird nur im Metadatenattribut, Aktualisierungsrichtlinienattribut und Eigenschaftswerte im Bereich `Resources` und den Bereichen `Outputs` einer Vorlage unterstützt.

Verwenden des **`Condition`** Schlüssels

Sobald eine Bedingung definiert ist, können Sie sie mithilfe des `Condition` Schlüssels an verschiedenen Stellen in der Vorlage anwenden`Outputs`, z. B. `Resources` und. Der `Condition` Schlüssel verweist auf den logischen Namen einer Bedingung und gibt das ausgewertete Ergebnis der angegebenen Bedingung zurück.

Themen

- Ordnen Sie Bedingungen Ressourcen zu
- Ordnet Bedingungen Ausgaben zu
- Referenzbedingungen in anderen Bedingungen
- Geben Sie Eigenschaftswerte bedingt zurück mit Fn::If

Ordnen Sie Bedingungen Ressourcen zu

Um Ressourcen bedingt zu erstellen, fügen Sie der Ressource den `Condition` Schlüssel und die logische ID der Bedingung als Attribut hinzu. CloudFormation erstellt die Ressource nur, wenn die Bedingung als wahr ausgewertet wird.

JSON

```
"NewVolume" : {
  "Type" : "AWS::EC2::Volume",
```

```
  "Condition" : "IsProduction",
  "Properties" : {
     "Size" : "100",
     "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone" ]}
  }
}
```

YAML

```
NewVolume:
  Type: AWS::EC2::Volume
  Condition: IsProduction
  Properties:
    Size: 100
    AvailabilityZone: !GetAtt EC2Instance.AvailabilityZone
```

Ordnet Bedingungen Ausgaben zu

Sie können auch Bedingungen mit Ausgaben verknüpfen. CloudFormation erstellt die Ausgabe nur, wenn die zugehörige Bedingung als wahr ausgewertet wird.

JSON

```
"Outputs" : {
  "VolumeId" : {
    "Condition" : "IsProduction",
    "Value" : { "Ref" : "NewVolume" }
  }
}
```

YAML

```
Outputs:
  VolumeId:
    Condition: IsProduction
    Value: !Ref NewVolume
```

Referenzbedingungen in anderen Bedingungen

Bei der Definition von Bedingungen in `Conditions` diesem Abschnitt können Sie mithilfe des `Condition` Schlüssels auf andere Bedingungen verweisen. Auf diese Weise können Sie eine komplexere bedingte Logik erstellen, indem Sie mehrere Bedingungen kombinieren.

Im folgenden Beispiel wird die `IsProdAndFeatureEnabled` Bedingung nur dann als wahr ausgewertet, wenn die `IsFeatureEnabled` Bedingungen `IsProduction` und als wahr ausgewertet werden.

JSON

```
"Conditions": {
  "IsProduction" : {"Fn::Equals" : [{"Ref" : "Environment"}, "prod"]},
  "IsFeatureEnabled" : { "Fn::Equals" : [{"Ref" : "FeatureFlag"}, "enabled"]},
  "IsProdAndFeatureEnabled" : {
    "Fn::And" : [
      {"Condition" : "IsProduction"},
      {"Condition" : "IsFeatureEnabled"}
    ]
  }
}
```

YAML

```
Conditions:
  IsProduction:
    !Equals [!Ref Environment, "prod"]
  IsFeatureEnabled:
    !Equals [!Ref FeatureFlag, "enabled"]
  IsProdAndFeatureEnabled: !And
    - !Condition IsProduction
    - !Condition IsFeatureEnabled
```

Geben Sie Eigenschaftswerte bedingt zurück mit **`Fn::If`**

Für eine detailliertere Steuerung können Sie die `Fn::If` systeminterne Funktion verwenden, um einen von zwei Eigenschaftswerten innerhalb von Ressourcen oder Ausgaben bedingt zurückzugeben. Diese Funktion wertet eine Bedingung aus und gibt einen Wert zurück, wenn die Bedingung wahr ist, und einen anderen Wert, wenn die Bedingung falsch ist.

Bedingte Eigenschaftswerte

Das folgende Beispiel zeigt, wie ein EC2 Instanztyp auf der Grundlage einer Umgebungsbedingung festgelegt wird. Wenn die `IsProduction` Bedingung als wahr ausgewertet wird, wird der Instanztyp auf `c5.xlarge` gesetzt. Andernfalls ist sie auf gesetzt`t3.small`.

JSON

```
"Properties" : {
  "InstanceType" : {
    "Fn::If" : [
      "IsProduction",
      "c5.xlarge",
      "t3.small"
    ]
  }
}
```

YAML

```
Properties:
  InstanceType: !If
    - IsProduction
    - c5.xlarge
    - t3.small
```

Bedingte Entfernung von Eigentum

Sie können den `AWS::NoValue` Pseudo-Parameter auch als Rückgabewert verwenden, um die entsprechende Eigenschaft zu entfernen, wenn eine Bedingung falsch ist.

JSON

```
"DBSnapshotIdentifier" : {
  "Fn::If" : [
    "UseDBSnapshot",
    {"Ref" : "DBSnapshotName"},
    {"Ref" : "AWS::NoValue"}
  ]
}
```

YAML

```
DBSnapshotIdentifier: !If
  - UseDBSnapshot
  - !Ref DBSnapshotName
  - !Ref "AWS::NoValue"
```

Beispiele

Themen

- Umgebungsbasierte Ressourcenerstellung
- Bereitstellung von Ressourcen unter mehreren Bedingungen

Umgebungsbasierte Ressourcenerstellung

In den folgenden Beispielen wird eine EC2 Instanz bereitgestellt und ein neues EBS-Volume nur dann unter bestimmten Bedingungen erstellt und angehängt, wenn der Umgebungstyp `prod` Wenn es sich um eine Umgebung handelt`test`, wird die EC2 Instanz einfach ohne das zusätzliche Volume erstellt.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Parameters": {
        "EnvType": {
            "Description": "Environment type",
            "Default": "test",
            "Type": "String",
            "AllowedValues": [
                "prod",
                "test"
            ],
            "ConstraintDescription": "must specify prod or test"
        }
    },
    "Conditions": {
        "IsProduction": {
            "Fn::Equals": [
                {
                    "Ref": "EnvType"
                },
                "prod"
            ]
        }
    },
    "Resources": {
        "EC2Instance": {
```

```
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId": "ami-1234567890abcdef0",
                "InstanceType": "c5.xlarge"
            }
        },
        "MountPoint": {
            "Type": "AWS::EC2::VolumeAttachment",
            "Condition": "IsProduction",
            "Properties": {
                "InstanceId": {
                    "Ref": "EC2Instance"
                },
                "VolumeId": {
                    "Ref": "NewVolume"
                },
                "Device": "/dev/sdh"
            }
        },
        "NewVolume": {
            "Type": "AWS::EC2::Volume",
            "Condition": "IsProduction",
            "Properties": {
                "Size": 100,
                "AvailabilityZone": {
                    "Fn::GetAtt": [
                        "EC2Instance",
                        "AvailabilityZone"
                    ]
                }
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvType:
    Description: Environment type
    Default: test
    Type: String
```

```
    AllowedValues:
      - prod
      - test
    ConstraintDescription: must specify prod or test
Conditions:
  IsProduction: !Equals
    - !Ref EnvType
    - prod
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-1234567890abcdef0
      InstanceType: c5.xlarge
  MountPoint:
    Type: AWS::EC2::VolumeAttachment
    Condition: IsProduction
    Properties:
      InstanceId: !Ref EC2Instance
      VolumeId: !Ref NewVolume
      Device: /dev/sdh
  NewVolume:
    Type: AWS::EC2::Volume
    Condition: IsProduction
    Properties:
      Size: 100
      AvailabilityZone: !GetAtt
        - EC2Instance
        - AvailabilityZone
```

Bereitstellung von Ressourcen unter mehreren Bedingungen

In den folgenden Beispielen wird unter bestimmten Bedingungen ein S3-Bucket erstellt, wenn ein Bucket-Name angegeben wird, und eine Bucket-Richtlinie wird nur angehängt, wenn die Umgebung auf eingestellt ist. `prod` Wenn kein Bucket-Name angegeben wird oder die Umgebung bereits angegeben ist`test`, werden keine Ressourcen erstellt.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Parameters": {
        "EnvType": {
```

```
            "Type": "String",
            "AllowedValues": [
                "prod",
                "test"
            ]
        },
        "BucketName": {
            "Default": "",
            "Type": "String"
        }
    },
    "Conditions": {
        "IsProduction": {
            "Fn::Equals": [
                {
                    "Ref": "EnvType"
                },
                "prod"
            ]
        },
        "CreateBucket": {
            "Fn::Not": [
                {
                    "Fn::Equals": [
                        {
                            "Ref": "BucketName"
                        },
                        ""
                    ]
                }
            ]
        },
        "CreateBucketPolicy": {
            "Fn::And": [
                {
                    "Condition": "IsProduction"
                },
                {
                    "Condition": "CreateBucket"
                }
            ]
        }
    },
    "Resources": {
```

```
        "Bucket": {
            "Type": "AWS::S3::Bucket",
            "Condition": "CreateBucket",
            "Properties": {
                "BucketName": {
                    "Ref": "BucketName"
                }
            }
        },
        "Policy": {
            "Type": "AWS::S3::BucketPolicy",
            "Condition": "CreateBucketPolicy",
            "Properties": {
                "Bucket": {
                    "Ref": "Bucket"
                },
                "PolicyDocument": { ... }
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvType:
    Type: String
    AllowedValues:
      - prod
      - test
  BucketName:
    Default: ''
    Type: String
Conditions:
  IsProduction: !Equals
    - !Ref EnvType
    - prod
  CreateBucket: !Not
    - !Equals
      - !Ref BucketName
      - ''
  CreateBucketPolicy: !And
```

```
    - !Condition IsProduction
    - !Condition CreateBucket
Resources:
  Bucket:
    Type: AWS::S3::Bucket
    Condition: CreateBucket
    Properties:
      BucketName: !Ref BucketName
  Policy:
    Type: AWS::S3::BucketPolicy
    Condition: CreateBucketPolicy
    Properties:
      Bucket: !Ref Bucket
      PolicyDocument: ...
```

In diesem Beispiel zeigt die `CreateBucketPolicy` Bedingung, wie mithilfe des `Condition` Schlüssels auf andere Bedingungen verwiesen wird. Die Richtlinie wird nur erstellt, wenn `IsProduction` sowohl die `CreateBucket` Bedingungen als auch als wahr bewertet werden.

> **Note**
>
> Komplexere Beispiele für die Verwendung von Bedingungen finden Sie unter dem Thema ConditionAttribute im Referenzhandbuch für AWS CloudFormation Vorlagen.

CloudFormation TransformAbschnitt „Vorlage“

Im optionalen `Transform` Abschnitt werden ein oder mehrere Makros angegeben, die CloudFormation verwendet werden, um Ihre Vorlage auf irgendeine Weise zu verarbeiten.

Makros können einfache Aufgaben wie das Suchen und Ersetzen von Text ausführen, oder sie können umfangreichere Transformationen an der gesamten Vorlage vornehmen. CloudFormation führt Makros in der Reihenfolge aus, in der sie angegeben sind. Wenn Sie einen Änderungssatz erstellen, CloudFormation generiert er einen Änderungssatz, der den verarbeiteten Vorlageninhalt enthält. Sie können dann die Änderungen überprüfen und den Änderungssatz ausführen. Weitere Informationen zur Funktionsweise von Makros finden Sie unterFühren Sie eine benutzerdefinierte Verarbeitung von CloudFormation Vorlagen mit Vorlagenmakros durch.

CloudFormation unterstützt auch Transformationen, bei denen es sich um Makros handelt, die von gehostet werden. CloudFormation CloudFormation behandelt diese Transformationen in Bezug

auf Ausführungsreihenfolge und Umfang genauso wie alle Makros, die Sie erstellen. Weitere Informationen finden Sie unter Transformationsreferenz.

Um mehrere Makros zu deklarieren, verwenden Sie ein Listenformat und geben Sie ein oder mehrere Makros an.

Im nachfolgenden Vorlagenbeispiel werden beispielsweise die Inhalte der gesamten Vorlage CloudFormation ausgewertet `MyMacro` und anschließend verarbeitet`AWS::Serverless`, da sie in den `Transform` Abschnitt aufgenommen wurden.

```
# Start of processable content for MyMacro and AWS::Serverless
Transform:
  - MyMacro
  - 'AWS::Serverless'
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
      Tags: [{"key":"value"}]
      CorsConfiguration: []
  MyEc2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-1234567890abcdef0
# End of processable content for MyMacro and AWS::Serverless
```

CloudFormation Versionssyntax für das Vorlagenformat

Der `AWSTemplateFormatVersion` Abschnitt (optional) identifiziert die Version des Vorlagenformats, der die Vorlage entspricht. Die neueste Vorlagenformatversion ist die Version `2010-09-09`. Dies ist derzeit der einzige gültige Wert.

Die Version im Vorlagenformat entspricht nicht der API-Version. Die Version im Vorlagenformat kann sich unabhängig von den API-Versionen ändern.

Der zur Deklarierung der Vorlagenformatversion verwendete Wert muss eine Literalzeichenfolge sein. Parameter oder Funktionen dürfen nicht zur Angabe der Vorlagenformatversion verwendet werden. Wenn Sie keinen Wert angeben, CloudFormation wird von der neuesten Version des

Vorlagenformats ausgegangen. Der folgende Codeausschnitt ist ein Beispiel für eine gültige Deklaration der Vorlagenformatversion:

JSON

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
```

CloudFormation DescriptionVorlagensyntax

`Description`In diesem Abschnitt (optional) können Sie eine Textzeichenfolge angeben, die die Vorlage beschreibt. Dieser Abschnitt muss immer auf den Abschnitt folgen, der die Vorlagenformatversion festlegt.

Der Wert für die Deklaration der Beschreibung muss eine Literalzeichenfolge mit einer Länge zwischen 0 und 1 024 Byte sein. Parameter oder Funktionen dürfen nicht zur Angabe der Beschreibung verwendet werden. Der folgende Codeausschnitt ist ein Beispiel für die Deklaration einer Beschreibung:

> **Important**
>
> Der Abschnitt `Description` kann während einer Stack-Aktualisierung nicht allein aktualisiert werden. Sie können ihn nur aktualisieren, wenn Sie Änderungen einschließen, die Ressourcen hinzufügen, ändern oder löschen.

JSON

```
"Description" : "Here are some details about the template."
```

YAML

```
Description: > Here are some details about the template.
```

Visuelles Erstellen von Vorlagen mit Infrastructure Composer

AWS-Infrastruktur-Composer (früher bekannt als Application Composer) hilft Ihnen dabei, moderne Anwendungen visuell zu erstellen und zu konfigurieren AWS. Anstatt Code zu schreiben, können Sie verschiedene Ressourcen per Drag-and-Drop verschieben, um Ihre Anwendung visuell zu erstellen.

Infrastructure Composer im CloudFormation Konsolenmodus ist das empfohlene Tool für die visuelle Arbeit mit CloudFormation Vorlagen. Diese Version von Infrastructure Composer, auf die Sie von der CloudFormation Konsole aus zugreifen können, ist eine Verbesserung gegenüber einem älteren Tool namens AWS CloudFormation Designer.

Mit Infrastructure Composer im CloudFormation Konsolenmodus können Sie eine Vielzahl von Ressourcen, so genannten Karten, per Drag-and-Drop auf eine visuelle Leinwand ziehen, dort ablegen, konfigurieren und verbinden. Dieser visuelle Ansatz macht es einfach, Ihre Anwendungsarchitektur zu entwerfen und zu bearbeiten, ohne direkt mit Vorlagen arbeiten zu müssen. Um von der CloudFormation Konsole aus auf diesen Modus zuzugreifen, wählen Sie im Navigationsmenü auf der linken Seite Infrastructure Composer aus.

Weitere Informationen finden Sie im AWS-Infrastruktur-Composer Entwicklerhandbuch unter How to Compose AWS-Infrastruktur-Composer in.

Warum sollte Infrastructure Composer im CloudFormation Konsolenmodus verwendet werden?

Die Visualisierung Ihrer Vorlagen in Infrastructure Composer hilft Ihnen dabei, Lücken und Verbesserungsmöglichkeiten in Ihren CloudFormation Vorlagen und Ihrer Anwendungsarchitektur zu identifizieren. Infrastructure Composer verbessert Ihre Entwicklungserfahrung durch die einfache und effiziente visuelle Erstellung und Änderung von CloudFormation Stacks. Sie können mit einem ersten Entwurf beginnen, bereitstellbaren Code erstellen und Ihre Entwickler-Workflows in den visuellen Designer in Infrastructure Composer integrieren.

Wie unterscheidet sich dieser Modus von der Infrastructure Composer-Konsole?

Die CloudFormation Konsolenversion von Infrastructure Composer hat zwar ähnliche Funktionen wie die Infrastructure Composer-Standardkonsole, es gibt jedoch einige Unterschiede. Lambda-bezogene Karten (Lambda Function und Lambda Layer) erfordern Code-Builds und Paketierungslösungen,

die in Infrastructure Composer im Konsolenmodus nicht verfügbar sind. CloudFormation Die lokale Synchronisierung ist in diesem Modus ebenfalls nicht verfügbar.

Sie können jedoch diese Lambda-bezogenen Karten und die lokale Synchronisierungsfunktion in der Infrastructure Composer-Konsole oder im verwenden. AWS Toolkit for Visual Studio CodeWeitere Informationen finden Sie im AWS-Infrastruktur-Composer Entwicklerhandbuch und in Infrastructure Composer im AWS Toolkit for Visual Studio Code Benutzerhandbuch.

Generieren Sie Vorlagen aus vorhandenen Ressourcen mit dem IaC-Generator

Mit dem CloudFormation Infrastructure-as-Code-Generator (IaC-Generator) können Sie eine Vorlage mithilfe von AWS Ressourcen generieren, die in Ihrem Konto bereitgestellt und noch nicht von verwaltet werden. CloudFormation

Der IaC-Generator bietet folgende Vorteile:

- CloudFormation Verwalte ganze Anwendungen oder migriere sie in eine AWS CDK App.
- Generieren Sie Vorlagen, ohne eine Ressourceneigenschaft nach Eigenschaft beschreiben und diese dann in JSON- oder YAML-Syntax übersetzen zu müssen.
- Verwenden Sie die Vorlage, um Ressourcen in einem neuen Konto oder einer neuen Region zu replizieren.

Der IaC-Generierungsprozess besteht aus den folgenden Schritten:

1. Ressourcen scannen — Der erste Schritt besteht darin, einen Scan Ihrer Ressourcen zu starten. Dieser Scan gilt für die gesamte Region und läuft nach 30 Tagen ab. Während dieser Zeit können Sie aus demselben Scan mehrere Vorlagen erstellen.
2. Erstellen Sie Ihre Vorlage — Um die Vorlage zu erstellen, haben Sie zwei Möglichkeiten:
   - Erstellen Sie eine völlig neue Vorlage und fügen Sie die gescannten Ressourcen und die zugehörigen Ressourcen hinzu.
   - Verwenden Sie einen vorhandenen CloudFormation Stapel als Ausgangspunkt und fügen Sie die gescannten Ressourcen und die zugehörigen Ressourcen zu seiner Vorlage hinzu.
3. Ressourcen importieren — Verwenden Sie Ihre Vorlage, um die Ressourcen als CloudFormation Stapel zu importieren oder sie in eine AWS CDK App zu migrieren.

Die IaC-Generatorfunktion ist in allen Handelsregionen verfügbar und unterstützt viele gängige AWS Ressourcentypen. Eine vollständige Liste der unterstützten Ressourcen finden Sie unterUnterstützung für Ressourcentypen.

Themen

- Überlegungen
- Für das Scannen von Ressourcen sind IAM-Berechtigungen erforderlich
- Häufig verwendete Befehle für die Generierung, Verwaltung und Löschung von Vorlagen
- Migrieren Sie eine Vorlage auf die AWS CDK
- Starten Sie einen Ressourcenscan mit dem CloudFormation IaC-Generator
- Die Scan-Zusammenfassung in der CloudFormation Konsole anzeigen
- Erstellen Sie eine CloudFormation Vorlage aus Ressourcen, die mit dem IaC-Generator gescannt wurden
- Einen CloudFormation Stapel aus gescannten Ressourcen erstellen
- Nur schreibgeschützte Eigenschaften auflösen

Überlegungen

Sie können JSON- oder YAML-Vorlagen für AWS Ressourcen generieren, auf die Sie Lesezugriff haben. Die Vorlagen für die IaC-Generatorfunktion modellieren Cloud-Ressourcen zuverlässig und schnell, ohne dass eine Ressource eigenschaftsweise beschrieben werden muss.

In der folgenden Tabelle sind die Kontingente aufgeführt, die für die IaC-Generierungsfunktion verfügbar sind.

Name	Vollständiger Scan	Teilweiser Scan
Maximale Anzahl von Ressourcen, die in einem Scan verarbeitet werden können	100 000	100 000
Anzahl der Scans pro Tag (für Scans mit weniger als 10.000 Ressourcen)	10	10
Anzahl der Scans pro Tag (für Scans mit mehr als 10.000 Ressourcen)	1	1

Name	Vollständiger Scan	Teilweiser Scan
Gleichzeitige Anzahl von Vorlagen, die pro Konto generiert werden	5	5
Gleichzeitige Anzahl von Ressourcen, die für eine Vorlagengenerierung modelliert wurden	5	5
Gesamtzahl der Ressourcen, die in einer Vorlage modelliert werden können	500	500
Maximale Anzahl generierter Vorlagen pro Konto	1.000	1.000

> **Important**
>
> Der IaC-Generator unterstützt nur AWS Ressourcen, die von der Cloud Control API in Ihrer Region unterstützt werden. Weitere Informationen finden Sie unter Unterstützung für Ressourcentypen.

Für das Scannen von Ressourcen sind IAM-Berechtigungen erforderlich

Um Ressourcen mit dem IaC-Generator zu scannen, muss Ihr IAM-Prinzipal (Benutzer, Rolle oder Gruppe) über Folgendes verfügen:

- CloudFormation Scanberechtigungen
- Leseberechtigungen für AWS Zieldienste

Der Scanbereich ist auf Ressourcen beschränkt, auf die Sie Lesezugriff haben. Fehlende Berechtigungen führen nicht zu einem Scanfehler, sondern schließen diese Ressourcen aus.

Ein Beispiel für eine IAM-Richtlinie, die Berechtigungen zum Scannen und zur Vorlagenverwaltung gewährt, finden Sie unterAlle IaC-Generatoroperationen zulassen.

Häufig verwendete Befehle für die Generierung, Verwaltung und Löschung von Vorlagen

Zu den häufig verwendeten Befehlen für die Arbeit mit dem IaC-Generator gehören:

- start-resource-scanum einen Scan der Ressourcen im Konto in einem AWS-Region zu starten.
- describe-resource-scanum den Fortschritt eines Ressourcenscans zu überwachen.
- list-resource-scansum die Ressourcenscans in einem aufzulisten AWS-Region.
- list-resource-scan-resourcesum die beim Ressourcenscan gefundenen Ressourcen aufzulisten.
- list-resource-scan-related-resources, um die Ressourcen aufzulisten, die sich auf Ihre gescannten Ressourcen beziehen.
- create-generated-templateum eine CloudFormation Vorlage aus einer Reihe gescannter Ressourcen zu generieren.
- update-generated-templateum die generierte Vorlage zu aktualisieren.
- describe-generated-templateum Informationen über eine generierte Vorlage zurückzugeben.
- list-generated-templatesum alle generierten Vorlagen in Ihrem Konto und Ihrer aktuellen Region aufzulisten.
- delete-generated-templateum eine generierte Vorlage zu löschen.

Migrieren Sie eine Vorlage auf die AWS CDK

Das AWS Cloud Development Kit (AWS CDK) ist ein Open-Source-Framework für die Softwareentwicklung, mit dem Sie CloudFormation Ressourcen mithilfe gängiger Programmiersprachen entwickeln, verwalten und bereitstellen können.

Die AWS CDK CLI bietet eine Integration mit dem IaC-Generator. Verwenden Sie den AWS CDK `cdk migrate` CLI-Befehl, um die CloudFormation Vorlage zu konvertieren und eine neue CDK-App zu erstellen, die Ihre Ressourcen enthält. Anschließend können Sie die verwenden, AWS CDK um Ihre Ressourcen zu verwalten und bereitzustellen. CloudFormation

Weitere Informationen finden Sie unter Migrieren zu AWS CDK im AWS Cloud Development Kit (AWS CDK) Entwicklerhandbuch.

Starten Sie einen Ressourcenscan mit dem CloudFormation IaC-Generator

Bevor Sie eine Vorlage aus vorhandenen Ressourcen erstellen, müssen Sie zunächst einen Ressourcenscan starten, um Ihre aktuellen Ressourcen und deren Beziehungen zu ermitteln.

Sie können einen Ressourcenscan mit einer der folgenden Optionen starten. Für Erstbenutzer des IaC-Generators empfehlen wir die erste Option.

- Alle Ressourcen scannen (vollständiger Scan) — Scannt alle vorhandenen Ressourcen im aktuellen Konto und in der Region. Dieser Scanvorgang kann bei 1.000 Ressourcen bis zu 10 Minuten dauern.
- Bestimmte Ressourcen scannen (teilweiser Scan) — Wählen Sie manuell aus, welche Ressourcentypen im aktuellen Konto und in der Region gescannt werden sollen. Diese Option ermöglicht einen schnelleren und gezielteren Scanvorgang und ist somit ideal für die iterative Vorlagenentwicklung.

Nach Abschluss des Scans können Sie auswählen, welche Ressourcen und die zugehörigen Ressourcen bei der Generierung Ihrer Vorlage berücksichtigt werden sollen. Wenn Sie das teilweise Scannen verwenden, sind zugehörige Ressourcen bei der Vorlagengenerierung nur verfügbar, wenn einer der folgenden Fälle zutrifft:

- Sie haben sie ausdrücklich ausgewählt, bevor Sie den Scan gestartet haben, oder
- Sie wurden benötigt, um Ihre ausgewählten Ressourcentypen zu ermitteln.

Wenn Sie beispielsweise auswählen, `AWS::EKS::Nodegroup` ohne sie auszuwählen`AWS::EKS::Cluster`, bezieht der IaC-Generator automatisch `AWS::EKS::Cluster` Ressourcen in den Scan mit ein, da für die Erkennung der Knotengruppe zuerst der Cluster erkannt werden muss. In allen anderen Fällen umfasst der Scan nur die Ressourcen, die Sie speziell ausgewählt haben.

> **Note**
>
> Bevor Sie fortfahren, vergewissern Sie sich, dass Sie über die erforderlichen Berechtigungen für die Arbeit mit dem IaC-Generator verfügen. Weitere Informationen finden Sie unter Für das Scannen von Ressourcen sind IAM-Berechtigungen erforderlich.

Themen

- Starten Sie einen Ressourcenscan (Konsole)
- Starten Sie einen Ressourcenscan (AWS CLI)

Starten Sie einen Ressourcenscan (Konsole)

Um einen Ressourcenscan aller Ressourcentypen zu starten (vollständiger Scan)

1. Öffnen Sie die IaC-Generatorseite der CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, die AWS-Region die zu scannenden Ressourcen enthält.
3. Wählen Sie im Bereich Scans die Option Neuen Scan starten und anschließend Alle Ressourcen scannen aus.

Um einen Ressourcenscan bestimmter Ressourcentypen zu starten (teilweiser Scan)

1. Öffnen Sie die IaC-Generatorseite der CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, die AWS-Region die zu scannenden Ressourcen enthält.
3. Wählen Sie im Bereich Scans die Option Neuen Scan starten und anschließend Bestimmte Ressourcen scannen aus.
4. Wählen Sie im Dialogfeld Teilscan starten bis zu 100 Ressourcentypen aus und wählen Sie dann Scan starten aus.

Starten Sie einen Ressourcenscan (AWS CLI)

Um einen Ressourcenscan aller Ressourcentypen zu starten (vollständiger Scan)

Verwenden Sie den folgenden start-resource-scan-Befehl. *us-east-1*Ersetzen Sie es durch AWS-Region das, das die zu scannenden Ressourcen enthält.

```
aws cloudformation start-resource-scan --region us-east-1
```

Bei Erfolg gibt dieser Befehl den ARN des Scans zurück. Notieren Sie sich den ARN in der `ResourceScanId` Immobilie. Sie benötigen es, um Ihre Vorlage zu erstellen.

```
{
    "ResourceScanId":
      "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-
a3ef-3e6ecfa5da60"
}
```

Um einen Ressourcenscan bestimmter Ressourcentypen zu starten (Teilscan)

1. Verwenden Sie den folgenden cat-Befehl, um die Ressourcentypen, die Sie scannen möchten, in einer JSON-Datei mit dem Namen `config.json` in Ihrem Home-Verzeichnis zu speichern. Im Folgenden finden Sie ein Beispiel für eine Scankonfiguration, die nach EC2 Amazon-Instances, Sicherheitsgruppen und allen Amazon S3-Ressourcen scannt.

   ```
   $ cat > config.json
   [
     {
       "Types":[
         "AWS::EC2::Instance",
         "AWS::EC2::SecurityGroup",
         "AWS::S3::*"
       ]
     }
   ]
   ```

2. Verwenden Sie den start-resource-scanBefehl mit der `--scan-filters` Option zusammen mit der von Ihnen erstellten `config.json` Datei, um den Teilscan zu starten. *us-east-1*Ersetzen Sie es durch AWS-Region das, das die zu scannenden Ressourcen enthält.

   ```
   aws cloudformation start-resource-scan --scan-filters file://config.json --
   region us-east-1
   ```

   Bei Erfolg gibt dieser Befehl den ARN des Scans zurück. Notieren Sie sich den ARN in der `ResourceScanId` Immobilie. Sie benötigen es, um Ihre Vorlage zu erstellen.

   ```
   {
       "ResourceScanId":
         "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-
   a3ef-3e6ecfa5da60"
   }
   ```

Um den Fortschritt eines Ressourcenscans zu überwachen

Verwenden Sie den describe-resource-scan-Befehl. Ersetzen Sie für `--resource-scan-id` diese Option den Beispiel-ARN durch den tatsächlichen ARN.

```
aws cloudformation describe-resource-scan --region us-east-1 \
  --resource-scan-id arn:aws:cloudformation:us-
east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60
```

Ist der Befehl erfolgreich, wird eine Ausgabe zurückgegeben, die wie folgt aussehen sollte:

```
{
    "ResourceScanId": "arn:aws:cloudformation:region:account-
id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60",
    "Status": "COMPLETE",
    "StartTime": "2023-08-21T03:10:38.485000+00:00",
    "EndTime": "2023-08-21T03:20:28.485000+00:00",
    "PercentageCompleted": 100.0,
    "ResourceTypes": [
        "AWS::CloudFront::CachePolicy",
        "AWS::CloudFront::OriginRequestPolicy",
        "AWS::EC2::DHCPOptions",
        "AWS::EC2::InternetGateway",
        "AWS::EC2::KeyPair",
        "AWS::EC2::NetworkAcl",
        "AWS::EC2::NetworkInsightsPath",
        "AWS::EC2::NetworkInterface",
        "AWS::EC2::PlacementGroup",
        "AWS::EC2::Route",
        "AWS::EC2::RouteTable",
        "AWS::EC2::SecurityGroup",
        "AWS::EC2::Subnet",
        "AWS::EC2::SubnetCidrBlock",
        "AWS::EC2::SubnetNetworkAclAssociation",
        "AWS::EC2::SubnetRouteTableAssociation",
        ...
    ],
    "ResourcesRead": 676
}
```

Bei einem Teilscan sieht die Ausgabe etwa wie folgt aus:

```
{
```

```
    "ResourceScanId": "arn:aws:cloudformation:region:account-
id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60",
    "Status": "COMPLETE",
    "StartTime": "2025-03-06T18:24:19.542000+00:00",
    "EndTime": "2025-03-06T18:25:23.142000+00:00",
    "PercentageCompleted": 100.0,
    "ResourceTypes": [
        "AWS::EC2::Instance",
        "AWS::EC2::SecurityGroup",
        "AWS::S3::Bucket",
        "AWS::S3::BucketPolicy"
    ],
    "ResourcesRead": 65,
    "ScanFilters": [
        {
            "Types": [
                "AWS::EC2::Instance",
                "AWS::EC2::SecurityGroup",
                "AWS::S3::*"
            ]
        }
    ]
}
```

Eine Beschreibung der Felder in der Ausgabe finden Sie DescribeResourceScanin der AWS CloudFormation API-Referenz.

Die Scan-Zusammenfassung in der CloudFormation Konsole anzeigen

Nach Abschluss des Scans können Sie sich eine Visualisierung der während des Scans gefundenen Ressourcen anzeigen lassen, anhand derer Sie die Ressourcenkonzentration in den verschiedenen Produkttypen ermitteln können.

Um Informationen zu Ressourcen anzuzeigen, die während des Scans gefunden wurden

1. Öffnen Sie die IaC-Generatorseite der CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die Datei aus, AWS-Region die den anzuzeigenden Ressourcenscan enthält.
3. Wählen Sie im Navigationsbereich die Option IaC-Generator aus.
4. Unter Aufschlüsselung der gescannten Ressourcen finden Sie eine visuelle Aufschlüsselung der gescannten Ressourcen nach Produkttyp, z. B. Computer und Speicher.

5. Um die Anzahl der angezeigten Produkttypen anzupassen, wählen Sie „Angezeigte Daten filtern“. Auf diese Weise können Sie die Visualisierung so anpassen, dass sie sich auf die Produkttypen konzentriert, die Sie am meisten interessieren.
6. Auf der rechten Seite der Seite befindet sich der Bereich mit den Scanzusammenfassungsdetails. Um das Fenster zu öffnen, wählen Sie das Symbol „Bedienfeld öffnen“.

Erstellen Sie eine CloudFormation Vorlage aus Ressourcen, die mit dem IaC-Generator gescannt wurden

In diesem Thema wird erklärt, wie Sie eine Vorlage aus Ressourcen erstellen, die mit der IaC-Generatorfunktion gescannt wurden.

Erstellen Sie eine Vorlage aus gescannten Ressourcen (Konsole)

Um eine Stack-Vorlage aus gescannten Ressourcen zu erstellen

1. Öffnen Sie die IaC-Generatorseite der CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, die AWS-Region die gescannten Ressourcen enthält.
3. Wählen Sie im Bereich Vorlagen die Option Vorlage erstellen aus.
4. Wählen Sie Mit einer neuen Vorlage beginnen aus.
   a. Geben Sie unter Vorlagenname einen Namen für Ihre Vorlage ein.
   b. (Optional) Konfigurieren Sie Ihre Löschrichtlinie und die Richtlinie zum Ersetzen von Updates.
   c. Wählen Sie Weiter, um gescannte Ressourcen zur Vorlage hinzuzufügen.
5. Durchsuchen Sie unter Gescannte Ressourcen hinzufügen die Liste der gescannten Ressourcen und wählen Sie die Ressourcen aus, die Sie zu Ihrer Vorlage hinzufügen möchten. Sie können die Ressourcen nach Ressourcen-ID, Ressourcentyp oder Tags filtern. Die Filter schließen sich gegenseitig ein.
6. Wenn Sie Ihrer Vorlage alle benötigten Ressourcen hinzugefügt haben, klicken Sie auf Weiter, um die Seite Gescannte Ressourcen hinzufügen zu verlassen und mit der Seite Verwandte Ressourcen hinzufügen fortzufahren.

7. Sehen Sie sich eine empfohlene Liste verwandter Ressourcen an. Verwandte Ressourcen, wie EC2 Amazon-Instances und Sicherheitsgruppen, sind voneinander abhängig und gehören in der Regel zu derselben Arbeitslast. Wählen Sie die zugehörigen Ressourcen aus, die Sie in die generierte Vorlage aufnehmen möchten.

> **Note**
>
> Wir empfehlen Ihnen, alle verwandten Ressourcen zu dieser Vorlage hinzuzufügen.

8. Überprüfen Sie die Vorlagendetails, die gescannten Ressourcen und die zugehörigen Ressourcen.
9. Wählen Sie Vorlage erstellen, um die Seite Überprüfen und erstellen zu verlassen und die Vorlage zu erstellen.

Erstellen Sie eine Vorlage aus gescannten Ressourcen (AWS CLI)

Um eine Stapelvorlage aus gescannten Ressourcen zu erstellen

1. Verwenden Sie den list-resource-scan-resourcesBefehl, um die beim Scan gefundenen Ressourcen aufzulisten, und geben Sie optional die `--resource-identifier` Option an, die Ausgabe einzuschränken. Ersetzen Sie für `--resource-scan-id` diese Option den Beispiel-ARN durch den tatsächlichen ARN.

```
aws cloudformation list-resource-scan-resources \
  --resource-scan-id arn:aws:cloudformation:us-
east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60 \
  --resource-identifier MyApp
```

Im Folgenden finden Sie eine Beispielantwort, in der `ManagedByStack` angegeben wird, ob die Ressource bereits CloudFormation verwaltet wird. Kopieren Sie die Ausgabe. Sie benötigen sie für den nächsten Schritt.

```
{
    "Resources": [
        {
            "ResourceType": "AWS::EKS::Cluster",
            "ResourceIdentifier": {
                "ClusterName": "MyAppClusterName"
            },
```

```
            "ManagedByStack": false
        },
        {
            "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
            "ResourceIdentifier": {
                "AutoScalingGroupName": "MyAppASGName"
            },
            "ManagedByStack": false
        }
    ]
}
```

Eine Beschreibung der Felder in der Ausgabe finden Sie ScannedResourcein der AWS CloudFormation API-Referenz.

2. Verwenden Sie den `cat` Befehl, um die Ressourcentypen und Bezeichner in einer JSON-Datei mit dem Namen `resources.json` in Ihrem Home-Verzeichnis zu speichern. Im Folgenden finden Sie ein JSON-Beispiel, das auf der Beispielausgabe im vorherigen Schritt basiert.

```
$ cat > resources.json
[
    {
        "ResourceType": "AWS::EKS::Cluster",
        "ResourceIdentifier": {
            "ClusterName": "MyAppClusterName"
        }
    },
    {
        "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
        "ResourceIdentifier": {
            "AutoScalingGroupName": "MyAppASGName"
        }
    }
]
```

3. Verwenden Sie den Befehl list-resource-scan-related-resources zusammen mit der von Ihnen erstellten `resources.json` Datei, um die Ressourcen aufzulisten, die sich auf Ihre gescannten Ressourcen beziehen.

```
aws cloudformation list-resource-scan-related-resources \
  --resource-scan-id arn:aws:cloudformation:us-
east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60 \
```

```
  --resources file://resources.json
```

Im Folgenden finden Sie ein Antwortbeispiel, das `ManagedByStack` angibt, ob die Ressource bereits CloudFormation verwaltet wird. Fügen Sie diese Ressourcen der JSON-Datei hinzu, die Sie im vorherigen Schritt erstellt haben. Sie benötigen es, um Ihre Vorlage zu erstellen.

```
{
    "RelatedResources": [
        {
            "ResourceType": "AWS::EKS::Nodegroup",
            "ResourceIdentifier": {
                "NodegroupName": "MyAppNodegroupName"
            },
            "ManagedByStack": false
        },
        {
            "ResourceType": "AWS::IAM::Role",
            "ResourceIdentifier": {
                "RoleId": "arn:aws::iam::account-id:role/MyAppIAMRole"
            },
            "ManagedByStack": false
        }
    ]
}
```

Eine Beschreibung der Felder in der Ausgabe finden Sie ScannedResourcein der AWS CloudFormation API-Referenz.

> **Note**
>
> Die Eingabeliste der Ressourcen darf eine Länge von 100 nicht überschreiten. Um verwandte Ressourcen für mehr als 100 Ressourcen aufzulisten, führen Sie den list-resource-scan-related-resources Befehl stapelweise zu 100 aus und konsolidieren Sie die Ergebnisse.
> Beachten Sie, dass die Ausgabe doppelte Ressourcen in der Liste enthalten kann.

4. Verwenden Sie den create-generated-templateBefehl, um wie folgt eine neue Stack-Vorlage mit den folgenden Änderungen zu erstellen:

   - *`us-east-1`*Ersetzen Sie es durch das AWS-Region , das die gescannten Ressourcen enthält.

- *MyTemplate* Ersetzen Sie durch den Namen der zu erstellenden Vorlage.

```
aws cloudformation create-generated-template --region us-east-1 \
 --generated-template-name MyTemplate \
  --resources file://resources.json
```

Im Folgenden sehen Sie ein Beispiel für eine `resources.json`-Datei.

```
[
    {
        "ResourceType": "AWS::EKS::Cluster",
        "LogicalResourceId":"MyCluster",
        "ResourceIdentifier": {
            "ClusterName": "MyAppClusterName"
        }
    },
    {
        "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
        "LogicalResourceId":"MyASG",
        "ResourceIdentifier": {
            "AutoScalingGroupName": "MyAppASGName"
        }
    },
    {
        "ResourceType": "AWS::EKS::Nodegroup",
        "LogicalResourceId":"MyNodegroup",
        "ResourceIdentifier": {
            "NodegroupName": "MyAppNodegroupName"
        }
    },
    {
        "ResourceType": "AWS::IAM::Role",
        "LogicalResourceId":"MyRole",
        "ResourceIdentifier": {
            "RoleId": "arn:aws::iam::account-id:role/MyAppIAMRole"
        }
    }
]
```

Bei Erfolg gibt dieser Befehl Folgendes zurück.

```
{
  "Arn":
    "arn:aws:cloudformation:region:account-id:generatedtemplate/7fc8512c-d8cb-4e02-
b266-d39c48344e48",
  "Name": "MyTemplate"
}
```

Einen CloudFormation Stapel aus gescannten Ressourcen erstellen

Nachdem Sie Ihre Vorlage erstellt haben, können Sie mit Infrastructure Composer eine Vorschau der generierten Vorlage anzeigen, bevor Sie den Stack erstellen und die gescannten Ressourcen importieren. Auf diese Weise können Sie die vollständige Anwendungsarchitektur mit den Ressourcen und ihren Beziehungen visualisieren. Weitere Informationen zu Infrastructure Composer finden Sie unterVisuelles Erstellen von Vorlagen mit Infrastructure Composer.

Um den Stack zu erstellen und die gescannten Ressourcen zu importieren

1. Öffnen Sie die IaC-Generatorseite der CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm das AWS-Region für Ihre Vorlage aus.
3. Wählen Sie den Tab Vorlagen und dann den Namen Ihrer Vorlage, um weitere Informationen anzuzeigen.
4. Auf der Registerkarte Vorlagendefinition oben im Abschnitt Vorlage können Sie die Vorlage je nach Ihren Wünschen von der YAML- zur JSON-Syntax umschalten.
5. Überprüfen Sie die Details Ihrer Vorlage, um sicherzustellen, dass alles korrekt eingerichtet ist. Um das Überprüfen und Verstehen der Vorlage zu erleichtern, können Sie mit Infrastructure Composer von der Standardcodeansicht zu einer grafischen Ansicht der in der Vorlage beschriebenen Infrastruktur wechseln. Wählen Sie dazu unter Vorlage die Option Canvas statt Template aus.

   Canvas-Aktionen

   - Um sich auf die Details einer bestimmten Ressource in Ihrer Vorlage zu konzentrieren, doppelklicken Sie auf eine Karte, um den Bereich mit den Ressourceneigenschaften aufzurufen.

- Um Karten auf der Leinwand visuell anzuordnen und zu organisieren, wählen Sie oben links auf der Leinwand die Option „Anordnen“.
- Verwenden Sie die Zoom-Steuerelemente unten rechts auf der Leinwand, um die Ansicht zu vergrößern oder zu verkleinern.

6. Um eine bestimmte Ressource in der Konsole anzuzeigen, wählen Sie die Registerkarte Vorlagenressourcen und dann die physische ID der Ressource aus, die Sie sich ansehen möchten. Dadurch gelangen Sie zur Konsole für diese spezifische Ressource. Sie können auch Ressourcen in der Vorlagendefinition auf der Registerkarte Vorlagenressourcen hinzufügen, entfernen und erneut synchronisieren.
7. Auf der Registerkarte Vorlagendefinition gibt der IaC-Generator möglicherweise Warnungen vor Ressourcen aus, die schreibgeschützte Eigenschaften enthalten. Nachdem Sie sich die Warnungen angesehen haben, können Sie die generierte Vorlage herunterladen und alle erforderlichen Änderungen vornehmen. Weitere Informationen finden Sie unter Nur schreibgeschützte Eigenschaften auflösen.
8. Wenn Sie mit Ihrer Vorlagendefinition zufrieden sind, wählen Sie auf der Registerkarte Vorlagendefinition die Option In den Stapel importieren und dann Weiter aus.
9. Geben Sie auf der Seite „Stack-Details angeben“ im Bereich „Stapel angeben“ den Namen Ihres Stacks ein, und wählen Sie dann Weiter aus.
10. Überprüfen Sie die Parameter für den Stack und geben Sie sie ein. Wählen Sie Weiter aus.
11. Überprüfen Sie Ihre Optionen auf der Seite Änderungen überprüfen und wählen Sie Weiter.
12. Überprüfe deine Angaben auf der Seite Überprüfen und importieren und wähle Ressourcen importieren aus.

Nur schreibgeschützte Eigenschaften auflösen

Mit dem CloudFormation IaC-Generator können Sie eine Vorlage mithilfe von Ressourcen generieren, die in Ihrem Konto bereitgestellt wurden und die noch nicht von verwaltet wurden. CloudFormation Bestimmte Ressourceneigenschaften sind jedoch als schreibgeschützt gekennzeichnet, was bedeutet, dass sie zwar geschrieben, aber nicht gelesen werden können CloudFormation, z. B. mit einem Datenbankkennwort.

Beim Generieren von CloudFormation Vorlagen aus vorhandenen Ressourcen stellen schreibgeschützte Eigenschaften eine Herausforderung dar. CloudFormation Konvertiert diese Eigenschaften in den meisten Fällen in Parameter in der generierten Vorlage. Auf diese Weise

können Sie die Eigenschaften bei Importvorgängen als Parameterwerte eingeben. Es gibt jedoch Szenarien, in denen diese Konvertierung nicht möglich ist, und CloudFormation behandelt diese Fälle unterschiedlich.

Sich gegenseitig ausschließende Eigenschaften

Einige Ressourcen verfügen über mehrere Gruppen sich gegenseitig ausschließender Eigenschaften, von denen zumindest einige schreibgeschützt sind. In diesen Fällen kann der IaC-Generator nicht ermitteln, welcher Satz exklusiver Eigenschaften bei der Erstellung auf die Ressource angewendet wurde. Sie können beispielsweise den Code für eine bereitstellen, AWS::Lambda::Functionindem Sie einen dieser Eigenschaftensätze verwenden.

- `Code/S3BucketCode/S3Key`, und optional `Code/S3ObjectVersion`
- `Code/ImageUri`
- `Code/ZipFile`

Alle diese Eigenschaften sind schreibgeschützt. Der IaC-Generator wählt einen der exklusiven Eigenschaftssätze aus und fügt ihn der generierten Vorlage hinzu. Für jede der schreibgeschützten Eigenschaften werden Parameter hinzugefügt. Die Parameternamen enthalten `OneOf` und die Parameterbeschreibungen weisen darauf hin, dass die entsprechende Eigenschaft durch andere exklusive Eigenschaften ersetzt werden kann. Der IaC-Generator legt den Warnungstyp `MUTUALLY_EXCLUSIVE_PROPERTIES` für die enthaltenen Eigenschaften fest.

Sich gegenseitig ausschließende Typen

In einigen Fällen kann eine Eigenschaft, die nur über Schreibzugriff verfügt, mehrere Datentypen haben. Beispielsweise AWS::ApiGateway::RestApikann die `Body` Eigenschaft von entweder ein `object` oder ein sein. `string` Wenn dies der Fall ist, nimmt der IaC-Generator die Eigenschaft unter Verwendung des Typs von in die generierte Vorlage auf `string` und legt den Warnungstyp `MUTUALLY_EXCLUSIVE_TYPES` fest.

Array-Eigenschaften

Wenn eine schreibgeschützte Eigenschaft den Typ von hat`array`, kann der IaC-Generator sie nicht in die generierte Vorlage aufnehmen, da Parameter nur Skalarwerte sein können. In diesem Fall wird die Eigenschaft in der Vorlage weggelassen, und es wird der Warnungstyp von festgelegt. `UNSUPPORTED_PROPERTIES`

Optionale Eigenschaften

Bei optionalen, schreibgeschützten Eigenschaften kann der IaC-Generator nicht erkennen, ob die Eigenschaft beim Einrichten der Ressource verwendet wurde. In diesem Fall wird die Eigenschaft in der generierten Vorlage weggelassen und es wird der `UNSUPPORTED_PROPERTIES` Warnungstyp festgelegt.

Warnungen und nächste Schritte

Um festzustellen, welche Eigenschaften schreibgeschützt sind, müssen Sie sich die von der IaC-Generatorkonsole zurückgegebenen Warnungen ansehen. Die Referenz zu AWS Ressourcen- und Eigenschaftstypen gibt nicht an, ob eine Eigenschaft schreibgeschützt ist oder ob sie mehrere Typen unterstützt.

Alternativ können Sie anhand der Schemas des Ressourcenanbieters sehen, welche Eigenschaften schreibgeschützt sind. Informationen zum Herunterladen der Ressourcenanbieter-Schemas finden Sie in den CloudFormation Ressourcenanbieter-Schemas.

Um Probleme mit schreibgeschützten Eigenschaften zu lösen

1. Öffnen Sie die IaC-Generatorseite der Konsole. CloudFormation
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm das AWS-Region für Ihre Vorlage aus.
3. Wählen Sie den Tab Vorlagen und dann den Namen der Vorlage, die Sie erstellt haben.
4. Wenn die generierte Vorlage Ressourcen mit schreibgeschützten Eigenschaften enthält, zeigt die IaC-Generatorkonsole auf der Registerkarte Vorlagendefinition eine Warnung mit einer Zusammenfassung der Art der Probleme an. Zum Beispiel:

5. Wählen Sie Warndetails anzeigen aus, um weitere Informationen zu erhalten. Die Ressourcen mit Schreibschutzeigenschaften werden anhand der logischen ID identifiziert, die in der generierten Vorlage verwendet wird, und des Ressourcentyps.

   Verwenden Sie die Liste der Warnungen, um Ressourcen mit Schreibschutzeigenschaften zu identifizieren, und sehen Sie sich jede Ressource an, um festzustellen, welche Änderungen (falls vorhanden) an der generierten Vorlage vorgenommen werden müssen.

6. Wenn Ihre Vorlage aktualisiert werden muss, um Probleme mit schreibgeschützten Eigenschaften zu beheben, gehen Sie wie folgt vor:

a. Wählen Sie Herunterladen, um eine Kopie der Vorlage herunterzuladen.

b. Bearbeiten Sie Ihre Vorlage.

c. Wenn die Änderungen abgeschlossen sind, können Sie auf die Schaltfläche Bearbeitete Vorlage importieren klicken, um den Importvorgang fortzusetzen.

So lösen Sie Probleme mit schreibgeschützten Eigenschaften in Ressourcen AWS::ApiGateway::RestAPI

In diesem Thema wird erklärt, wie Sie Probleme mit schreibgeschützten Eigenschaften in AWS::ApiGateway::RestApiRessourcen beheben können, wenn Sie den IaC-Generator verwenden.

Problem

Wenn eine generierte Vorlage `AWS::ApiGateway::RestApi` Ressourcen enthält, werden Warnungen generiert, die darauf hinweisen, dass `BodyBodyS3Location`, und `CloneFrom` Eigenschaften als identifiziert werden. `UNSUPPORTED_PROPERTIES` Dies liegt daran, dass es sich um optionale Eigenschaften handelt, die nur zum Schreiben bestimmt sind. Der IaC-Generator weiß nicht, ob diese Eigenschaften jemals auf die Ressource angewendet wurden. Daher werden diese Eigenschaften in der generierten Vorlage weggelassen.

Auflösung

Um die `Body` Eigenschaft für Ihre REST-API festzulegen, aktualisieren Sie Ihre generierte Vorlage.

1. Verwenden Sie die Amazon API Gateway GetExportAPI-Aktion, um die API herunterzuladen. Zum Beispiel mit dem aws apigateway get-export AWS CLI Befehl. Weitere Informationen finden Sie unter Exportieren einer REST-API aus API Gateway im API Gateway Developer Guide.
2. Rufen Sie die `Body` Eigenschaft aus der Antwort auf die `GetExport` API-Aktion ab. Laden Sie es in einen Amazon S3 S3-Bucket hoch.
3. Laden Sie die generierte Vorlage herunter.
4. Fügen Sie der Vorlage die `BodyS3Location/Key` Eigenschaften `BodyS3Location/Bucket` und hinzu und geben Sie den Bucket-Namen und den Schlüssel an, in dem das gespeichert `Body` ist.
5. Öffnen Sie die generierte Vorlage in der IaC-Generatorkonsole und wählen Sie Bearbeitete Vorlage importieren.

So lösen Sie Probleme mit schreibgeschützten Eigenschaften in Ressourcen AWS::Lambda::Function

In diesem Thema wird erklärt, wie Sie Probleme mit schreibgeschützten Eigenschaften in AWS::Lambda::FunctionRessourcen beheben können, wenn Sie den IaC-Generator verwenden.

Problem

Die `AWS::Lambda::Function` Ressource hat drei sich gegenseitig ausschließende Eigenschaftssätze zur Angabe des Lambda-Codes:

- `Code/S3Bucket`und `Code/S3Key` Eigenschaften und optional die `Code/S3ObjectVersion` Eigenschaft
- `Code/ImageUri` Eigenschaft
- `Code/ZipFile` Eigenschaft

Nur einer dieser Sätze kann für eine bestimmte `AWS::Lambda::Function` Ressource verwendet werden.

Der IaC-Generator kann nicht ermitteln, welcher Satz exklusiver schreibgeschützter Eigenschaften zum Erstellen oder Aktualisieren der Ressource verwendet wurde. Daher enthält er nur den ersten Satz von Eigenschaften in der generierten Vorlage. Die `Code/ZipFile` Eigenschaften `Code/ImageUri` und werden weggelassen.

Darüber hinaus gibt der IaC-Generator die folgenden Warnungen aus:

- **`MUTUALLY_EXCLUSIVE_PROPERTIES`**— Warnt davor `Code/S3Bucket` und `Code/S3Key` werden als sich gegenseitig ausschließende Eigenschaften identifiziert.
- **`UNSUPPORTED_PROPERTIES`**— Warnt davor, dass die `Code/S3ObjectVersion` Eigenschaft nicht unterstützt wird.

Um `AWS::Lambda::Function` Ressourcen in eine generierte Vorlage aufzunehmen, müssen Sie die Vorlage herunterladen und mit den richtigen Codeeigenschaften aktualisieren.

Auflösung

Wenn Sie Ihren Lambda-Code in einem Amazon S3 S3-Bucket speichern und die **`S3ObjectVersion`** Eigenschaft nicht verwenden, können Sie die generierte Vorlage ohne

Änderungen importieren. Der IaC-Generator fragt Sie während des Importvorgangs nach dem Amazon S3 S3-Bucket und dem Schlüssel als Vorlagenparameter.

Wenn Sie Ihren Lambda-Code als Amazon ECR-Repository speichern, können Sie Ihre Vorlage mithilfe der folgenden Anweisungen aktualisieren:

1. Laden Sie die generierte Vorlage herunter.
2. Entfernen Sie die Eigenschaften und die entsprechenden Parameter für die `Code/S3Key` Eigenschaften `Code/S3Bucket` und aus der generierten Vorlage.
3. Ersetzen Sie die entfernten Eigenschaften in der generierten Vorlage durch die `Code/ImageUri` Eigenschaft und geben Sie dabei die URL für das Amazon ECR-Repository an.
4. Öffnen Sie die generierte Vorlage in der IaC-Generatorkonsole und wählen Sie die Schaltfläche Bearbeitete Vorlage importieren.

Wenn Sie Ihren Lambda-Code wie in einer Zip-Datei speichern, können Sie Ihre Vorlage mithilfe der folgenden Anweisungen aktualisieren:

1. Laden Sie die generierte Vorlage herunter.
2. Entfernen Sie die Eigenschaften und die entsprechenden Parameter für die `Code/S3Key` Eigenschaften `Code/S3Bucket` und aus der generierten Vorlage.
3. Ersetzen Sie die entfernten Eigenschaften in der generierten Vorlage durch die `Code/ZipFile` Eigenschaft.
4. Öffnen Sie die generierte Vorlage in der IaC-Generatorkonsole und wählen Sie die Schaltfläche Bearbeitete Vorlage importieren.

Wenn Sie keine Kopie Ihres Lambda-Codes haben, können Sie Ihre Vorlage mithilfe der folgenden Anweisungen aktualisieren:

1. Verwenden Sie die AWS Lambda `GetFunction`API-Aktion (z. B. mithilfe des `aws lambda get-function` AWS CLI Befehls).
2. In der Antwort gibt der `RepositoryType` Parameter an, `S3` ob sich der Code in einem Amazon S3 S3-Bucket befindet oder `ECR` ob sich der Code in einem Amazon ECR-Repository befindet.
3. In der Antwort enthält der `Location` Parameter eine vorsignierte URL, mit der Sie das Bereitstellungspaket für 10 Minuten herunterladen können. Laden Sie den Code herunter.
4. Laden Sie den Code in einen Amazon S3 S3-Bucket hoch.

5. Führen Sie einen Importvorgang mit der generierten Vorlage aus und geben Sie den Bucket-Namen und den Schlüssel als Parameterwerte an.

Ruft mithilfe dynamischer Verweise Werte ab, die in anderen Diensten gespeichert sind

Dynamische Verweise bieten Ihnen eine bequeme Möglichkeit, externe Werte anzugeben, die in anderen Diensten gespeichert und verwaltet werden, und vertrauliche Informationen von Ihren Vorlagen zu entkoppeln. infrastructure-as-code CloudFormation ruft den Wert der angegebenen Referenz ab, falls dies bei Stack- und Change-Set-Vorgängen erforderlich ist.

Mit dynamischen Verweisen können Sie:

- Sichere Zeichenfolgen verwenden — Verwenden Sie für vertrauliche Daten immer sichere Zeichenkettenparameter im AWS Systems Manager Parameter Store oder Secrets in AWS Secrets Manager , um sicherzustellen, dass Ihre Daten im Ruhezustand verschlüsselt sind.
- Zugriff einschränken — Beschränken Sie den Zugriff auf die Parameter Store-Parameter oder Secrets Manager Manager-Geheimnisse auf autorisierte Prinzipale und Rollen.
- Zugangsdaten rotieren — Wechseln Sie Ihre sensiblen Daten, die in Parameter Store oder Secrets Manager gespeichert sind, regelmäßig, um ein hohes Maß an Sicherheit zu gewährleisten.
- Automatisieren Sie die Rotation — Nutzen Sie die automatischen Rotationsfunktionen von Secrets Manager, um Ihre sensiblen Daten regelmäßig zu aktualisieren und in Ihren Anwendungen und Umgebungen zu verteilen.

Allgemeine Überlegungen

Die folgenden allgemeinen Überlegungen sollten Sie berücksichtigen, bevor Sie dynamische Verweise in Ihren CloudFormation Vorlagen angeben:

- Vermeiden Sie es, dynamische Verweise oder sensible Daten in Ressourceneigenschaften aufzunehmen, die Teil der primären Kennung einer Ressource sind. CloudFormation verwendet möglicherweise den tatsächlichen Klartext-Wert in der primären Ressourcen-ID, was ein Sicherheitsrisiko darstellen könnte. Diese Ressourcen-ID kann in allen abgeleiteten Ausgaben oder Zielen vorkommen.

Informationen dazu, welche Ressourceneigenschaften die primäre Kennung eines Ressourcentyps enthalten, finden Sie in der Ressourcenreferenzdokumentation für diese Ressource unter Ressourcen - und AWS Eigenschaftstypen-Referenz. Im Abschnitt Return values (Rückgbewerte) stellt der Rückgabewert der `Ref`-Funktion die Ressourceneigenschaften dar, welche die primäre Kennung des Ressourcentyps bilden.

- Sie können bis zu 60 dynamische Referenzen in eine Stack-Vorlage aufnehmen.
- Wenn Sie Transformationen (wie `AWS::Include` oder`AWS::Serverless`) verwenden, löst dynamische Verweise CloudFormation nicht auf, bevor die Transformation angewendet wird. Stattdessen wird die Literalzeichenfolge der dynamischen Referenz an die Transformation übergeben und die Verweise aufgelöst, wenn Sie den Änderungssatz mithilfe der Vorlage ausführen.
- Dynamische Verweise können nicht für sichere Werte (wie die in Parameter Store oder Secrets Manager gespeicherten) in benutzerdefinierten Ressourcen verwendet werden.
- Dynamische Verweise werden auch in `AWS::CloudFormation::Init` Metadaten und EC2 `UserData` Amazon-Eigenschaften nicht unterstützt.
- Erstellen Sie keine dynamische Referenz, die mit einem umgekehrten Schrägstrich (\) endet. CloudFormationkann diese Verweise nicht auflösen, was dazu führen wird, dass Stack-Operationen fehlschlagen.

Die folgenden Themen enthalten Informationen und weitere Überlegungen zur Verwendung dynamischer Verweise.

Themen

- Ruft einen Klartext-Wert aus dem Systems Manager Parameter Store ab
- Rufen Sie einen sicheren Zeichenkettenwert aus dem Systems Manager Parameter Store ab
- Holen Sie sich einen geheimen oder geheimen Wert von Secrets Manager

Ruft einen Klartext-Wert aus dem Systems Manager Parameter Store ab

Wenn Sie eine CloudFormation Vorlage erstellen, möchten Sie möglicherweise Klartextwerte verwenden, die im Parameter Store gespeichert sind. Parameter Store ist eine Funktion von. AWS Systems Manager Eine Einführung in den Parameterspeicher finden Sie unter AWS Systems Manager Parameter Store im AWS Systems Manager Benutzerhandbuch.

Um einen Klartextwert aus dem Parameter Store in Ihrer Vorlage zu verwenden, verwenden Sie eine `ssm` dynamische Referenz. Mit dieser Referenz können Sie auf Werte aus Parametern des Typs `String` oder `StringList` im Parameter Store zugreifen.

Um zu überprüfen, welche Version einer `ssm` dynamischen Referenz in einem Stack-Vorgang verwendet wird, erstellen Sie einen Änderungssatz für den Stack-Vorgang. Überprüfen Sie dann die verarbeitete Vorlage auf der Registerkarte Vorlage. Weitere Informationen finden Sie unter Einen Änderungssatz für einen CloudFormation Stapel erstellen.

Bei der Verwendung `ssm` dynamischer Verweise sind einige wichtige Dinge zu beachten:

- CloudFormation unterstützt keine Drifterkennung bei dynamischen Referenzen. Für `ssm` dynamische Verweise, bei denen Sie die Parameterversion nicht angegeben haben, empfehlen wir, dass Sie, wenn Sie die Parameterversion in Systems Manager aktualisieren, auch einen Stack-Aktualisierungsvorgang für alle Stacks durchführen, die die `ssm` dynamische Referenz enthalten, um die neueste Parameterversion abzurufen.
- Um eine `ssm` dynamische Referenz im `Parameters` Abschnitt Ihrer CloudFormation Vorlage verwenden zu können, müssen Sie eine Versionsnummer angeben. CloudFormation erlaubt es Ihnen nicht, auf einen Parameter Store-Wert ohne Versionsnummer in diesem Abschnitt zu verweisen. Alternativ können Sie Ihren Parameter in Ihrer Vorlage als Systems Manager Manager-Parametertyp definieren. Wenn Sie dies tun, können Sie einen Systems Manager Manager-Parameterschlüssel als Standardwert für Ihren Parameter angeben. CloudFormation ruft dann die neueste Version des Parameterwerts aus dem Parameter Store ab, ohne dass Sie eine Versionsnummer angeben müssen. Dadurch können Ihre Vorlagen einfacher und pflegeleichter werden. Weitere Informationen finden Sie unter Geben Sie vorhandene Ressourcen zur Laufzeit mit von CloudFormation -bereitgestellten Parametertypen an.
- CloudFormation Löst bei benutzerdefinierten Ressourcen die `ssm` dynamischen Verweise auf, bevor die Anfrage an die benutzerdefinierte Ressource gesendet wird.
- CloudFormation unterstützt nicht die Verwendung dynamischer Verweise zum Verweisen auf einen Parameter, der von einem anderen AWS-Konto gemeinsam genutzt wird.
- CloudFormation unterstützt die Verwendung von Systems Manager Manager-Parameterbeschriftungen in dynamischen Verweisen nicht.

Berechtigungen

Um einen im Systems Manager Manager-Parameterspeicher gespeicherten Parameter anzugeben, benötigen Sie die Berechtigung, GetParametersden angegebenen Parameter aufzurufen.

Informationen zum Erstellen von IAM-Richtlinien, die den Zugriff auf bestimmte Systems Manager Manager-Parameter ermöglichen, finden Sie unter Beschränken des Zugriffs auf Systems Manager Manager-Parameter mithilfe von IAM-Richtlinien im AWS Systems Manager Benutzerhandbuch.

Referenzmuster

Verwenden Sie das folgende Referenzmuster, um auf einen Klartextwert zu `ssm` verweisen, der in Ihrer CloudFormation Vorlage im Systems Manager Parameter Store gespeichert ist.

```
{{resolve:ssm:parameter-name:version}}
```

Ihre Referenz muss sich an das folgende reguläre Ausdrucksmuster für Parametername und Version halten:

```
{{resolve:ssm:[a-zA-Z0-9_.\-/]+(:\d+)?}}
```

`parameter-name`

Der Name des Parameters im Parameter Store. Der Parametername unterscheidet Groß- und Kleinschreibung.

Erforderlich

`version`

Eine ganze Zahl, die die Version des zu verwendenden Parameters angibt. Wenn Sie nicht die genaue Version angeben, CloudFormation verwendet immer die neueste Version des Parameters, wenn Sie den Stack erstellen oder aktualisieren. Weitere Informationen finden Sie im AWS Systems Manager Benutzerhandbuch unter Arbeiten mit Parameterversionen.

Optional.

Beispiel

Im folgenden Beispiel wird eine EC2 Startvorlage erstellt, die auf eine benutzerdefinierte AMI-ID verweist, die im Parameter Store gespeichert ist. Die dynamische Referenz ruft die AMI-ID aus der Version *2* des *golden-ami* Parameters jedes Mal ab, wenn eine Instance über die Startvorlage gestartet wird.

JSON

```
{
    "Resources": {
        "MyLaunchTemplate": {
            "Type": "AWS::EC2::LaunchTemplate",
            "Properties": {
                "LaunchTemplateName": {
                    "Fn::Sub": "${AWS::StackName}-launch-template"
                },
                "LaunchTemplateData": {
                    "ImageId": "{{resolve:ssm:golden-ami:2}}",
                    "InstanceType": "t2.micro"
                }
            }
        }
    }
}
```

YAML

```
Resources:
  MyLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
      LaunchTemplateData:
        ImageId: '{{resolve:ssm:golden-ami:2}}'
        InstanceType: t2.micro
```

Rufen Sie einen sicheren Zeichenkettenwert aus dem Systems Manager Parameter Store ab

In können Sie vertrauliche Daten wie Passwörter oder Lizenzschlüssel verwenden CloudFormation, ohne sie direkt in Ihren Vorlagen verfügbar zu machen, indem Sie die vertraulichen Daten als „sichere Zeichenfolge“ im AWS Systems Manager Parameter Store speichern. Eine Einführung in Parameter Store finden Sie unter AWS Systems Manager Parameter Store im AWS Systems Manager Benutzerhandbuch.

Um eine sichere Zeichenfolge für den Parameter Store in Ihrer Vorlage zu verwenden, verwenden Sie eine `ssm-secure` dynamische Referenz. CloudFormation speichert niemals den tatsächlichen Wert

einer sicheren Zeichenfolge. Stattdessen wird nur die wörtliche dynamische Referenz gespeichert, die den Klartext-Parameternamen der sicheren Zeichenfolge enthält.

Greift bei der Stackerstellung oder bei Aktualisierungen CloudFormation nach Bedarf auf den sicheren Zeichenkettenwert zu, ohne den tatsächlichen Wert offenzulegen. Sichere Zeichenketten können nur für Ressourceneigenschaften verwendet werden, die das `ssm-secure` dynamische Referenzmuster unterstützen. Weitere Informationen finden Sie unter Ressourcen, die dynamische Parametermuster für sichere Zeichenfolgen unterstützen.

CloudFormation gibt in keinem API-Aufruf den tatsächlichen Parameterwert für sichere Zeichenketten zurück. Es gibt nur die wörtliche dynamische Referenz zurück. Beim Vergleich von Änderungen mithilfe von Änderungssätzen wird CloudFormation nur die wörtliche dynamische Referenzzeichenfolge verglichen. Die tatsächlichen Werte der sicheren Zeichenfolge werden nicht aufgelöst und verglichen.

Bei der Verwendung `ssm-secure` dynamischer Referenzen sind einige wichtige Dinge zu beachten:

- CloudFormation kann nicht auf Parameter Store-Werte von anderen zugreifen AWS-Konten.
- CloudFormation unterstützt nicht die Verwendung von Systems Manager Manager-Parameterbeschriftungen oder öffentlichen Parametern in dynamischen Verweisen.
- In den `cn-northwest-1` Regionen `cn-north-1` und werden sichere Zeichenketten von Systems Manager nicht unterstützt.
- Dynamische Verweise auf sichere Werte, wie z. B.`ssm-secure`, werden derzeit in benutzerdefinierten Ressourcen nicht unterstützt.
- Wenn ein Stack-Update rückgängig gemacht CloudFormation werden muss und die zuvor angegebene Version eines sicheren Zeichenkettenparameters nicht mehr verfügbar ist, schlägt der Rollback-Vorgang fehl. In solchen Fällen haben Sie zwei Möglichkeiten:
  - Verwenden Sie `CONTINUE_UPDATE_ROLLBACK`, um die Ressource zu überspringen.
  - Erstellen Sie den sicheren String-Parameter im Systems Manager Parameter Store neu und aktualisieren Sie ihn, bis die Parameterversion die in der Vorlage verwendete Version erreicht. Verwenden Sie dann die Ressource, `CONTINUE_UPDATE_ROLLBACK` ohne sie zu überspringen.

Ressourcen, die dynamische Parametermuster für sichere Zeichenfolgen unterstützen

Zu den Ressourcen, die das `ssm-secure` dynamische Referenzmuster unterstützen, gehören:

Ressource	Eigenschaftstyp	Eigenschaften
AWS::DirectoryService::MicrosoftAD		`Password`
AWS::DirectoryService::SimpleAD		`Password`
AWS::ElastiCache::ReplicationGroup		`AuthToken`
AWS::IAM::User	LoginProfile	`Password`
AWS::KinesisFirehose::DeliveryStream	RedshiftDestinationConfiguration	`Password`
AWS::OpsWorks::App	Quelle	`Password`
AWS::OpsWorks::Stack	CustomCookbooksSource	`Password`
AWS::OpsWorks::Stack	RdsDbInstances	`DbPassword`
AWS: :RDS:: DBCluster		`MasterUserPassword`
AWS: :RDS:: DBInstance		`MasterUserPassword`
AWS::Redshift::Cluster		`MasterUserPassword`

Referenzmuster

Verwenden Sie das folgende Referenzmuster, um in Ihrer CloudFormation Vorlage auf einen sicheren Zeichenkettenwert aus dem Systems Manager Parameter Store zu `ssm-secure` verweisen.

```
{{resolve:ssm-secure:parameter-name:version}}
```

Ihre Referenz muss sich an das folgende reguläre Ausdrucksmuster für Parametername und Version halten:

```
{{resolve:ssm-secure:[a-zA-Z0-9_.\-/]+(:\d+)?}}
```

`parameter-name`

Der Name des Parameters im Parameter Store. Der Parametername unterscheidet Groß- und Kleinschreibung.

Erforderlich

`version`

Eine ganze Zahl, die die Version des zu verwendenden Parameters angibt. Wenn Sie nicht die genaue Version angeben, CloudFormation verwendet immer die neueste Version des Parameters, wenn Sie den Stack erstellen oder aktualisieren. Weitere Informationen finden Sie im AWS Systems Manager Benutzerhandbuch unter Arbeiten mit Parameterversionen.

Optional.

Beispiel

Im folgenden Beispiel wird eine `ssm-secure` dynamische Referenz verwendet, um das Passwort für einen IAM-Benutzer auf eine sichere Zeichenfolge festzulegen, die im Parameter Store gespeichert ist. Wie angegeben, CloudFormation wird die Version *`10`* des *`IAMUserPassword`* Parameters für Stack- und Change-Set-Operationen verwendet.

JSON

```
"MyIAMUser": {
  "Type": "AWS::IAM::User",
  "Properties": {
    "UserName": "MyUserName",
    "LoginProfile": {
      "Password": "{{resolve:ssm-secure:IAMUserPassword:10}}"
    }
  }
}
```

YAML

```
MyIAMUser:
  Type: AWS::IAM::User
  Properties:
    UserName: 'MyUserName'
```

```
      LoginProfile:
        Password: '{{resolve:ssm-secure:IAMUserPassword:10}}'
```

Holen Sie sich einen geheimen oder geheimen Wert von Secrets Manager

Secrets Manager ist ein Dienst, mit dem Sie Geheimnisse wie Datenbankanmeldedaten, Passwörter und API-Schlüssel von Drittanbietern sicher speichern und verwalten können. Mit Secrets Manager können Sie den Zugriff auf diese Geheimnisse zentral speichern und steuern, sodass Sie hartcodierte Anmeldeinformationen in Ihrem Code (einschließlich Passwörtern) durch einen API-Aufruf an Secrets Manager ersetzen können, um das Geheimnis programmgesteuert abzurufen. Weitere Informationen finden Sie unter Was ist AWS Secrets Manager? im AWS Secrets Manager Benutzerhandbuch.

Um ganze Geheimnisse oder geheime Werte, die in Secrets Manager gespeichert sind, in Ihren CloudFormation Vorlagen zu verwenden, verwenden Sie `secretsmanager` dynamische Verweise.

Überlegungen

Bei der Verwendung `secretsmanager` dynamischer Verweise sind einige wichtige Sicherheitsaspekte zu beachten:

- Die dynamische `secretsmanager`-Referenz kann in allen Ressourceneigenschaften verwendet werden. Die Verwendung der `secretsmanager` dynamischen Referenz bedeutet, dass weder Secrets Manager noch CloudFormation Protokolle einen aufgelösten geheimen Wert beibehalten sollten. Der Geheimnis-Wert kann jedoch in dem Service erscheinen, in dessen Ressource er verwendet wird. Überprüfen Sie Ihre Nutzung um zu vermeiden, dass Geheimnis-Daten preisgegeben werden.
- Durch das Aktualisieren eines Secrets in Secrets Manager wird das Secret in nicht automatisch aktualisiert CloudFormation. Um eine `secretsmanager` dynamische Referenz CloudFormation zu aktualisieren, müssen Sie ein Stack-Update durchführen, das die Ressource aktualisiert, die die dynamische Referenz enthält, indem Sie entweder die Ressourceneigenschaft aktualisieren, die die `secretsmanager` dynamische Referenz enthält, oder indem Sie eine andere Eigenschaft der Ressource aktualisieren.

  Angenommen, Sie geben in Ihrer Vorlage die Eigenschaft `MasterPassword` einer AWS::RDS::DBInstance-Ressource an, die eine dynamische Referenz `secretsmanager` sein soll, und erstellen dann einen Stack aus der Vorlage. Später aktualisieren Sie den Wert dieses Geheimnisses im Secret Manager, aktualisieren jedoch nicht die Ressource `AWS::RDS::DBInstance` in Ihrer Vorlage. In diesem Fall wird, selbst wenn Sie eine Stack-

Aktualisierung durchführen, der Secret-Wert in der Eigenschaft `MasterPassword` nicht aktualisiert und bleibt der vorherige Secret-Wert.

Darüber hinaus sollten Sie den Secrets Manager verwenden, um das Secret für einen gesicherten Dienst oder eine gesicherte Datenbank automatisch zu rotieren. Weitere Informationen finden Sie unter Drehen von AWS Secrets Manager Geheimnissen.

- Dynamische Verweise für sichere Werte, wie z. B.`secretsmanager`, werden derzeit in benutzerdefinierten Ressourcen nicht unterstützt.

Berechtigungen

Um ein in Secrets Manager gespeichertes Geheimnis anzugeben, müssen Sie berechtigt sein, das Geheimnis abzurufen GetSecretValue.

Referenzmuster

Verwenden Sie das folgende Referenzmuster, um Secrets Manager-Geheimnisse in Ihrer CloudFormation Vorlage zu `secretsmanager` referenzieren.

```
{{resolve:secretsmanager:secret-id:secret-string:json-key:version-stage:version-id}}
```

`secret-id`

Der Name oder ARN des Secrets.

Um auf ein Geheimnis in Ihrem zuzugreifen AWS-Konto, müssen Sie nur den geheimen Namen angeben. Um auf ein Geheimnis in einem anderen zuzugreifen AWS-Konto, geben Sie den vollständigen ARN des Geheimnisses an.

Erforderlich

`secret-string`

Der einzige unterstützte Wert ist `SecretString`. Der Standardwert ist `SecretString`.

`json-key`

Der Schlüsselname des Schlüssel/Wert-Paares, dessen Wert Sie abrufen möchten. Wenn Sie kein a angeben`json-key`, wird der gesamte geheime Text CloudFormation abgerufen.

Dieses Segment darf nicht das Doppelpunktzeichen (`:`) enthalten.

`version-stage`

Das Staging-Label der zu verwendenden Version des Geheimnisses. Secrets Manager verwendet Staging-Markierungen, um während des Rotationsprozesses den Überblick über verschiedene Versionen zu behalten. Wenn Sie `version-stage` verwenden, geben Sie `version-id` nicht an. Wenn Sie weder `version-stage` noch `version-id` angeben, dann ist der Standard die `AWSCURRENT`-Version.

Dieses Segment darf nicht das Doppelpunktzeichen (`:`) enthalten.

`version-id`

Die eindeutige ID der Version des zu verwendenden Secrets. Wenn Sie `version-id` angeben, dürfen Sie `version-stage` nicht angeben. Wenn Sie weder `version-stage` noch `version-id` angeben, dann ist der Standard die `AWSCURRENT`-Version.

Dieses Segment darf nicht das Doppelpunktzeichen (`:`) enthalten.

Beispiele

Themen

- Die Werte für den Benutzernamen und das Passwort werden aus einem Geheimnis abgerufen
- Die gesamte Datei wird abgerufen SecretString
- Abrufen eines Werts aus einer bestimmten Version eines Geheimnisses
- Geheimnisse von einem anderen abrufen AWS-Konto

Die Werte für den Benutzernamen und das Passwort werden aus einem Geheimnis abgerufen

Im folgenden AWS::RDS::DBInstanceBeispiel werden die im Secret gespeicherten Werte für den Benutzernamen und das *`MySecret`* Kennwort abgerufen. Die abgerufene geheime Version ist die Version mit dem Versionsstufenwert von`AWSCURRENT`.

JSON

```
{
    "MyRDSInstance": {
        "Type": "AWS::RDS::DBInstance",
        "Properties": {
```

```
            "DBName": "MyRDSInstance",
            "AllocatedStorage": "20",
            "DBInstanceClass": "db.t2.micro",
            "Engine": "mysql",
            "MasterUsername":
 "{{resolve:secretsmanager:MySecret:SecretString:username}}",
            "MasterUserPassword":
 "{{resolve:secretsmanager:MySecret:SecretString:password}}"
        }
    }
}
```

YAML

```
  MyRDSInstance:
    Type: 'AWS::RDS::DBInstance'
    Properties:
      DBName: MyRDSInstance
      AllocatedStorage: '20'
      DBInstanceClass: db.t2.micro
      Engine: mysql
      MasterUsername: '{{resolve:secretsmanager:MySecret:SecretString:username}}'
      MasterUserPassword: '{{resolve:secretsmanager:MySecret:SecretString:password}}'
```

Die gesamte Datei wird abgerufen SecretString

Die folgende dynamische Referenz ruft das `SecretString` für ab. *MySecret*

```
{{resolve:secretsmanager:MySecret}}
```

Alternative Vorgehensweise:

```
{{resolve:secretsmanager:MySecret::::}}
```

Abrufen eines Werts aus einer bestimmten Version eines Geheimnisses

Die folgende dynamische Referenz ruft den *password* Wert für die *AWSPREVIOUS* Version von ab. *MySecret*

```
{{resolve:secretsmanager:MySecret:SecretString:password:AWSPREVIOUS}}
```

Geheimnisse von einem anderen abrufen AWS-Konto

Die folgende dynamische Referenz ruft das `SecretString` für ab*`MySecret`*, das sich in einem anderen befindet. AWS-Konto Sie müssen den vollständigen geheimen ARN angeben, um auf Geheimnisse in einem anderen zugreifen zu können AWS-Konto.

```
{{resolve:secretsmanager:arn:aws:secretsmanager:us-
west-2:123456789012:secret:MySecret}}
```

Die folgende dynamische Referenz ruft den *`password`* Wert ab, der *`MySecret`* sich in einer anderen AWS-Konto befindet. Sie müssen den vollständigen geheimen ARN angeben, um auf Geheimnisse in einem anderen zugreifen zu können AWS-Konto.

```
{{resolve:secretsmanager:arn:aws:secretsmanager:us-
west-2:123456789012:secret:MySecret:SecretString:password}}
```

AWS Werte mithilfe von Pseudo-Parametern abrufen

Pseudo-Parameter sind integrierte Variablen, die den Zugriff auf wichtige AWS Umgebungsinformationen wie Konten IDs, Regionsnamen und Stack-Details ermöglichen, die sich zwischen Bereitstellungen oder Umgebungen ändern können.

Sie können Pseudo-Parameter anstelle von fest codierten Werten verwenden, um Ihre Vorlagen portabler zu machen und sie in verschiedenen Regionen leichter wiederzuverwenden. AWS-Konten

Syntax

Sie können auf Pseudo-Parameter verweisen, indem Sie entweder die `Ref` intrinsische Funktion oder die intrinsische Funktion verwenden. `Fn::Sub`

Punkt

Die `Ref` systeminterne Funktion verwendet die folgende allgemeine Syntax. Weitere Informationen finden Sie unter Ref.

JSON

```
{ "Ref" : "AWS::PseudoParameter" }
```

YAML

```
!Ref AWS::PseudoParameter
```

Fn: :Sub

Die `Fn::Sub` systeminterne Funktion verwendet ein anderes Format, das die `${}` Syntax rund um den Pseudo-Parameter beinhaltet. Weitere Informationen finden Sie unter Fn::Sub.

JSON

```
{ "Fn::Sub" : "${AWS::PseudoParameter}" }
```

YAML

```
!Sub '${AWS::PseudoParameter}'
```

Verfügbare Pseudoparameter

`AWS::AccountId`

Gibt die AWS-Konto ID des Kontos zurück, in dem der Stack erstellt wird, z. B. `123456789012`

Dieser Pseudo-Parameter wird häufig bei der Definition von IAM-Rollen, Richtlinien und anderen Ressourcenrichtlinien verwendet, die kontospezifische Richtlinien beinhalten. ARNs

`AWS::NotificationARNs`

Gibt die Liste der Amazon-Ressourcennamen (ARNs) für die Amazon SNS-Themen zurück, die Stack-Ereignisbenachrichtigungen erhalten. Sie können diese ARNs über die `--notification-arns` Option in der AWS CLI oder über die Konsole angeben, während Sie Ihren Stack erstellen oder aktualisieren.

Im Gegensatz zu anderen Pseudo-Parametern, die einen einzelnen Wert zurückgeben, `AWS::NotificationARNs` gibt dies eine Liste von ARNs zurück. Verwenden Sie die `Fn::Select` systeminterne Funktion, um auf einen bestimmten ARN in der Liste zuzugreifen. Weitere Informationen finden Sie unter Fn::Select.

AWS::NoValue

Entfernt die entsprechende Ressourceneigenschaft, wenn als Rückgabewert in der intrinsischen Funktion `Fn::If` angegeben. Weitere Informationen finden Sie unter Fn::If.

Dieser Pseudo-Parameter ist besonders nützlich, um bedingte Ressourceneigenschaften zu erstellen, die nur unter bestimmten Bedingungen enthalten sein sollten.

AWS::Partition

Gibt die Partition zurück, in der sich die Ressource befindet. Standardmäßig ist AWS-Regionen die Partition`aws`. Für Ressourcen in anderen Partitionen lautet die Partition `aws-`*`partitionname`*. Zum Beispiel ist die Partition für Ressourcen in den Regionen China (Peking und Ningxia) `aws-cn` und die Partition für Ressourcen in der Region AWS GovCloud (USA West). `aws-us-gov`

Die Partition ist Teil des ARN für Ressourcen. Durch die Verwendung `AWS::Partition` wird sichergestellt, dass Ihre Vorlagen auf verschiedenen AWS Partitionen korrekt funktionieren.

AWS::Region

Gibt eine Zeichenfolge zurück, die die Region repräsentiert, in der die umfassende Ressource erstellt wird, z. B. `us-west-2`.

Dies ist einer der am häufigsten verwendeten Pseudo-Parameter, da er es Vorlagen ermöglicht, sich AWS-Regionen ohne Änderung an andere anzupassen.

AWS::StackId

Gibt die ID (ARN) des Stacks zurück, z. `arn:aws:cloudformation:us-west-2:123456789012:stack/teststack/51af3dc0-da77-11e4-872e-1234567db123` B.

AWS::StackName

Gibt den Namen des Stacks zurück, z. `teststack` B.

Der Stackname wird häufig verwendet, um eindeutige Ressourcennamen zu erstellen, die leicht als zu einem bestimmten Stack gehörend identifiziert werden können.

AWS::URLSuffix

Gibt das Suffix für die AWS Domain zurück, AWS-Region in der der Stack bereitgestellt wird. Das Suffix ist typischamazonaws.com, aber für die Region China (Peking) ist das Suffix. amazonaws.com.cn

Dieser Parameter ist besonders nützlich bei der Konstruktion URLs für AWS Dienstendpunkte.

Beispiele

Themen

- Grundlegende Verwendung
- Verwenden von AWS::NotificationARNs
- Bedingte Eigenschaften mit AWS::NoValue

Grundlegende Verwendung

In den folgenden Beispielen werden zwei Ressourcen erstellt: ein Amazon SNS SNS-Thema und ein CloudWatch Alarm, der Benachrichtigungen zu diesem Thema sendet. Sie verwendenAWS::StackName, undAWS::Region, AWS::AccountId um den Stacknamen, die aktuelle AWS-Region ID und die Konto-ID dynamisch in Ressourcennamen, Beschreibungen und ARNs einzufügen.

JSON

```
{
    "Resources": {
        "MyNotificationTopic": {
            "Type": "AWS::SNS::Topic",
            "Properties": {
                "DisplayName": { "Fn::Sub": "Notifications for ${AWS::StackName}" }
            }
        },
        "CPUAlarm": {
            "Type": "AWS::CloudWatch::Alarm",
            "Properties": {
                "AlarmDescription": { "Fn::Sub": "Alarm for high CPU in
 ${AWS::Region}" },
                "AlarmName": { "Fn::Sub": "${AWS::StackName}-HighCPUAlarm" },
                "MetricName": "CPUUtilization",
```

```
                "Namespace": "AWS/EC2",
                "Statistic": "Average",
                "Period": 300,
                "EvaluationPeriods": 1,
                "Threshold": 80,
                "ComparisonOperator": "GreaterThanThreshold",
                "AlarmActions": [{ "Fn::Sub": "arn:aws:sns:${AWS::Region}:
${AWS::AccountId}:${MyNotificationTopic}" }]
            }
        }
    }
}
```

YAML

```
Resources:
  MyNotificationTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: !Sub Notifications for ${AWS::StackName}
  CPUAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: !Sub Alarm for high CPU in ${AWS::Region}
      AlarmName: !Sub ${AWS::StackName}-HighCPUAlarm
      MetricName: CPUUtilization
      Namespace: AWS/EC2
      Statistic: Average
      Period: 300
      EvaluationPeriods: 1
      Threshold: 80
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - !Sub arn:aws:sns:${AWS::Region}:${AWS::AccountId}:${MyNotificationTopic}
```

Verwenden von AWS::NotificationARNs

In den folgenden Beispielen wird eine Auto Scaling Scaling-Gruppe so konfiguriert, dass sie Benachrichtigungen für Instance-Startereignisse und Startfehler sendet. Die Konfiguration verwendet den `AWS::NotificationARNs` Pseudo-Parameter, der eine Liste der Amazon SNS-Themen bereitstellt ARNs , die bei der Stack-Erstellung angegeben wurden. Die `Fn::Select` Funktion wählt den ersten ARN aus dieser Liste aus.

JSON

```
"myASG": {
   "Type": "AWS::AutoScaling::AutoScalingGroup",
   "Properties": {
      "LaunchTemplate": {
         "LaunchTemplateId": { "Ref": "myLaunchTemplate" },
         "Version": { "Fn::GetAtt": [ "myLaunchTemplate", "LatestVersionNumber" ] }
       },
       "MaxSize": "1",
       "MinSize": "1",
       "VPCZoneIdentifier": [
          "subnetIdAz1",
          "subnetIdAz2",
          "subnetIdAz3"
      ],
      "NotificationConfigurations" : [{
         "TopicARN" : { "Fn::Select" : [ "0", { "Ref" : "AWS::NotificationARNs" } ] },
         "NotificationTypes" : [ "autoscaling:EC2_INSTANCE_LAUNCH",
 "autoscaling:EC2_INSTANCE_LAUNCH_ERROR" ]
      }]
   }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MinSize: '1'
    MaxSize: '1'
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    NotificationConfigurations:
      - TopicARN:
          Fn::Select:
          - '0'
          - Ref: AWS::NotificationARNs
```

```
        NotificationTypes:
        - autoscaling:EC2_INSTANCE_LAUNCH
        - autoscaling:EC2_INSTANCE_LAUNCH_ERROR
```

Bedingte Eigenschaften mit AWS::NoValue

In den folgenden Beispielen wird eine Amazon RDS-DB-Instance erstellt, die nur dann einen Snapshot verwendet, wenn eine Snapshot-ID angegeben wird. Wenn die `UseDBSnapshot` Bedingung als wahr ausgewertet wird, CloudFormation wird der `DBSnapshotName` Parameterwert für die `DBSnapshotIdentifier` Eigenschaft verwendet. Wenn die Bedingung als falsch ausgewertet wird, CloudFormation wird die `DBSnapshotIdentifier` Eigenschaft entfernt.

JSON

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "MySQL",
    "EngineVersion" : "5.5",
    "MasterUsername" : { "Ref" : "DBUser" },
    "MasterUserPassword" : { "Ref" : "DBPassword" },
    "DBParameterGroupName" : { "Ref" : "MyRDSParamGroup" },
    "DBSnapshotIdentifier" : {
      "Fn::If" : [
        "UseDBSnapshot",
        {"Ref" : "DBSnapshotName"},
        {"Ref" : "AWS::NoValue"}
      ]
    }
  }
}
```

YAML

```
MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: MySQL
```

```
    EngineVersion: '5.5'
    MasterUsername:
      Ref: DBUser
    MasterUserPassword:
      Ref: DBPassword
    DBParameterGroupName:
      Ref: MyRDSParamGroup
    DBSnapshotIdentifier:
      Fn::If:
        - UseDBSnapshot
        - Ref: DBSnapshotName
        - Ref: AWS::NoValue
```

Holen Sie sich exportierte Ausgaben aus einem bereitgestellten CloudFormation Stack

Wenn Sie mehrere Stapel in derselben AWS-Konto Region haben, möchten Sie möglicherweise Informationen zwischen ihnen austauschen. Dies ist nützlich, wenn ein Stapel Ressourcen verwenden muss, die von einem anderen Stapel erstellt wurden.

Beispielsweise könnten Sie über einen Stack verfügen, der Netzwerkressourcen wie Subnetze und Sicherheitsgruppen für Ihre Webserver erstellt. Andere Stapel, die die eigentlichen Webserver bilden, können dann die Netzwerkressourcen verwenden, die durch den ersten Stapel erstellt wurden. Sie müssen die Ressourcen nicht IDs in der Vorlage des Stacks fest codieren oder IDs als Eingabeparameter übergeben.

Um Informationen zwischen Stapeln auszutauschen, exportieren Sie Ausgabewerte aus einem Stapel und importieren sie in einen anderen Stapel. Funktionsweise:

1. In der Vorlage des ersten Stacks (z. B. dem Netzwerk-Stack) definieren Sie mithilfe des `Export` Felds im `Outputs` Abschnitt bestimmte Werte für den Export. Weitere Informationen finden Sie unter CloudFormation OutputsVorlagensyntax.
2. Wenn Sie diesen Stack erstellen oder aktualisieren, werden die CloudFormation Ausgabewerte exportiert AWS-Konto und sie anderen Stacks in derselben Region zur Verfügung gestellt.
3. In der Vorlage des anderen Stacks verwenden Sie die Fn::ImportValueFunktion, um die exportierten Werte aus dem ersten Stapel zu importieren.
4. Wenn Sie den zweiten Stack (z. B. den Webserver-Stack) erstellen oder aktualisieren, ruft er CloudFormation automatisch die exportierten Werte aus dem ersten Stack ab und verwendet sie.

Eine Anleitung sowie Beispielvorlagen finden Sie unter Verweisen Sie auf Ressourcenausgaben in einem anderen CloudFormation Stack.

Exportieren von Stack-Ausgabewerten im Vergleich zur Verwendung geschachtelter Stacks

Ein geschachtelter Stack ist ein Stack, den Sie innerhalb eines anderes Stacks mithilfe der `AWS::CloudFormation::Stack`-Ressource erstellen können. Bei geschachtelten Stacks können Sie alle Ressourcen aus einem einzigen Stack bereitstellen und verwalten. Sie können Ausgaben aus einem Stack in den geschachtelten Stack-Gruppen als Eingaben für einen anderen Stack in der Gruppe verwenden. Dies unterscheidet sich vom Exportieren von Werten.

Wenn Sie den Informationsaustausch auf innerhalb einer geschachtelten Stack-Gruppe isolieren möchten, wird empfohlen, geschachtelte Stacks zu verwenden. Um Informationen mit anderen Stacks (nicht nur innerhalb der Gruppe geschachtelter Stacks) auszutauschen, exportieren Sie Werte. Sie können z. B. einen einzelnen Stack mit einem Subnetz erstellen und dann seine ID exportieren. Andere Stacks können dieses Subnetz verwenden, indem sie seine ID importieren. Jeder Stack muss kein eigenes Subnetz erstellen. Solange Stacks die Subnetz-ID importieren, können sie nicht geändert oder gelöscht werden.

Weitere Informationen zu verschachtelten Stacks finden Sie unter. Teilen Sie eine Vorlage mithilfe verschachtelter Stapel in wiederverwendbare Teile auf

Überlegungen

Die folgenden Einschränkungen gelten für Cross-Stack-Referenzen:

- Jeder AWS-Konto`Export` Name muss innerhalb einer Region eindeutig sein.
- Cross-Stack-Referenzen können nicht regionsübergreifend erstellt werden. Sie können die intrinsische Funktion `Fn::ImportValue` verwenden, um nur Werte zu importieren, die innerhalb derselben Region exportiert wurden.
- Für Ausgaben kann der Wert der `Name`-Eigenschaft eines `Export` keine `Ref`- oder `GetAtt`-Funktionen verwenden, die von einer Ressource abhängen.

  Ebenso darf die `ImportValue`-Funktion keine `Ref`- oder `GetAtt`-Funktionen enthalten, die von einer Ressource abhängen.
- Nachdem ein anderer Stack einen Ausgabewert importiert hat, können Sie den Stack, der den Ausgabewert exportiert, nicht löschen oder den exportierten Ausgabewert ändern. Alle Importe

müssen entfernt werden, bevor Sie den exportierenden Stack löschen oder den Ausgabewert ändern können.

Auflistung von exportierten Ausgabewerten

Wenn Sie die exportierten Ausgabewerte aus Ihren Stacks anzeigen möchten, verwenden Sie eine der folgenden Methoden:

So listen Sie exportierte Ausgabewerte auf (Konsole)

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie im linken Navigationsbereich Exporte aus.

So listen Sie exportierte Ausgabewerte auf (AWS CLI)

Verwenden Sie den folgenden list-exports-Befehl. Ersetze es *us-east-1* durch dein AWS-Region.

```
aws cloudformation list-exports --region us-east-1
```

Es folgt eine Beispielausgabe.

```
{
    "Exports": [
        {
            "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
            "Name": "private-vpc-subnet-a",
            "Value": "subnet-07b410xmplddcfa03"
        },
        {
            "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
            "Name": "private-vpc-subnet-b",
            "Value": "subnet-075ed3xmplebd2fb1"
        },
        {
            "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
```

```
            "Name": "private-vpc-vpcid",
            "Value": "vpc-011d7xmpl100e9841"
        }
    ]
}
```

CloudFormation zeigt die Namen und Werte der exportierten Ausgaben für die aktuelle Region und den Stack, aus dem sie exportiert wurden. Um einen exportierten Ausgabewert in der Vorlage eines anderen Stacks zu verwenden, können Sie ihn mit dem Exportnamen und der `Fn::ImportValue` Funktion referenzieren.

Auflistung von Stacks, die einen exportierten Ausgabewert importieren

Um exportierte Ausgabewerte zu löschen oder zu ändern, müssen Sie zunächst herausfinden, welche Stapel sie importieren.

Verwenden Sie eine der folgenden Methoden, um die Stapel anzuzeigen, die einen exportierten Ausgabewert importieren:

Auflisten von Stacks, die einen exportierten Ausgabewert importieren (Konsole)

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie im linken Navigationsbereich Exports aus.
3. Um zu sehen, welche Stapel einen bestimmten Exportwert importieren, wählen Sie den Exportnamen für diesen Exportwert aus. CloudFormation zeigt die Seite mit den Exportdetails an, auf der alle Stapel aufgeführt sind, die den Wert importieren.

Auflisten von Stacks, die einen exportierten Ausgabewert importieren (AWS CLI)

Verwenden Sie den list-imports-Befehl. *us-east-1*Ersetzen Sie es durch Ihren AWS-Region und *private-vpc-vpcid* durch den Namen des exportierten Ausgabewerts.

```
aws cloudformation list-imports --region us-east-1 \
    --export-name private-vpc-vpcid
```

CloudFormation gibt eine Liste von Stacks zurück, die den Wert importieren.

```
{
```

```
    "Imports": [
        "my-app-stack"
    ]
}
```

Sobald Sie wissen, welche Stapel einen bestimmten exportierten Wert importieren, müssen Sie diese Stapel ändern, um die `Fn::ImportValue` Funktionen zu entfernen, die auf die Ausgabewerte verweisen. Sie müssen alle Importe entfernen, die auf exportierte Ausgabewerte verweisen, bevor Sie die exportierten Ausgabewerte löschen oder ändern können.

Geben Sie vorhandene Ressourcen zur Laufzeit mit von CloudFormation -bereitgestellten Parametertypen an

Beim Erstellen Ihrer Vorlage können Sie Parameter erstellen, bei denen Benutzer Kennungen vorhandener AWS Ressourcen oder Systems Manager Manager-Parameter eingeben müssen, indem Sie spezielle Parametertypen verwenden, die von CloudFormation bereitgestellt werden.

Themen

- Übersicht
- Beispiel
- Überlegungen
- Unterstützte AWS-spezifische Parametertypen
- Unterstützte Systems Manager Manager-Parametertypen
- Nicht unterstützte Systems Manager Manager-Parametertypen

Übersicht

In können Sie Parameter verwenden CloudFormation, um Ihre Stapel anzupassen, indem Sie bei der Erstellung oder Aktualisierung des Stacks Eingabewerte angeben. Diese Funktion macht Ihre Vorlagen für verschiedene Szenarien wiederverwendbar und flexibel.

Parameter werden im `Parameters` Abschnitt einer CloudFormation Vorlage definiert. Jeder Parameter hat einen Namen und einen Typ und kann zusätzliche Einstellungen wie einen Standardwert und zulässige Werte haben. Weitere Informationen finden Sie unter CloudFormation ParametersVorlagensyntax.

Der Parametertyp bestimmt, welche Art von Eingabewert der Parameter akzeptieren kann. Akzeptiert beispielsweise `Number` nur numerische Werte, `String` akzeptiert aber Texteingaben.

CloudFormation stellt mehrere zusätzliche Parametertypen bereit, die Sie in Ihren Vorlagen verwenden können, um auf vorhandene AWS Ressourcen und Systems Manager Manager-Parameter zu verweisen.

Diese Parametertypen lassen sich in zwei Kategorien einteilen:

- AWS-spezifische Parametertypen — CloudFormation stellt eine Reihe von Parametertypen bereit, mit deren Hilfe ungültige Werte beim Erstellen oder Aktualisieren eines Stacks erkannt werden können. Wenn Sie diese Parametertypen verwenden, muss jeder, der Ihre Vorlage verwendet, gültige Werte aus der AWS-Konto Region angeben, in der der Stack erstellt wird.

  Wenn sie den verwenden AWS Management Console, CloudFormation wird eine vorausgefüllte Liste mit vorhandenen Werten aus ihrem Konto und ihrer Region bereitgestellt. Auf diese Weise muss sich der Benutzer nicht an einen bestimmten Namen oder eine ID erinnern und diese korrekt eingeben. Stattdessen wählen sie einfach Werte aus einer Drop-down-Liste aus. In einigen Fällen können sie sogar anhand von ID, Name oder `Name` Tag-Wert nach Werten suchen.
- Systems Manager-Parametertypen — stellt CloudFormation auch Parametertypen bereit, die vorhandenen Parametern im Systems Manager Parameter Store entsprechen. Wenn Sie diese Parametertypen verwenden, muss jeder, der Ihre Vorlage verwendet, einen Parameter Store-Schlüssel als Wert für den Systems Manager Manager-Parametertyp angeben und CloudFormation dann den neuesten Wert aus dem Parameter Store abrufen, um ihn in seinem Stack zu verwenden. Dies kann nützlich sein, wenn Sie Anwendungen häufig mit neuen Eigenschaftswerten aktualisieren müssen, z. B. mit einem neuen Amazon Machine Image (AMI) IDs. Informationen zum Parameterspeicher finden Sie unter Systems Manager Parameter Store.

Sobald Ihre Parameter in `Parameters` diesem Abschnitt definiert sind, können Sie mithilfe der `Ref` Funktion in der gesamten CloudFormation Vorlage auf Parameterwerte verweisen.

Beispiel

Das folgende Beispiel zeigt eine Vorlage, die die folgenden Parametertypen verwendet.

- `AWS::EC2::VPC::Id`
- `AWS::EC2::Subnet::Id`
- `AWS::EC2::KeyPair::KeyName`

- `AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>`

Um einen Stack aus dieser Vorlage zu erstellen, müssen Sie eine bestehende VPC-ID, eine Subnetz-ID und einen Schlüsselpaarnamen aus Ihrem Konto angeben. Sie können auch einen vorhandenen Parameter Store-Schlüssel angeben, der auf die gewünschte AMI-ID verweist, oder den Standardwert von beibehalten*/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2*. Dieser öffentliche Parameter ist ein Alias für die regionale AMI-ID für das neueste Amazon Linux 2-AMI. Weitere Informationen zu öffentlichen Parametern finden Sie unter Discovering public parameters in Parameter Store im AWS Systems Manager Benutzerhandbuch.

JSON

```
{
    "Parameters": {
        "VpcId": {
            "Description": "ID of an existing Virtual Private Cloud (VPC).",
            "Type": "AWS::EC2::VPC::Id"
        },
        "PublicSubnetId": {
            "Description": "ID of an existing public subnet within the specified VPC.",
            "Type": "AWS::EC2::Subnet::Id"
        },
        "KeyName": {
            "Description": "Name of an existing EC2 key pair to enable SSH access to
 the instance.",
            "Type": "AWS::EC2::KeyPair::KeyName"
        },
        "AMIId": {
            "Description": "Name of a Parameter Store parameter that stores the ID of
 the Amazon Machine Image (AMI).",
            "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
            "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
        }
    },
    "Resources": {
        "InstanceSecurityGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "Enable SSH access via port 22",
                "VpcId": { "Ref": "VpcId" },
                "SecurityGroupIngress": [
                    {
```

```
                        "IpProtocol": "tcp",
                        "FromPort": 22,
                        "ToPort": 22,
                        "CidrIp": "0.0.0.0/0"
                    }
                ]
            }
        },
        "Ec2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "KeyName": { "Ref": "KeyName" },
                "ImageId": { "Ref": "AMIId" },
                "NetworkInterfaces": [
                    {
                        "AssociatePublicIpAddress": "true",
                        "DeviceIndex": "0",
                        "SubnetId": { "Ref": "PublicSubnetId" },
                        "GroupSet": [{ "Ref": "InstanceSecurityGroup" }]
                    }
                ]
            }
        }
    },
    "Outputs": {
        "InstanceId": {
            "Value": { "Ref": "Ec2Instance" }
        }
    }
}
```

YAML

```
Parameters:
  VpcId:
    Description: ID of an existing Virtual Private Cloud (VPC).
    Type: 'AWS::EC2::VPC::Id'
  PublicSubnetId:
    Description: ID of an existing public subnet within the specified VPC.
    Type: 'AWS::EC2::Subnet::Id'
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instance.
    Type: 'AWS::EC2::KeyPair::KeyName'
```

```
  AMIId:
    Description: Name of a Parameter Store parameter that stores the ID of the Amazon
 Machine Image (AMI).
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
Resources:
  InstanceSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable SSH access via port 22
      VpcId: !Ref VpcId
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      KeyName: !Ref KeyName
      ImageId: !Ref AMIId
      NetworkInterfaces:
        - AssociatePublicIpAddress: "true"
          DeviceIndex: "0"
          SubnetId: !Ref PublicSubnetId
          GroupSet:
            - !Ref InstanceSecurityGroup
Outputs:
  InstanceId:
    Value: !Ref Ec2Instance
```

AWS CLI Befehl zum Erstellen des Stacks

Der folgende create-stackBefehl erstellt einen Stack, der auf der Beispielvorlage basiert.

```
aws cloudformation create-stack --stack-name MyStack \
  --template-body file://sampletemplate.json \
  --parameters \
ParameterKey="VpcId",ParameterValue="vpc-a123baa3" \
ParameterKey="PublicSubnetId",ParameterValue="subnet-123a351e" \
ParameterKey="KeyName",ParameterValue="MyKeyName" \
ParameterKey="AMIId",ParameterValue="MyParameterKey"
```

Um einen Parametertyp zu verwenden, der eine Liste von Zeichenfolgen akzeptiert`List<AWS::EC2::Subnet::Id>`, müssen Sie die Kommas innerhalb von `ParameterValue` mit einem doppelten Backslash maskieren, wie im folgenden Beispiel gezeigt.

```
--parameters ParameterKey="SubnetIDs",ParameterValue="subnet-5ea0c127\\,subnet-6194ea3b
\\,subnet-c87f2be0"
```

Überlegungen

Es wird dringend empfohlen, dynamische Verweise zu verwenden, um den Zugriff auf vertrauliche Konfigurationsdefinitionen wie Anmeldeinformationen von Drittanbietern einzuschränken. Weitere Informationen finden Sie unter Ruft mithilfe dynamischer Verweise Werte ab, die in anderen Diensten gespeichert sind.

Wenn Sie Vorlagenbenutzern die Möglichkeit geben möchten, Werte aus verschiedenen Quellen anzugeben AWS-Konten, verwenden Sie keine AWS-spezifischen Parametertypen. Definieren Sie stattdessen Parameter vom Typ `String` oder`CommaDelimitedList`.

Bei den Systems Manager Manager-Parametertypen sind einige Dinge zu beachten:

- Sie können sich die aufgelösten Parameterwerte auf der Registerkarte Parameter des Stacks in der Konsole ansehen oder, indem Sie describe-stacksoder ausführen describe-change-set. Denken Sie daran, dass diese Werte bei der Erstellung oder Aktualisierung des Stacks festgelegt werden, sodass sie sich möglicherweise von den neuesten Werten im Parameter Store unterscheiden.
- Wenn Sie bei Stack-Aktualisierungen die Option Bestehenden Wert verwenden (oder `UsePreviousValue` auf true setzen) verwenden, bedeutet dies, dass Sie weiterhin denselben Parameter Store-Schlüssel verwenden möchten, nicht seinen Wert. CloudFormation ruft immer den neuesten Wert ab.
- Wenn Sie zulässige Werte oder andere Einschränkungen angeben, werden diese anhand der von Ihnen angegebenen Parameterschlüssel CloudFormation überprüft, nicht jedoch anhand ihrer Werte. Sie sollten die Werte im Parameter Store selbst validieren.
- Wenn Sie Stacks erstellen oder aktualisieren und Änderungssätze erstellen, CloudFormation verwendet es den Wert, der zu diesem Zeitpunkt im Parameter Store vorhanden ist. Wenn ein angegebener Parameter nicht im Parameter Store unter dem des Aufrufers vorhanden ist AWS-Konto, wird ein Validierungsfehler CloudFormation zurückgegeben.
- Wenn Sie einen Änderungssatz ausführen, CloudFormation verwendet die Werte, die im Änderungssatz angegeben sind. Sie sollten diese Werte vor der Ausführung des Änderungssatzes

prüfen, da sie sich im Parameter Store zwischen dem Zeitpunkt, an dem der Änderungssatz erstellt wurde, und dem Zeitpunkt, an dem er ausgeführt wird, ändern können.

- Für Parameter Store-Parameter, die in demselben gespeichert sind AWS-Konto, müssen Sie den Parameternamen angeben. Für Parameter Store-Parameter, die von anderen gemeinsam genutzt werden AWS-Konto, müssen Sie den vollständigen Parameter-ARN angeben.

Unterstützte AWS-spezifische Parametertypen

CloudFormation unterstützt die folgenden AWS-spezifischen Typen:

`AWS::EC2::AvailabilityZone::Name`

Gibt eine Availability Zone an (zum Beispiel `us-west-2a`).

`AWS::EC2::Image::Id`

Eine EC2 Amazon-Image-ID, z. `ami-0ff8a91507f77f867` B. Beachten Sie, dass die CloudFormation Konsole keine Dropdownliste mit Werten für diesen Parametertyp anzeigt.

`AWS::EC2::Instance::Id`

Eine EC2 Amazon-Instance-ID, z. `i-1e731a32` B.

`AWS::EC2::KeyPair::KeyName`

Ein EC2 Amazon-Schlüsselpaarname.

`AWS::EC2::SecurityGroup::GroupName`

Ein standardmäßiger VPC-Sicherheitsgruppenname, z. B. `my-sg-abc`

`AWS::EC2::SecurityGroup::Id`

Gibt die ID einer Sicherheitsgruppe an (zum Beispiel `sg-a123fd85`).

`AWS::EC2::Subnet::Id`

Gibt eine Subnetz-ID an (zum Beispiel `subnet-123a351e`).

`AWS::EC2::Volume::Id`

Gibt die ID eines Amazon-EBS-Volumes an (zum Beispiel `vol-3cdd3f56`).

`AWS::EC2::VPC::Id`

Gibt eine VPC-ID an (zum Beispiel `vpc-a123baa3`).

`AWS::Route53::HostedZone::Id`

Gibt die ID einer gehosteten Amazon Route 53-Zone an (zum Beispiel `Z23YXV4OVPL04A`).

`List<AWS::EC2::AvailabilityZone::Name>`

Gibt ein Array von Availability Zones in einer Region an (zum Beispiel `us-west-2a, us-west-2b`).

`List<AWS::EC2::Image::Id>`

Eine Reihe von EC2 Amazon-Bildern IDs, wie `ami-0ff8a91507f77f867, ami-0a584ac55a7631c0c` z. Beachten Sie, dass die CloudFormation Konsole keine Dropdownliste mit Werten für diesen Parametertyp anzeigt.

`List<AWS::EC2::Instance::Id>`

Eine Reihe von EC2 Amazon-Instances IDs, wie `i-1e731a32, i-1e731a34` z.

`List<AWS::EC2::SecurityGroup::GroupName>`

Eine Reihe von standardmäßigen VPC-Sicherheitsgruppennamen, z. B. `my-sg-abc, my-sg-def`

`List<AWS::EC2::SecurityGroup::Id>`

Eine Reihe von Sicherheitsgruppen IDs, z. B. `sg-a123fd85, sg-b456fd85`

`List<AWS::EC2::Subnet::Id>`

Ein Array von Subnetzen IDs, z. B. `subnet-123a351e, subnet-456b351e`

`List<AWS::EC2::Volume::Id>`

Eine Reihe von Amazon EBS-Volumes IDs, wie `vol-3cdd3f56, vol-4cdd3f56` z.

`List<AWS::EC2::VPC::Id>`

Ein Array von VPC IDs, wie `vpc-a123baa3, vpc-b456baa3` z.

`List<AWS::Route53::HostedZone::Id>`

Eine Reihe von von Amazon Route 53-gehosteten Zonen IDs, z. `Z23YXV4OVPL04A, Z23YXV4OVPL04B` B.

Unterstützte Systems Manager Manager-Parametertypen

CloudFormation unterstützt die folgenden Systems Manager Manager-Parametertypen:

`AWS::SSM::Parameter::Name`

Der Name eines Systems Manager-Parameterschlüssels. Verwenden Sie diesen Parametertyp nur, um zu überprüfen, ob ein erforderlicher Parameter vorhanden ist. CloudFormation ruft den tatsächlichen Wert, der dem Parameter zugeordnet ist, nicht ab.

`AWS::SSM::Parameter::Value<String>`

Ein Systems Manager-Parameter, dessen Wert eine Zeichenfolge ist. Dies entspricht dem Parametertyp `String` im Parameter Store.

`AWS::SSM::Parameter::Value<List<String>>` oder
`AWS::SSM::Parameter::Value<CommaDelimitedList>`

Ein Systems Manager-Parameter, dessen Wert eine Liste von Zeichenfolgen ist. Dies entspricht dem Parametertyp `StringList` im Parameter Store.

`AWS::SSM::Parameter::Value<`*`AWS-specific parameter type`*`>`

Ein Systems Manager Manager-Parameter, dessen Wert ein AWS-spezifischer Parametertyp ist.

Das folgende Beispiel gibt z. B. den Typ `AWS::EC2::KeyPair::KeyName` an:

- `AWS::SSM::Parameter::Value<AWS::EC2::KeyPair::KeyName>`

`AWS::SSM::Parameter::Value<List<`*`AWS-specific parameter type`*`>>`

Ein Systems Manager Manager-Parameter, dessen Wert eine Liste von AWS-spezifischen Parametertypen ist.

Das folgende Beispiel gibt z. B. eine Liste der `AWS::EC2::KeyPair::KeyName`-Typen an:

- `AWS::SSM::Parameter::Value<List<AWS::EC2::KeyPair::KeyName>>`

Nicht unterstützte Systems Manager Manager-Parametertypen

CloudFormation unterstützt den folgenden Systems Manager Manager-Parametertyp nicht:

- Listen von Systems Manager Manager-Parametertypen — zum Beispiel: `List<AWS::SSM::Parameter::Value<String>>`

Unterstützt außerdem CloudFormation nicht die Definition von Vorlagenparametern als `SecureString` Systems Manager Manager-Parametertypen. Sie können jedoch sichere

Zeichenfolgen als Parameterwerte für bestimmte Ressourcen angeben. Weitere Informationen finden Sie unter Ruft mithilfe dynamischer Verweise Werte ab, die in anderen Diensten gespeichert sind.

CloudFormation Komplettlösungen

Diese Dokumentation enthält eine Sammlung von exemplarischen Vorgehensweisen, die Ihnen praktische Übungen mit Stack-Bereitstellungen vermitteln sollen.

- Verweisen Sie auf Ressourcenausgaben in einem anderen CloudFormation Stack— Diese exemplarische Vorgehensweise zeigt Ihnen, wie Sie auf Ausgaben von einem Stack innerhalb eines anderen CloudFormation Stacks verweisen können. Anstatt alle Ressourcen in einem einzigen Stapel zusammenzufassen, können Sie verwandte AWS Ressourcen in separaten Stacks erstellen, um modularere und wiederverwendbarere Vorlagen zu erstellen.
- Peern Sie mit einer VPC in einer anderen AWS-Konto— Diese exemplarische Vorgehensweise führt Sie durch den Prozess der Erstellung einer Virtual Private Cloud (VPC) -Peering-Verbindung zwischen zwei VPCs unterschiedlichen Anbietern. AWS-Konten Mit VPC-Peering können Sie den Datenverkehr zwischen den Ressourcen weiterleiten VPCs und darauf zugreifen, als ob sie Teil desselben Netzwerks wären.
- Erstellen Sie eine skalierte Anwendung mit Load Balancing— Erfahren Sie, wie Sie damit eine skalierbare Anwendung CloudFormation mit Lastenausgleich erstellen können. In dieser exemplarischen Vorgehensweise wird das Erstellen einer Auto Scaling Scaling-Gruppe, eines Load Balancers und anderer verwandter Ressourcen behandelt, um sicherzustellen, dass Ihre Anwendung unterschiedliche Datenverkehrslasten bewältigen und eine hohe Verfügbarkeit aufrechterhalten kann.
- Stellen Sie Anwendungen auf Amazon bereit EC2— Erfahren Sie, wie CloudFormation Sie Ihre Anwendung auf EC2 Amazon-Instances automatisch installieren, konfigurieren und starten können. Auf diese Weise können Sie Bereitstellungen problemlos duplizieren und bestehende Installationen aktualisieren, ohne eine direkte Verbindung zu den Instances herstellen zu müssen.
- Aktualisieren eines Stacks— Gehen Sie eine einfache Abfolge von Updates für einen laufenden Stack mit CloudFormation durch.
- Führen Sie blue/green ECS-Bereitstellungen durch, indem Sie CodeDeploy CloudFormation— Erfahren Sie, wie CloudFormation Sie mithilfe von AWS CodeDeploy blue/green deployments on Amazon ECS. Blue/green Bereitstellungen Ihre Anwendungen oder Dienste mit minimalen Ausfallzeiten aktualisieren können.

Verweisen Sie auf Ressourcenausgaben in einem anderen CloudFormation Stack

Diese exemplarische Vorgehensweise zeigt Ihnen, wie Sie auf Ausgaben von einem CloudFormation Stapel in einem anderen Stapel verweisen, um modularere und wiederverwendbarere Vorlagen zu erstellen.

Anstatt alle Ressourcen in einem einzigen Stapel zusammenzufassen, erstellen Sie verwandte AWS Ressourcen in separaten Stacks. Anschließend können Sie auf die erforderlichen Ressourcenausgaben aus anderen Stacks verweisen. Durch Einschränken von Stack-übergreifenden Verweisen auf Ausgaben können Sie die Teile eines Stacks steuern, auf die von anderen Stacks verwiesen wird.

Angenommen, Sie verfügen über einen Netzwerk-Stack mit einer VPC, einer Sicherheitsgruppe und einem Subnetz für öffentliche Webanwendungen sowie über einen öffentlichen Webanwendungs-Stack. Um sicherzustellen, dass die Webanwendungen die Sicherheitsgruppe und das Subnetz aus dem Netzwerk-Stack verwenden, erstellen Sie einen Stack-übergreifenden Verweis, über den der Webanwendungs-Stack auf Ressourcenausgaben vom Netzwerk-Stack verweisen kann. Mit einem Stack-übergreifenden Verweis müssen die Besitzer der Webanwendungs-Stacks keine Netzwerkregeln oder Komponenten erstellen oder verwalten.

Um einen Stack-übergreifenden Verweis zu erstellen, verwenden Sie das `Export`-Ausgabefeld, um den Wert einer Ressourcenausgabe für den Export zu markieren. Verwenden Sie dann die intrinsische Funktion `Fn::ImportValue` , um den Wert zu importieren. Weitere Informationen finden Sie unter Holen Sie sich exportierte Ausgaben aus einem bereitgestellten CloudFormation Stack.

> **Note**
>
> CloudFormation ist ein kostenloser Service. Die AWS Ressourcen, die Sie in Ihren Stacks aufnehmen, werden Ihnen jedoch zum jeweils aktuellen Tarif in Rechnung gestellt. Weitere Informationen zu AWS -Preisen finden Sie auf der Detailseite der einzelnen Produkte.

Themen

- Verwenden Sie eine Beispielvorlage, um einen Netzwerk-Stack zu erstellen
- Verwenden Sie eine Beispielvorlage, um einen Webanwendungsstapel zu erstellen
- Stellen Sie sicher, dass der Stack wie vorgesehen funktioniert

- Behebung von AMI-Zuordnungsfehlern
- Bereinigen Ihrer Ressourcen

Verwenden Sie eine Beispielvorlage, um einen Netzwerk-Stack zu erstellen

Bevor Sie mit dieser exemplarischen Vorgehensweise beginnen, überprüfen Sie, ob Sie über IAM-Berechtigungen zur Nutzung aller der folgenden Dienste verfügen: Amazon VPC, Amazon EC2 und. CloudFormation

Der Netzwerk-Stack umfasst die VPC, die Sicherheitsgruppe und das Subnetz, das Sie im Webanwendungs-Stack verwenden. Zusätzlich zu diesen Regeln erstellt der Netzwerk-Stack ein Internet-Gateway und Routing-Tabellen, um den öffentlichen Zugriff zu ermöglichen.

Sie müssen diesen Stack erstellen, bevor Sie den Webanwendungs-Stack erstellen. Wenn Sie den Webanwendungs-Stack zuerst erstellen, verfügt dieser nicht über eine Sicherheitsgruppe oder ein Subnetz.

Die Stack-Vorlage ist unter der folgenden URL verfügbar: https://s3.amazonaws.com/cloudformation-examples/user - .template. guide/cross-stack/SampleNetworkCrossStack Um die Ressourcen zu sehen, die der Stack erstellt, wählen Sie den Link aus, woraufhin die Vorlage geöffnet wird. In `Outputs` diesem Abschnitt können Sie die Netzwerkressourcen sehen, die mit der Beispielvorlage exportiert werden. Den Namen der exportierten Ressourcen ist der Stack-Name vorangestellt, falls Sie Netzwerkressourcen von anderen Stacks exportieren. Wenn Benutzer Netzwerkressourcen importieren, können sie angeben, von welchem Stack die Ressourcen importiert werden.

So erstellen Sie den Netzwerk-Stack

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der Seite „Stacks“ oben rechts die Option „Stack erstellen“ und anschließend „Mit neuen Ressourcen (Standard)“ aus.
3. Wählen Sie Bestehende Vorlage auswählen und wählen Sie im Abschnitt Vorlage angeben die Option Amazon S3 S3-URL aus.
4. Fügen Sie für Amazon S3 S3-URL die folgende URL ein:**`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template`**.
5. Wählen Sie Weiter aus.

6. Geben Sie für Stack name **`SampleNetworkCrossStack`** ein und wählen Sie dann Next aus.

> **Note**
>
> Notieren Sie sich den Namen dieses Stacks. Sie benötigen den Stack-Namen, wenn Sie den Webanwendungs-Stack öffnen.

7. Wählen Sie Weiter aus. Für diese Anleitung müssen Sie keine Tags hinzufügen oder erweiterte Einstellungen angeben.
8. Stellen Sie sicher, dass der Stack-Name und die Vorlagen-URL richtig sind und wählen Sie Create stack (Stack erstellen) aus.

   Es kann mehrere Minuten dauern CloudFormation , bis Ihr Stack erstellt ist. Warten Sie, bis alle Ressourcen erfolgreich erstellt wurden, bevor Sie mit der Erstellung des Webanwendungs-Stacks fortfahren.
9. Betrachten Sie die Stack-Ereignisse, um den Fortschritt zu überwachen. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

Verwenden Sie eine Beispielvorlage, um einen Webanwendungsstapel zu erstellen

Der Webanwendungsstapel erstellt eine EC2 Instanz, die die Sicherheitsgruppe und das Subnetz aus dem Netzwerkstapel verwendet.

Sie müssen diesen Stack genauso AWS-Region wie den Netzwerk-Stack erstellen.

Die Stack-Vorlage ist unter der folgenden URL verfügbar: https://s3.amazonaws.com/cloudformation-examples/user- guide/cross-stack/SampleWebAppCrossStack .template. Um die Ressourcen zu sehen, die der Stack erstellt, wählen Sie den Link aus, woraufhin die Vorlage geöffnet wird. Sehen Sie sich in `Resources` diesem Abschnitt die Eigenschaften der EC2 Instanz an. Mithilfe der `Fn::ImportValue`-Funktion können sehen, wie die Netzwerkressourcen von einem anderen Stack importiert werden.

So erstellen Sie den Webanwendungs-Stack

1. Wählen Sie auf der Seite „Stacks“ oben rechts die Option „Stack erstellen“ und dann „Mit neuen Ressourcen (Standard)“ aus.
2. Wählen Sie Bestehende Vorlage auswählen und wählen Sie im Abschnitt Vorlage angeben die Option Amazon S3 S3-URL aus.

3. Fügen Sie für Amazon S3 S3-URL die folgende URL ein:**`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template`**.
4. Wählen Sie Weiter aus.
5. Geben Sie für Stack name **`SampleWebAppCrossStack`** ein. Verwenden Sie im Abschnitt Parameter den Standardwert für den NetworkStackNameParameter und wählen Sie dann Weiter.

   Die Beispielvorlage verwendet den Parameterwert, um anzugeben, von welchem Stack Werte importiert werden sollen.
6. Wählen Sie Weiter aus. Für diese Anleitung müssen Sie keine Tags hinzufügen oder erweiterte Einstellungen angeben.
7. Stellen Sie sicher, dass der Stack-Name und die Vorlagen-URL richtig sind und wählen Sie Create stack (Stack erstellen) aus.

   Die Erstellung Ihres Stacks kann mehrere Minuten CloudFormation dauern.

Stellen Sie sicher, dass der Stack wie vorgesehen funktioniert

Nachdem der Stack erstellt wurde, zeigen Sie seine Ressourcen an, und notieren Sie sich die Instance-ID. Weitere Informationen zum Anzeigen von Stack-Ressourcen finden Sie unter Stack-Informationen von der CloudFormation Konsole aus anzeigen.

Um die Sicherheitsgruppe und das Subnetz der Instance zu überprüfen, sehen Sie sich die Eigenschaften der Instance in der EC2 Amazon-Konsole an. Wenn die Instance die Sicherheitsgruppe und das Subnetz aus dem `SampleNetworkCrossStack`-Stack verwendet, haben Sie erfolgreich einen Stack-übergreifenden Verweis erstellt.

Verwenden Sie die Konsole, um die Stack-Ausgaben und die URL der Beispiel-Website anzuzeigen, die die Webanwendung ausführt. Weitere Informationen finden Sie unter Stack-Informationen von der CloudFormation Konsole aus anzeigen.

Behebung von AMI-Zuordnungsfehlern

Wenn Sie den Fehler erhalten`Template error: Unable to get mapping for AWSRegionArch2AMI::[region]::HVM64`, enthält die Vorlage keine AMI-Zuordnung für Ihre AWS-Region. Anstatt das Mapping zu aktualisieren, empfehlen wir, öffentliche Parameter von Systems Manager zu verwenden, um dynamisch auf die neuesten Daten zu verweisen AMIs:

1. Laden Sie die `SampleWebAppCrossStack` Vorlage von: https://s3.amazonaws.com/cloudformation-examples/user- guide/cross-stack/SampleWebAppCrossStack .template auf Ihren lokalen Computer herunter.
2. Löschen Sie den gesamten `AWSRegionArch2AMI` Mapping-Bereich.
3. Fügen Sie den folgenden Systems Manager Manager-Parameter hinzu:

```
"LatestAmiId": {
  "Description": "The latest Amazon Linux 2 AMI from the Parameter Store",
    "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
    "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
  }
```

4. Ersetzen Sie die bestehende `ImageId` Referenz:

```
"ImageId": { "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" } ,
 "HVM64" ] },
```

   mit:

```
"ImageId": { "Ref": "LatestAmiId" },
```

   Dieser Parameter wird automatisch auf das neueste Amazon Linux 2-AMI für die Region aufgelöst, in der Sie den Stack bereitstellen.

   Verwenden Sie für andere Linux-Distributionen den entsprechenden Parameterpfad. Weitere Informationen finden Sie unter Discovering public parameters in Parameter Store im AWS Systems Manager Benutzerhandbuch.
5. Laden Sie die geänderte Vorlage in einen S3-Bucket in Ihrem Konto hoch:

```
aws s3 cp SampleWebAppCrossStack.template s3://amzn-s3-demo-bucket/
```

6. Geben Sie beim Erstellen des Stacks Ihre S3-Vorlagen-URL anstelle der Beispiel-URL an.

Bereinigen Ihrer Ressourcen

Um sicherzustellen, dass keine Kosten für unerwünschte Services anfallen, löschen Sie die Stacks.

So löschen Sie die Stacks

1. Wählen Sie in der CloudFormation Konsole den `SampleWebAppCrossStack` Stack aus.
2. Wählen Sie die Option Actions (Aktionen) und anschließend Delete stack (Stack löschen) aus.
3. Wählen Sie in der Bestätigungsmitteilung die Option Löschen aus.
4. Nachdem der Stapel gelöscht wurde, wiederholen Sie dieselben Schritte für den `SampleNetworkCrossStack` Stapel.

> **Note**
>
> Warten Sie, bis der `SampleWebAppCrossStack` Stapel CloudFormation vollständig gelöscht ist. Wenn die EC2 Instance noch in der VPC läuft, CloudFormation wird die VPC im Stack nicht gelöscht. `SampleNetworkCrossStack`

Peern Sie mit einer VPC in einer anderen AWS-Konto

Sie können ein Peering mit einer Virtual Private Cloud (VPC) in einer anderen durchführen, AWS-Konto indem Sie AWS::EC2::VPCPeeringConnection. Dadurch wird eine Netzwerkverbindung zwischen zwei Personen hergestellt VPCs , sodass Sie den Verkehr zwischen ihnen weiterleiten können, sodass sie so kommunizieren können, als ob sie sich innerhalb desselben Netzwerks befänden. Eine VPC-Peering-Verbindung kann den Datenzugriff und die Datenübertragung erleichtern.

Um eine VPC-Peering-Verbindung herzustellen, müssen Sie zwei separate Verbindungen AWS-Konten innerhalb eines einzigen Stacks autorisieren. CloudFormation

Weitere Informationen zu VPC-Peering und seinen Einschränkungen finden Sie im Amazon VPC Peering Guide.

Voraussetzungen

1. Sie benötigen eine Peer-VPC-ID, eine AWS-Konto Peer-ID und eine kontoübergreifende Zugriffsrolle für die Peering-Verbindung.

> **Note**
>
> Diese Anleitung bezieht sich auf zwei Konten: Das erste Konto ist ein Konto, das kontenübergreifendes Peering zulässt (annehmendes Konto). Das zweite Konto ist das Konto, das die Peering-Verbindung anfordert (Auftraggeberkonto).

2. Um die VPC-Peering-Verbindung zu akzeptieren, müssen Sie die Rolle für den kontoübergreifenden Zugriff übernehmen können. Die Ressource verhält sich wie die Ressource einer VPC-Peering-Verbindung in demselben Konto. Informationen darüber, wie ein IAM-Administrator Berechtigungen zur Übernahme der kontoübergreifenden Rolle erteilt, finden Sie im IAM-Benutzerhandbuch unter Gewähren von Benutzerberechtigungen zum Rollenwechsel.

Schritt 1: Erstellen einer VPC und einer kontoübergreifenden Rolle

In diesem Schritt erstellen Sie die VPC und die Rolle im annehmenden Konto.

So erstellen Sie eine VPC- und eine kontoübergreifende Zugriffsrolle

1. Melden Sie sich bei /cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole. https://console.aws.amazon.com
2. Wählen Sie auf der Seite „Stacks“ oben rechts die Option „Stack erstellen“ und anschließend „Mit neuen Ressourcen (Standard)“ aus.
3. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorhandene Vorlage auswählen und anschließend Vorlagendatei hochladen, Datei auswählen aus.
4. Öffnen Sie einen Texteditor auf Ihrem lokalen Computer und fügen Sie eine der folgenden Vorlagen hinzu. Speichern Sie die Datei und kehren Sie zur Konsole zurück, um sie als Vorlagendatei auszuwählen.

   Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and an assumable role for cross account VPC
 peering.",
  "Parameters": {
    "PeerRequesterAccountId": {
      "Type": "String"
    }
```

```
  },
  "Resources": {
    "vpc": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": "10.1.0.0/16",
        "EnableDnsSupport": false,
        "EnableDnsHostnames": false,
        "InstanceTenancy": "default"
      }
    },
    "peerRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Principal": {
                "AWS": {
                  "Ref": "PeerRequesterAccountId"
                }
              },
              "Action": [
                "sts:AssumeRole"
              ],
              "Effect": "Allow"
            }
          ]
        },
        "Path": "/",
        "Policies": [
          {
            "PolicyName": "root",
            "PolicyDocument": {
              "Version": "2012-10-17",
              "Statement": [
                {
                  "Effect": "Allow",
                  "Action": "ec2:AcceptVpcPeeringConnection",
                  "Resource": "*"
                }
              ]
            }
          }
```

```
        ]
      }
    }
  },
  "Outputs": {
    "VPCId": {
      "Value": {
        "Ref": "vpc"
      }
    },
    "RoleARN": {
      "Value": {
        "Fn::GetAtt": [
          "peerRole",
          "Arn"
        ]
      }
    }
  }
}
```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and an assumable role for cross account VPC peering.
Parameters:
  PeerRequesterAccountId:
    Type: String
Resources:
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.1.0.0/16
      EnableDnsSupport: false
      EnableDnsHostnames: false
      InstanceTenancy: default
  peerRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Principal:
              AWS: !Ref PeerRequesterAccountId
```

```
            Action:
              - 'sts:AssumeRole'
            Effect: Allow
      Path: /
      Policies:
        - PolicyName: root
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action: 'ec2:AcceptVpcPeeringConnection'
                Resource: '*'
Outputs:
  VPCId:
    Value: !Ref vpc
  RoleARN:
    Value: !GetAtt
      - peerRole
      - Arn
```

5. Wählen Sie Weiter aus.
6. Geben Sie dem Stack einen Namen (z. B.**`VPC-owner`**) und geben Sie dann die AWS-Konto ID des Anfordererkontos in das PeerRequesterAccountIdFeld ein.
7. Akzeptieren Sie die Standardwerte und wählen Sie dann Weiter aus.
8. Wählen Sie Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen erstellt werden, und wählen Sie dann Stack erstellen aus.

Schritt 2: Erstellen einer Vorlage, die **`AWS::EC2::VPCPeeringConnection`** enthält

Nachdem Sie die VPC und die kontoübergreifende Rolle erstellt haben, können Sie über ein anderes Konto AWS-Konto (das Konto des Anforderers) eine Peering-Verbindung mit der VPC herstellen.

Um eine Vorlage zu erstellen, die die VPCPeeringVerbindungsressource AWSEC2:::: enthält

1. Gehen Sie zurück zur Startseite der AWS CloudFormation Konsole.
2. Wählen Sie auf der Seite „Stacks“ oben rechts die Option „Stack erstellen“ und anschließend „Mit neuen Ressourcen (Standard)“ aus.
3. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorhandene Vorlage auswählen und anschließend Vorlagendatei hochladen, Datei auswählen aus.

4. Öffnen Sie einen Texteditor auf Ihrem lokalen Computer und fügen Sie eine der folgenden Vorlagen hinzu. Speichern Sie die Datei und kehren Sie zur Konsole zurück, um sie als Vorlagendatei auszuwählen.

   Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and a VPC Peering connection using the PeerRole to
 accept.",
  "Parameters": {
    "PeerVPCAccountId": {
      "Type": "String"
    },
    "PeerVPCId": {
      "Type": "String"
    },
    "PeerRoleArn": {
      "Type": "String"
    }
  },
  "Resources": {
    "vpc": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": "10.2.0.0/16",
        "EnableDnsSupport": false,
        "EnableDnsHostnames": false,
        "InstanceTenancy": "default"
      }
    },
    "vpcPeeringConnection": {
      "Type": "AWS::EC2::VPCPeeringConnection",
      "Properties": {
        "VpcId": {
          "Ref": "vpc"
        },
        "PeerVpcId": {
          "Ref": "PeerVPCId"
        },
        "PeerOwnerId": {
          "Ref": "PeerVPCAccountId"
        },
```

```
        "PeerRoleArn": {
          "Ref": "PeerRoleArn"
        }
      }
    }
  },
  "Outputs": {
    "VPCId": {
      "Value": {
        "Ref": "vpc"
      }
    },
    "VPCPeeringConnectionId": {
      "Value": {
        "Ref": "vpcPeeringConnection"
      }
    }
  }
}
```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and a VPC Peering connection using the PeerRole to
 accept.
Parameters:
  PeerVPCAccountId:
    Type: String
  PeerVPCId:
    Type: String
  PeerRoleArn:
    Type: String
Resources:
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.2.0.0/16
      EnableDnsSupport: false
      EnableDnsHostnames: false
      InstanceTenancy: default
  vpcPeeringConnection:
    Type: AWS::EC2::VPCPeeringConnection
    Properties:
```

```
      VpcId: !Ref vpc
      PeerVpcId: !Ref PeerVPCId
      PeerOwnerId: !Ref PeerVPCAccountId
      PeerRoleArn: !Ref PeerRoleArn
Outputs:
  VPCId:
    Value: !Ref vpc
  VPCPeeringConnectionId:
    Value: !Ref vpcPeeringConnection
```

5. Wählen Sie Weiter aus.
6. Geben Sie dem Stapel einen Namen (z. B. **`VPC-peering-connection`**).
7. Akzeptieren Sie die Standardwerte und wählen Sie dann Weiter aus.
8. Wählen Sie Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen erstellt werden, und wählen Sie dann Stapel erstellen aus.

Erstellen Sie eine Vorlage mit einer sehr restriktiven Richtlinie

Vielleicht möchten Sie eine stark einschränkende Richtlinie für das Peering zwischen Ihrer VPC und einer anderen AWS-Konto erstellen.

Die folgende Beispielvorlage zeigt, wie die Vorlage des VPC-Peer-Besitzers (das annehmende Konto, das Sie oben in Schritt 1 erstellt haben) geändert wird, damit sie restriktiver ist.

Example JSON

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Description":"Create a VPC and an assumable role for cross account VPC peering.",
  "Parameters":{
    "PeerRequesterAccountId":{
      "Type":"String"
    }
  },
  "Resources":{
    "peerRole":{
      "Type":"AWS::IAM::Role",
      "Properties":{
        "AssumeRolePolicyDocument":{
          "Statement":[
            {
```

```
              "Action":[
                "sts:AssumeRole"
              ],
              "Effect":"Allow",
              "Principal":{
                "AWS":{
                  "Ref":"PeerRequesterAccountId"
                }
              }
            }
          ]
        },
        "Path":"/",
        "Policies":[
          {
            "PolicyDocument":{
              "Statement":[
                {
                  "Action":"ec2:acceptVpcPeeringConnection",
                  "Effect":"Allow",
                  "Resource":{
                    "Fn::Sub":"arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/${vpc}"
                  }
                },
                {
                  "Action":"ec2:acceptVpcPeeringConnection",
                  "Condition":{
                    "StringEquals":{
                      "ec2:AccepterVpc":{
                        "Fn::Sub":"arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/
${vpc}"
                      }
                    }
                  },
                  "Effect":"Allow",
                  "Resource":{
                    "Fn::Sub":"arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc-
peering-connection/*"
                  }
                }
              ],
              "Version":"2012-10-17"
            },
            "PolicyName":"root"
```

```
          }
        ]
      }
    },
    "vpc":{
      "Type":"AWS::EC2::VPC",
      "Properties":{
        "CidrBlock":"10.1.0.0/16",
        "EnableDnsHostnames":false,
        "EnableDnsSupport":false,
        "InstanceTenancy":"default"
      }
    }
  },
  "Outputs":{
    "RoleARN":{
      "Value":{
        "Fn::GetAtt":[
          "peerRole",
          "Arn"
        ]
      }
    },
    "VPCId":{
      "Value":{
        "Ref":"vpc"
      }
    }
  }
}
```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and an assumable role for cross account VPC peering.
Parameters:
  PeerRequesterAccountId:
    Type: String
Resources:
  peerRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
```

```
        Statement:
          - Action:
              - 'sts:AssumeRole'
            Effect: Allow
            Principal:
              AWS:
                Ref: PeerRequesterAccountId
      Path: /
      Policies:
        - PolicyDocument:
            Statement:
              - Action: 'ec2:acceptVpcPeeringConnection'
                Effect: Allow
                Resource:
                  'Fn::Sub': 'arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/${vpc}'
              - Action: 'ec2:acceptVpcPeeringConnection'
                Condition:
                  StringEquals:
                    'ec2:AccepterVpc':
                      'Fn::Sub': 'arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/
${vpc}'
                Effect: Allow
                Resource:
                  'Fn::Sub': >-
                    arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc-peering-
connection/*
            Version: 2012-10-17
          PolicyName: root
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.1.0.0/16
      EnableDnsHostnames: false
      EnableDnsSupport: false
      InstanceTenancy: default
Outputs:
  RoleARN:
    Value:
      'Fn::GetAtt':
        - peerRole
        - Arn
  VPCId:
    Value:
```

```
      Ref: vpc
```

Um auf die VPC zuzugreifen, können Sie dieselbe Auftraggebervorlage wie in Schritt 2 oben verwenden.

Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für VPC-Peering im Amazon VPC Peering Guide.

Erstellen Sie eine skalierte Anwendung mit Load Balancing

Für diese Anleitung erstellen Sie einen Stack, der Ihnen hilft, eine skalierte und lastverteilte Anwendung einzurichten. Die schrittweise Anleitung enthält eine Beispielvorlage, mit denen Sie den Stack erstellen können. Die Beispielvorlage stellt eine Auto-Scaling-Gruppe, einen Application Load Balancer, Sicherheitsgruppen, die den Datenverkehr zum Load Balancer und zur Auto-Scaling-Gruppe steuern, sowie eine Amazon-SNS-Benachrichtigungskonfiguration zur Veröffentlichung von Benachrichtigungen über Skalierungsaktivitäten bereit.

Diese Vorlage erstellt eine oder mehrere EC2 Amazon-Instances und einen Application Load Balancer. Ihnen werden Gebühren für die verwendeten AWS -Ressourcen berechnet, wenn Sie einen Stack aus dieser Vorlage erstellen.

Vollständige Stack-Vorlage

Fangen wir mit der Vorlage an.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  InstanceType:
    Description: The EC2 instance type
    Type: String
    Default: t3.micro
    AllowedValues:
      - t3.micro
      - t3.small
      - t3.medium
  KeyName:
    Description: Name of an existing EC2 key pair to allow SSH access to the instances
    Type: 'AWS::EC2::KeyPair::KeyName'
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
```

```
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
  OperatorEmail:
    Description: The email address to notify when there are any scaling activities
    Type: String
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  Subnets:
    Type: 'List<AWS::EC2::Subnet::Id>'
    Description: At least two public subnets in different Availability Zones in the
 selected VPC
  VPC:
    Type: 'AWS::EC2::VPC::Id'
    Description: A virtual private cloud (VPC) that enables resources in public subnets
 to connect to the internet
Resources:
  ELBSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: ELB Security Group
      VpcId: !Ref VPC
      SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
  EC2SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: EC2 Security Group
      VpcId: !Ref VPC
      SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        SourceSecurityGroupId:
          Fn::GetAtt:
          - ELBSecurityGroup
          - GroupId
```

```
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
  EC2TargetGroup:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
      HealthCheckIntervalSeconds: 30
      HealthCheckProtocol: HTTP
      HealthCheckTimeoutSeconds: 15
      HealthyThresholdCount: 5
      Matcher:
        HttpCode: '200'
      Name: EC2TargetGroup
      Port: 80
      Protocol: HTTP
      TargetGroupAttributes:
      - Key: deregistration_delay.timeout_seconds
        Value: '20'
      UnhealthyThresholdCount: 3
      VpcId: !Ref VPC
  ALBListener:
    Type: AWS::ElasticLoadBalancingV2::Listener
    Properties:
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref EC2TargetGroup
      LoadBalancerArn: !Ref ApplicationLoadBalancer
      Port: 80
      Protocol: HTTP
  ApplicationLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Scheme: internet-facing
      Subnets: !Ref Subnets
      SecurityGroups:
        - !GetAtt ELBSecurityGroup.GroupId
  LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
      LaunchTemplateData:
        ImageId: !Ref LatestAmiId
        InstanceType: !Ref InstanceType
```

```
        KeyName: !Ref KeyName
        SecurityGroupIds:
          - !Ref EC2SecurityGroup
        UserData:
          Fn::Base64: !Sub |
            #!/bin/bash
            yum update -y
            yum install -y httpd
            systemctl start httpd
            systemctl enable httpd
            echo "<h1>Hello World!</h1>" > /var/www/html/index.html
  NotificationTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint: !Ref OperatorEmail
          Protocol: email
  WebServerGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      LaunchTemplate:
        LaunchTemplateId: !Ref LaunchTemplate
        Version: !GetAtt LaunchTemplate.LatestVersionNumber
      MaxSize: '3'
      MinSize: '1'
      NotificationConfigurations:
        - TopicARN: !Ref NotificationTopic
          NotificationTypes: ['autoscaling:EC2_INSTANCE_LAUNCH',
 'autoscaling:EC2_INSTANCE_LAUNCH_ERROR', 'autoscaling:EC2_INSTANCE_TERMINATE',
 'autoscaling:EC2_INSTANCE_TERMINATE_ERROR']
      TargetGroupARNs:
        - !Ref EC2TargetGroup
      VPCZoneIdentifier: !Ref Subnets
```

JSON

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Parameters":{
    "InstanceType":{
      "Description":"The EC2 instance type",
      "Type":"String",
      "Default":"t3.micro",
```

```
      "AllowedValues":[
        "t3.micro",
        "t3.small",
        "t3.medium"
      ]
    },
    "KeyName":{
      "Description":"Name of an existing EC2 key pair to allow SSH access to the
instances",
      "Type":"AWS::EC2::KeyPair::KeyName"
    },
    "LatestAmiId":{
      "Description":"The latest Amazon Linux 2 AMI from the Parameter Store",
      "Type":"AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default":"/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    },
    "OperatorEmail":{
      "Description":"The email address to notify when there are any scaling
activities",
      "Type":"String"
    },
    "SSHLocation":{
      "Description":"The IP address range that can be used to SSH to the EC2
instances",
      "Type":"String",
      "MinLength":9,
      "MaxLength":18,
      "Default":"0.0.0.0/0",
      "ConstraintDescription":"Must be a valid IP CIDR range of the form x.x.x.x/x."
    },
    "Subnets":{
      "Type":"List<AWS::EC2::Subnet::Id>",
      "Description":"At least two public subnets in different Availability Zones in the
selected VPC"
    },
    "VPC":{
      "Type":"AWS::EC2::VPC::Id",
      "Description":"A virtual private cloud (VPC) that enables resources in public
subnets to connect to the internet"
    }
  },
  "Resources":{
    "ELBSecurityGroup":{
      "Type":"AWS::EC2::SecurityGroup",
```

```
      "Properties":{
        "GroupDescription":"ELB Security Group",
        "VpcId":{
          "Ref":"VPC"
        },
        "SecurityGroupIngress":[
          {
            "IpProtocol":"tcp",
            "FromPort":80,
            "ToPort":80,
            "CidrIp":"0.0.0.0/0"
          }
        ]
      }
    },
    "EC2SecurityGroup":{
      "Type":"AWS::EC2::SecurityGroup",
      "Properties":{
        "GroupDescription":"EC2 Security Group",
        "VpcId":{
          "Ref":"VPC"
        },
        "SecurityGroupIngress":[
          {
            "IpProtocol":"tcp",
            "FromPort":80,
            "ToPort":80,
            "SourceSecurityGroupId":{
              "Fn::GetAtt":[
                "ELBSecurityGroup",
                "GroupId"
              ]
            }
          },
          {
            "IpProtocol":"tcp",
            "FromPort":22,
            "ToPort":22,
            "CidrIp":{
              "Ref":"SSHLocation"
            }
          }
        ]
      }
```

```
    },
    "EC2TargetGroup":{
      "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
      "Properties":{
        "HealthCheckIntervalSeconds":30,
        "HealthCheckProtocol":"HTTP",
        "HealthCheckTimeoutSeconds":15,
        "HealthyThresholdCount":5,
        "Matcher":{
          "HttpCode":"200"
        },
        "Name":"EC2TargetGroup",
        "Port":80,
        "Protocol":"HTTP",
        "TargetGroupAttributes":[
          {
            "Key":"deregistration_delay.timeout_seconds",
            "Value":"20"
          }
        ],
        "UnhealthyThresholdCount":3,
        "VpcId":{
          "Ref":"VPC"
        }
      }
    },
    "ALBListener":{
      "Type":"AWS::ElasticLoadBalancingV2::Listener",
      "Properties":{
        "DefaultActions":[
          {
            "Type":"forward",
            "TargetGroupArn":{
              "Ref":"EC2TargetGroup"
            }
          }
        ],
        "LoadBalancerArn":{
          "Ref":"ApplicationLoadBalancer"
        },
        "Port":80,
        "Protocol":"HTTP"
      }
    },
```

```
    "ApplicationLoadBalancer":{
      "Type":"AWS::ElasticLoadBalancingV2::LoadBalancer",
      "Properties":{
        "Scheme":"internet-facing",
        "Subnets":{
          "Ref":"Subnets"
        },
        "SecurityGroups":[
          {
            "Fn::GetAtt":[
              "ELBSecurityGroup",
              "GroupId"
            ]
          }
        ]
      }
    },
    "LaunchTemplate":{
      "Type":"AWS::EC2::LaunchTemplate",
      "Properties":{
        "LaunchTemplateName":{
          "Fn::Sub":"${AWS::StackName}-launch-template"
        },
        "LaunchTemplateData":{
          "ImageId":{
            "Ref":"LatestAmiId"
          },
          "InstanceType":{
            "Ref":"InstanceType"
          },
          "KeyName":{
            "Ref":"KeyName"
          },
          "SecurityGroupIds":[
            {
              "Ref":"EC2SecurityGroup"
            }
          ],
          "UserData":{
            "Fn::Base64":{
              "Fn::Join":[
                "",
                [
                  "#!/bin/bash\n",
```

```
                "yum update -y\n",
                "yum install -y httpd\n",
                "systemctl start httpd\n",
                "systemctl enable httpd\n",
                "echo \"<h1>Hello World!</h1>\" > /var/www/html/index.html"
              ]
            ]
          }
        }
      }
    }
  },
  "NotificationTopic":{
    "Type":"AWS::SNS::Topic",
    "Properties":{
      "Subscription":[
        {
          "Endpoint":{
            "Ref":"OperatorEmail"
          },
          "Protocol":"email"
        }
      ]
    }
  },
  "WebServerGroup":{
    "Type":"AWS::AutoScaling::AutoScalingGroup",
    "Properties":{
      "LaunchTemplate":{
        "LaunchTemplateId":{
          "Ref":"LaunchTemplate"
        },
        "Version":{
          "Fn::GetAtt":[
            "LaunchTemplate",
            "LatestVersionNumber"
          ]
        }
      },
      "MaxSize":"3",
      "MinSize":"1",
      "NotificationConfigurations":[
        {
          "TopicARN":{
```

```
                "Ref":"NotificationTopic"
              },
              "NotificationTypes":[
                "autoscaling:EC2_INSTANCE_LAUNCH",
                "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
                "autoscaling:EC2_INSTANCE_TERMINATE",
                "autoscaling:EC2_INSTANCE_TERMINATE_ERROR"
              ]
            }
          ],
          "TargetGroupARNs":[
            {
              "Ref":"EC2TargetGroup"
            }
          ],
          "VPCZoneIdentifier":{
            "Ref":"Subnets"
          }
        }
      }
    }
  }
```

Walkthrough zu Vorlagen

Der erste Teil dieser Vorlage spezifiziert die `Parameters`. Jedem Parameter muss zur Laufzeit ein Wert zugewiesen werden, AWS CloudFormation damit der Stack erfolgreich bereitgestellt werden kann. Später in der Vorlage angegebene Ressourcen verweisen auf diese Werte und verwenden die Daten.

- `InstanceType`: Der EC2 Instance-Typ, den Amazon EC2 Auto Scaling bereitstellt. Wenn Sie keinen anderen Wert angeben, wird der Standardwert von `t3.micro` verwendet.
- `KeyName`: Ein vorhandenes EC2 key pair, das den SSH-Zugriff auf die Instanzen ermöglicht.
- `LatestAmiId`: Das Amazon Machine Image (AMI) für die Instances. Falls nicht angegeben, werden Ihre Instances mit einem Amazon Linux 2-AMI gestartet, wobei ein AWS Systems Manager öffentlicher Parameter verwendet wird, der von verwaltet wird AWS. Weitere Informationen finden Sie unter Öffentliche Parameter kennzeichnen im AWS Systems Manager -Benutzerhandbuch.
- `OperatorEmail`: Die E-Mail-Adresse, an die Sie Skalierungsaktivitäten senden möchten.
- `SSHLocation`: Der IP-Adressbereich, der für SSH-Verbindungen mit den Instances verwendet werden kann.

- `Subnets`: Mindestens zwei Subnetze in verschiedenen Availability Zones.
- `VPC`: Eine Virtual Private Cloud (VPC) in Ihrem Konto, mit der Ressourcen in öffentlichen Subnetzen eine Verbindung zum Internet herstellen können.

> **Note**
>
> Sie können die Standard-VPC und die Standardsubnetze verwenden, um Instances den Zugriff auf das Internet zu ermöglichen. Stellen Sie in letzterem Fall sicher, dass Ihre VPC über ein Subnetz verfügt, das jeder Availability Zone der Region zugeordnet ist, in der Sie arbeiten. Mindestens zwei öffentliche Subnetze müssen verfügbar sein, um den Load Balancer zu erstellen.

Der nächste Teil dieser Vorlage spezifiziert die `Resources`. In diesem Abschnitt werden die Stack-Ressourcen und ihre Eigenschaften angegeben.

AWS::EC2::SecurityGroupRessource `ELBSecurityGroup`

- `SecurityGroupIngress`enthält eine TCP-Eingangsregel, die den Zugriff von allen IP-Adressen (“ CidrIp ": „0.0.0.0/0") auf Port 80 ermöglicht.

AWS::EC2::SecurityGroupRessource `EC2SecurityGroup`

- `SecurityGroupIngress` enthält zwei Eingangsregeln: 1) eine TCP-Eingangsregel, die den SSH-Zugriff (Port 22) von dem IP-Adressbereich aus erlaubt, den Sie für den `SSHLocation`-Eingabeparameter angeben, und 2) eine TCP-Eingangsregel, die den Zugriff vom Load Balancer aus erlaubt, indem Sie die Sicherheitsgruppe des Load Balancers angeben. Die GetAttFunktion wird verwendet, um die ID der Sicherheitsgruppe mit dem logischen Namen abzurufen`ELBSecurityGroup`.

AWS::ElasticLoadBalancingV2::TargetGroupRessource `EC2TargetGroup`

- `PortProtocol`, und `HealthCheckProtocol` geben Sie den EC2 Instance-Port (80) und das Protokoll (HTTP) an, zu denen der `ApplicationLoadBalancer` Traffic weitergeleitet wird und das Elastic Load Balancing verwendet, um den Zustand der EC2 Instances zu überprüfen.
- `HealthCheckIntervalSeconds`gibt an, dass die EC2 Instances zwischen den Zustandsprüfungen einen Abstand von 30 Sekunden haben. `HealthCheckTimeoutSeconds`

ist definiert als die Zeitdauer, die Elastic Load Balancing auf eine Reaktion vom Ziel der Zustandsprüfung wartet (in diesem Beispiel 15 Sekunden). Nach Ablauf des Timeouts markiert Elastic Load Balancing die Zustandsprüfung dieser EC2 Instance als fehlerhaft. Wenn eine EC2 Instance drei aufeinanderfolgende Zustandsprüfungen (`UnhealthyThresholdCount`) nicht besteht, stoppt Elastic Load Balancing die Weiterleitung des Datenverkehrs an diese EC2 Instance, bis diese Instance fünf aufeinanderfolgende Integritätsprüfungen (`HealthyThresholdCount`) durchgeführt hat. Zu diesem Zeitpunkt betrachtet Elastic Load Balancing die Instance als fehlerfrei und beginnt erneut, Datenverkehr an die Instance weiterzuleiten.

- `TargetGroupAttributes` aktualisiert den Wert der Deregistrierungsverzögerung der Zielgruppe auf 20 Sekunden. Standardmäßig wartet Elastic Load Balancing 300 Sekunden, bevor der Abmeldevorgang abgeschlossen wird.

AWS::ElasticLoadBalancingV2::ListenerRessource `ALBListener`

- `DefaultActions` gibt den Port an, an dem Load Balancer abhört, die Zielgruppe, an die Load Balancer Anforderungen weiterleitet, sowie das Protokoll, das zum Weiterleiten von Anforderungen verwendet wird.

AWS::ElasticLoadBalancingV2::LoadBalancerRessource `ApplicationLoadBalancer`

- `Subnets` verwendet den Wert des `Subnets`-Eingabeparameters als Liste der öffentlichen Subnetze, in denen die Load-Balancer-Knoten erstellt werden.
- `SecurityGroup` ruft die ID der Sicherheitsgruppe ab, die als virtuelle Firewall für Ihre Load-Balancer-Knoten dient, um den eingehendem Datenverkehr zu steuern. Die GetAttFunktion wird verwendet, um die ID der Sicherheitsgruppe mit dem logischen Namen abzurufen`ELBSecurityGroup`.

AWS::EC2::LaunchTemplateRessource `LaunchTemplate`

- `ImageId` verwendet den Wert des `LatestAmiId`-Eingabeparameters als das zu verwendende AMI.
- `KeyName`verwendet den Wert des `KeyName` Eingabeparameters als das zu verwendende EC2 key pair.
- `SecurityGroupIds`ruft die ID der Sicherheitsgruppe mit dem logischen Namen ab`EC2SecurityGroup`, die als virtuelle Firewall für Ihre EC2 Instances zur Steuerung des eingehenden Datenverkehrs fungiert.

- `UserData` ist ein Konfigurationsskript, das ausgeführt wird, nachdem die Instance gestartet wurde. In diesem Beispiel installiert das Skript Apache und erstellt eine index.html-Datei.

AWS::SNS::TopicRessource `NotificationTopic`

- `Subscription` verwendet den Wert des `OperatorEmail`-Eingabeparameters als E-Mail-Adresse für den Empfänger der Benachrichtigungen, wenn Skalierungsaktivitäten stattfinden.

AWS::AutoScaling::AutoScalingRessource gruppieren `WebServerGroup`

- `MinSize`und `MaxSize` legen Sie die minimale und maximale Anzahl von EC2 Instances in der Auto Scaling Scaling-Gruppe fest.
- `TargetGroupARNs` nimmt den ARN der Zielgruppe mit dem logischen Namen `EC2TargetGroup`. Bei der Skalierung dieser Auto-Scaling-Gruppe werden Instances bei dieser Zielgruppe automatisch registriert und deregistriert.
- `VPCZoneIdentifier`verwendet den Wert des `Subnets` Eingabeparameters als Liste der öffentlichen Subnetze, in denen die EC2 Instances erstellt werden können.

Schritt 1: Starten des Stacks

Bevor Sie den Stack starten, überprüfen Sie, ob Sie über AWS Identity and Access Management (IAM-) Berechtigungen zur Nutzung aller der folgenden Dienste verfügen: Amazon EC2, Amazon EC2 Auto Scaling AWS Systems Manager, Elastic Load Balancing, Amazon SNS und. AWS CloudFormation

Das folgende Verfahren beinhaltet das Hochladen der Beispiel-Stack-Vorlage aus einer Datei. Öffnen Sie einen Texteditor auf Ihrem lokalen Computer und fügen Sie eine der Vorlagen hinzu. Speichern Sie die Datei mit dem Namen `sampleloadbalancedappstack.template`.

So starten Sie die Stack-Vorlage

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com /cloudformation.
2. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
3. Unter Vorlage festlegen, wählen Sie Vorlagendatei hochladen aus und wählen Sie dann Datei wählen, um `sampleloadbalancedappstack.template` hochzuladen.

4. Wählen Sie Weiter aus.
5. Geben Sie auf der Seite Details angeben einen Namen für den Stack ein (beispielsweise **`SampleLoadBalancedAppStack`**.)
6. Überprüfen Sie unter Parameter die Parameter für den Stack und geben Sie Werte für alle Parameter an, die keine Standardwerte haben, einschließlich OperatorEmail, SSHLocation, KeyName, VPC und Subnetze.
7. Klicken Sie zweimal auf Weiter.
8. Überprüfen und bestätigen Sie die Einstellungen auf der Seite Review.
9. Wählen Sie Absenden aus.

   Sie können den Status des Stacks in der AWS CloudFormation Konsole in der Spalte Status einsehen. Wenn AWS CloudFormation der Stack erfolgreich erstellt wurde, erhalten Sie den Status CREATE_COMPLETE.

   > **Note**
   >
   > Nachdem Sie den Stack erstellt haben,müssen Sie das Abonnement bestätigen, bevor die E-Mail-Adresse Benachrichtigungen erleiden kann. Weitere Informationen finden Sie unter Amazon SNS-Benachrichtigungen erhalten, wenn Ihre Auto Scaling-Gruppe skaliert wird im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Schritt 2: Bereinigen von Beispielressourcen

Um sicherzustellen, dass Ihnen keine ungenutzten Beispielressourcen berechnet werden, löschen Sie den Stack.

So löschen Sie den Stack

1. Wählen Sie in der AWS CloudFormation Konsole den SampleLoadBalancedAppStackStack aus.
2. Wählen Sie Löschen aus.
3. Wählen Sie in der Bestätigungsmitteilung die Option Stack löschen aus.

   Der Status für SampleLoadBalancedAppStackÄnderungen an DELETE_IN_PROGRESS. Wenn das Löschen des Stacks AWS CloudFormation abgeschlossen ist, wird der Stapel aus der Liste entfernt.

Verwenden Sie diese Beispielvorlage aus dieser Anleitung, um eigene Stacks zu erstellen. Weitere Informationen finden Sie unter Tutorial: Eine skalierte Anwendung mit Lastenausgleich einrichten im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Stellen Sie Anwendungen auf Amazon bereit EC2

Sie können CloudFormation es verwenden, um Anwendungen auf EC2 Amazon-Instances automatisch zu installieren, zu konfigurieren und zu starten. Auf diese Weise können Sie auf einfache Weise Bereitstellungen übermitteln und vorhandene Installationen aktualisieren, ohne eine direkte Verbindung mit der Instance herzustellen, was Ihnen viel Zeit und Aufwand sparen kann.

CloudFormation enthält eine Reihe von Hilfsskripten (`cfn-initcfn-signal`,`cfn-get-metadata`, und`cfn-hup`), die auf basieren`cloud-init`. Sie rufen diese Hilfsskripte aus Ihren CloudFormation Vorlagen auf, um Anwendungen auf EC2 Amazon-Instances zu installieren, zu konfigurieren und zu aktualisieren, die sich in derselben Vorlage befinden. Weitere Informationen finden Sie in der Referenz zu CloudFormation Hilfsskripten im AWS CloudFormation Template Reference Guide.

Im Tutorial „Erste Schritte“ haben Sie `UserData` mithilfe eines einfachen Bash-Skripts einen einfachen Webserver erstellt. Das hat zwar für eine einfache „Hello World“ -Seite funktioniert, aber echte Anwendungen benötigen oft eine anspruchsvollere Konfiguration, darunter:

- Es wurden mehrere Softwarepakete in der richtigen Reihenfolge installiert.
- Komplexe Konfigurationsdateien, die mit bestimmten Inhalten erstellt wurden.
- Dienste wurden gestartet und so konfiguriert, dass sie automatisch ausgeführt werden.
- Fehlerbehandlung und Validierung des Einrichtungsvorgangs.

CloudFormationDie Hilfsskripte bieten eine robustere und wartungsfreundlichere Methode zur Konfiguration von EC2 Instanzen als grundlegende Bash-Skripte in. `UserData` Das `cfn-init` Hilfsskript liest Konfigurationsdaten aus den Metadaten Ihrer Vorlage und wendet sie systematisch auf Ihre Instanz an.

In diesem Tutorial erfahren Sie, wie Sie das `cfn-init` Hilfsskript verwenden und den Bootstrapping-Prozess überwachen.

> **Note**
>
> CloudFormation ist kostenlos, aber die EC2 Amazon-Ressourcen, die Sie erstellen, werden Ihnen in Rechnung gestellt. Wenn Sie jedoch noch keine Erfahrung damit haben AWS,

können Sie das kostenlose Kontingent nutzen, um die Kosten während dieses Lernprozesses zu minimieren oder ganz zu vermeiden.

Themen

- Voraussetzungen
- Bootstrap-Konzepte verstehen
- Beginnen Sie mit einem einfachen Bootstrap-Beispiel
- Hinzufügen von Dateien und Befehlen
- Netzwerksicherheit hinzufügen
- Die komplette Bootstrap-Vorlage
- Erstellen Sie den Stack mit der Konsole
- Überwachen Sie den Bootstrap-Prozess
- Testen Sie den Webserver mit Bootstrapping
- Behebung von Bootstrap-Problemen
- Bereinigen von -Ressourcen
- Nächste Schritte

Voraussetzungen

- Sie müssen das Erstelle deinen ersten Stack Tutorial abgeschlossen haben oder über gleichwertige Erfahrung mit den CloudFormation Grundlagen verfügen.
- Sie benötigen Zugriff auf einen Benutzer oder eine Rolle AWS-Konto mit einem IAM-Status, der über Berechtigungen zur Nutzung von Amazon EC2 verfügt CloudFormation, und/oder Administratorzugriff.
- Sie benötigen eine Virtual Private Cloud (VPC), die Zugriff auf das Internet hat. Für diese Tutorial-Vorlage ist eine Standard-VPC erforderlich, die automatisch mit neueren Versionen AWS-Konten geliefert wird. Wenn Sie keine Standard-VPC haben oder diese gelöscht wurde, finden Sie im Abschnitt zur Fehlerbehebung im Erstelle deinen ersten Stack Tutorial alternative Lösungen.

Bootstrap-Konzepte verstehen

Lassen Sie uns die wichtigsten Konzepte verstehen, die dafür sorgen, dass Bootstrapping funktioniert, bevor wir die Vorlage erstellen.

Das **`cfn-init`**-Hilfsskript

CloudFormation stellt Python-Hilfsskripte bereit, mit denen Sie Software installieren und Dienste auf einer EC2 Amazon-Instance starten können. Das `cfn-init` Skript liest Ressourcenmetadaten aus Ihrer Vorlage und wendet die Konfiguration auf Ihre Instance an.

Der Prozess funktioniert wie folgt:

1. Sie definieren die Konfiguration im `Metadata` Abschnitt Ihrer EC2 Ressource.
2. Sie rufen `cfn-init` vom `UserData` Skript aus an.
3. `cfn-init`liest die Metadaten und wendet die Konfiguration an.
4. Ihre Instanz ist gemäß Ihren Spezifikationen konfiguriert.

Struktur der Metadaten

Die Konfiguration ist in einer bestimmten Struktur innerhalb Ihrer EC2 Instanz definiert.

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:                       # Metadata section for the resource
      AWS::CloudFormation::Init:    # Required key that cfn-init looks for
        config:                     # Configuration name (you can have multiple)
          packages:                 # Install packages
          files:                    # Create files
          commands:                 # Run commands
          services:                 # Start/stop services
```

Das `cfn-init` Skript verarbeitet diese Abschnitte in einer bestimmten Reihenfolge: Pakete, Gruppen, Benutzer, Quellen, Dateien, Befehle und dann Dienste.

Beginnen Sie mit einem einfachen Bootstrap-Beispiel

Beginnen wir mit einem minimalen Bootstrap-Beispiel, das Apache einfach installiert und startet.

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
```

```
        config:
          packages:                 # Install Apache web server
            yum:
              httpd: []
          services:                 # Start Apache and enable it to start on boot
            sysvinit:
              httpd:
                enabled: true
                ensureRunning: true
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      UserData: !Base64             # Script that runs when instance starts
        Fn::Sub: |
          #!/bin/bash
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource EC2Instance --
region ${AWS::Region}
```

Dieses einfache Beispiel demonstriert die Kernkonzepte:

- `packages`Abschnitt installiert das `httpd` Paket mityum. Dies funktioniert auf Amazon Linux und anderen Linux-Distributionen, die verwendenyum.
- `services`Dieser Abschnitt stellt sicher, dass er automatisch `httpd` gestartet und ausgeführt wird.
- `UserData`installiert die neuesten Bootstrap-Tools und Aufrufe`cfn-init`.

Hinzufügen von Dateien und Befehlen

Lassen Sie uns nun unser Beispiel erweitern, indem wir eine benutzerdefinierte Webseite und eine Protokolldatei im `/var/log` Verzeichnis auf der EC2 Instanz hinzufügen.

Dateien erstellen

In `files` diesem Abschnitt können Sie Dateien mit bestimmten Inhalten auf der Instanz erstellen. Die vertikale Pipe (`|`) ermöglicht es Ihnen, einen wörtlichen Textblock (HTML-Code) als Inhalt der Datei (`/var/www/html/index.html`) zu übergeben.

```
files:
  /var/www/html/index.html:
    content: |
      <body>
```

```
        <h1>Congratulations, you have successfully launched the AWS CloudFormation
 sample.</h1>
      </body>
```

Ausführen von Befehlen

`commands`In diesem Abschnitt können Sie Shell-Befehle während des Bootstrap-Vorgangs ausführen. Dieser Befehl erstellt eine Protokolldatei `/var/log/welcome.txt` auf der EC2 Instanz. Um es anzuzeigen, benötigen Sie ein EC2 Amazon-Schlüsselpaar, das Sie für den SSH-Zugriff verwenden können, und einen IP-Adressbereich, der für die SSH-Verbindung zur Instance verwendet werden kann (hier nicht behandelt).

```
commands:
  createWelcomeLog:
    command: "echo 'cfn-init ran successfully!' > /var/log/welcome.txt"
```

Netzwerksicherheit hinzufügen

Da wir einen Webserver einrichten, müssen wir zulassen, dass Web-Traffic (HTTP) unsere EC2 Instanz erreicht. Zu diesem Zweck erstellen wir eine Sicherheitsgruppe, die eingehenden Datenverkehr über Port 80 von Ihrer IP-Adresse aus zulässt. EC2 Instanzen müssen außerdem Datenverkehr ins Internet senden, um beispielsweise Paket-Updates zu installieren. Standardmäßig lassen Sicherheitsgruppen den gesamten ausgehenden Datenverkehr zu. Anschließend verknüpfen wir diese Sicherheitsgruppe mithilfe der `SecurityGroupIds` Eigenschaft mit unserer EC2 Instance.

```
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow HTTP access from my IP address
    SecurityGroupIngress:
      - IpProtocol: tcp
        Description: HTTP
        FromPort: 80
        ToPort: 80
        CidrIp: !Ref MyIP
```

Die komplette Bootstrap-Vorlage

Lassen Sie uns nun alle Teile zusammenfügen. Hier ist die vollständige Vorlage, die alle Konzepte kombiniert, die wir besprochen haben.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Bootstrap an EC2 instance with Apache web server using cfn-init

Parameters:
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'

  InstanceType:
    Description: EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t3.micro
      - t2.micro
    ConstraintDescription: must be a valid EC2 instance type.

  MyIP:
    Description: Your IP address in CIDR format (e.g. 203.0.113.1/32)
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\/\d{1,2}$'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:
  WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow HTTP access from my IP address
      SecurityGroupIngress:
        - IpProtocol: tcp
          Description: HTTP
          FromPort: 80
          ToPort: 80
          CidrIp: !Ref MyIP

  WebServer:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
```

```
        config:
          packages:
            yum:
              httpd: []
          files:
            /var/www/html/index.html:
              content: |
                <body>
                  <h1>Congratulations, you have successfully launched the AWS
 CloudFormation sample.</h1>
                </body>
          commands:
            createWelcomeLog:
              command: "echo 'cfn-init ran successfully!' > /var/log/welcome.txt"
          services:
            sysvinit:
              httpd:
                enabled: true
                ensureRunning: true
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      SecurityGroupIds:
        - !Ref WebServerSecurityGroup
      UserData: !Base64
        Fn::Sub: |
          #!/bin/bash
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServer --
region ${AWS::Region}
      Tags:
        - Key: Name
          Value: Bootstrap Tutorial Web Server

Outputs:
  WebsiteURL:
    Value: !Sub 'http://${WebServer.PublicDnsName}'
    Description: EC2 instance public DNS name
```

Erstellen Sie den Stack mit der Konsole

Das folgende Verfahren beinhaltet das Hochladen der Beispiel-Stack-Vorlage aus einer Datei. Öffnen Sie einen Texteditor auf Ihrem lokalen Computer und fügen Sie die Vorlage hinzu. Speichern Sie die Datei mit dem Namen `samplelinux2stack.template`.

So starten Sie die Stack-Vorlage

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
3. Unter Vorlage festlegen, wählen Sie Vorlagendatei hochladen aus und wählen Sie dann Datei wählen, um `samplelinux2stack.template` hochzuladen.
4. Wählen Sie Weiter aus.
5. Geben **`BootstrapTutorialStack`** Sie auf der Seite „Stack-Details angeben“ den Stack-Namen ein.
6. Gehen Sie unter Parameter wie folgt vor.
   - LatestAmiId: Behalten Sie den Standardwert bei.
   - InstanceType: Wählen Sie entweder t2.micro oder t3.micro als Instance-Typ. EC2
   - MyIP: Geben Sie Ihre öffentliche IP-Adresse mit einem `/32` Suffix ein.
7. Wählen Sie zweimal Weiter und dann Senden, um den Stack zu erstellen.

Überwachen Sie den Bootstrap-Prozess

Bootstrap-Prozesse dauern länger als einfache EC2 Starts, da zusätzliche Software installiert und konfiguriert wird.

Um den Bootstrap-Fortschritt zu überwachen

1. Wählen Sie in der CloudFormation Konsole Ihren Stack aus und öffnen Sie den Tab Ereignisse.
2. Halte Ausschau nach dem `WebServer CREATE_IN_PROGRESS` Ereignis. Der Bootstrap-Prozess beginnt, nachdem die Instance gestartet wurde.
3. Der Bootstrap-Vorgang dauert in der Regel einige Minuten. Sie werden sehen`WebServer CREATE_COMPLETE`, wann es fertig ist.

Wenn Sie sehen möchten, was während des Bootstrap-Vorgangs passiert, können Sie die Instanzprotokolle überprüfen.

Um Bootstrap-Logs einzusehen (optional)

1. Öffnen Sie die EC2 Konsole und suchen Sie nach Ihrer Instanz.
2. Wählen Sie die Instanz aus und wählen Sie dann Aktionen, Überwachen und Problembehandlung sowie Systemprotokoll abrufen, um den Bootstrap-Fortschritt zu überprüfen.
3. Wenn Sie die Protokolle nicht sofort sehen, warten Sie und aktualisieren Sie die Seite.

Testen Sie den Webserver mit Bootstrapping

Wenn Ihr Stack angezeigt wird`CREATE_COMPLETE`, testen Sie Ihren Webserver.

Um den Webserver zu testen

1. Gehen Sie in der CloudFormation Konsole zur Registerkarte Ausgaben für Ihren Stack.
2. Klicken Sie auf den Wert WebsiteURL, um Ihren Webserver in einem neuen Tab zu öffnen.
3. Sie sollten Ihre benutzerdefinierte Webseite mit der Nachricht sehen. `Congratulations, you have successfully launched the AWS CloudFormation sample`

> **Note**
>
> Wenn die Seite nicht sofort geladen wird, warten Sie eine Minute und versuchen Sie es erneut. Der Bootstrap-Vorgang ist möglicherweise noch abgeschlossen, auch wenn der Stack angezeigt wird`CREATE_COMPLETE`.

Behebung von Bootstrap-Problemen

Wenn Ihr Bootstrap-Prozess fehlschlägt oder Ihr Webserver nicht funktioniert, finden Sie hier häufig auftretende Probleme und Lösungen.

Häufige Probleme

- Die Stack-Erstellung schlägt fehl — Suchen Sie auf der Registerkarte Ereignisse nach bestimmten Fehlermeldungen.

- Auf den Webserver kann nicht zugegriffen werden — Stellen Sie sicher, dass Ihre IP-Adresse im `MyIP` Parameter korrekt ist. Denken Sie daran, `/32` am Ende anzugeben.
- Der Bootstrap-Prozess schlägt fehl — Die Instanz wird möglicherweise gestartet, `cfn-init` schlägt aber fehl. Überprüfen Sie die Systemprotokolle, wie im Abschnitt Überwachung beschrieben.

Bereinigen von -Ressourcen

Um fortlaufende Gebühren zu vermeiden, können Sie den Stack und seine Ressourcen löschen, um zu bereinigen.

Löschen des Stacks und der entsprechenden Ressourcen

1. Öffnen Sie die CloudFormation-Konsole.
2. Wählen Sie auf der Seite Stacks die Option neben dem Namen des von Ihnen erstellten Stacks (**`BootstrapTutorialStack`**) aus und klicken Sie dann auf Löschen.
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).
4. Überwachen Sie den Fortschritt des Stack-Löschvorgangs auf der Registerkarte Ereignis. Der Status für **`BootstrapTutorialStack`** wechselt zu `DELETE_IN_PROGRESS`. Wenn das Löschen des Stacks CloudFormation abgeschlossen ist, wird der Stapel aus der Liste entfernt.

Nächste Schritte

Herzlichen Glückwunsch! Sie haben erfolgreich gelernt, EC2 Instances mit CloudFormation zu booten. Sie verstehen jetzt:

- Wie benutzt man `cfn-init` Hilfsskripte
- Wie strukturiert man Metadaten für Bootstrapping
- Wie installiert man Pakete, erstellt Dateien, führt Befehle aus und verwaltet Dienste
- Wie kann ich nach Bootstrap-Problemen suchen

Um weiter zu lernen:

- Erfahren Sie, wie Sie einen Windows Stack booten. Weitere Informationen finden Sie unter Bootstrapping-basierte Stacks Windows CloudFormation .

- Erkunden Sie komplexere Bootstrap-Szenarien mit mehreren Konfigurationssätzen. Weitere Informationen finden Sie unter cfn-init und AWS::CloudFormation::Initim Template Reference Guide.AWS CloudFormation
- Erfahren Sie mehr darüber`cfn-signal`, wie Sie den Status des Bootstrap-Abschlusses melden können. Weitere Informationen finden Sie unter cfn-signal im AWS CloudFormation Template Reference Guide.

Aktualisieren eines Stacks

Mit CloudFormation können Sie die Eigenschaften für Ressourcen in Ihren vorhandenen Stacks aktualisieren. Diese Änderungen können von einfachen Konfigurationsänderungen wie der Aktualisierung des Alarmschwellenwerts bei einem CloudWatch Alarm bis hin zu komplexeren Änderungen wie der Aktualisierung des Amazon Machine Image (AMI) reichen, das auf einer EC2 Amazon-Instance ausgeführt wird. Viele der AWS Ressourcen in einer Vorlage können aktualisiert werden, und wir fügen weiterhin Unterstützung für weitere hinzu.

Dieser Abschnitt führt Sie durch eine einfache Verarbeitung von Aktualisierungen für einen ausgeführten Stack. Es zeigt, wie die Verwendung von Vorlagen es ermöglicht, ein Versionskontrollsystem für die Konfiguration Ihrer AWS Infrastruktur zu verwenden, genauso wie Sie die Versionskontrolle für die Software verwenden, die Sie ausführen. Wir führen Sie durch die folgenden Schritte:

1. Erstellen des anfänglichen Stack – Erstellung eines Stack mit einem grundlegenden Amazon Linux AMI sowie Installation des Apache Web Server und einer einfachen PHP-Anwendung unter Verwendung der CloudFormation-Hilfsskripte.
2. Aktualisieren der Anwendung— aktualisieren Sie eine der Dateien in der Anwendung und stellen Sie die Software mithilfe von bereit CloudFormation.
3. Aktualisieren des Instance-Typs— ändert den Instance-Typ der zugrunde liegenden EC2 Amazon-Instance.
4. Aktualisieren Sie das AMI auf einer EC2 Amazon-Instance— ändern Sie das Amazon Machine Image (AMI) für die EC2 Amazon-Instance in Ihrem Stack.
5. Hinzufügen eines Schlüsselpaares zu einer Instance— fügen Sie der Instance ein EC2 Amazon-Schlüsselpaar hinzu und aktualisieren Sie dann die Sicherheitsgruppe, um SSH-Zugriff auf die Instance zu ermöglichen.

6. Ändern der Stack-Ressourcen – Hinzufügen und Entfernen von Instances zum und aus dem Stack sowie Umwandlung in eine automatisch skalierte und ausgeglichene Anwendung durch Aktualisierung der Vorlage.

Eine einfache Anwendung

Wir beginnen mit der Erstellung eines Stack, den wir für den Rest dieses Abschnitts verwenden können. Wir habe eine Beispielvorlage bereitgestellt, mit der eine PHP-Webanwendung mit einzelner Instance auf einem Apache Web Server gehostet und auf einer Amazon Linux AMI ausgeführt wird.

Apache Web Server, PHP und die einfache PHP-Anwendung wurden mithilfe der CloudFormation -Hilfsskripts installiert, die standardmäßig auf der Amazon Linux AMI installiert sind. Der folgenden Vorlagenausschnitt zeigt die Metadaten, die die zu installierenden Pakete und Dateien beschreiben, in diesem Fall den Apache Web Server und die PHP-Infrastruktur aus dem Yum-Repository für die Amazon Linux AMI. Der Ausschnitt zeigt auch den Abschnitt „Services", der gewährleistet, dass der Apache Web Server ausgeführt wird. Im Abschnitt Eigenschaften der EC2 Amazon-Instance-Definition enthält die UserData Eigenschaft das CloudInit Skript, das cfn-init aufruft, um die Pakete und Dateien zu installieren.

```
"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",
  "Metadata" : {
    "AWS::CloudFormation::Init" : {
      "config" : {
        "packages" : {
          "yum" : {
            "httpd"             : [],
            "php"               : []
          }
        },

        "files" : {

          "/var/www/html/index.php" : {
            "content" : { "Fn::Join" : ["", [
              "<?php\n",
              "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",
              "echo '<p>", { "Ref" : "WelcomeMessage" }, "</p>';\n",
              "?>\n"
            ]]},
```

```
              "mode"    : "000644",
              "owner"   : "apache",
              "group"   : "apache"
            },
          },

          :

          "services" : {
            "sysvinit" : {
              "httpd"    : { "enabled" : "true", "ensureRunning" : "true" }
            }
          }
        }
      }
    },

    "Properties": {
      :
      "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
        "#!/bin/bash\n",
        "yum install -y aws-cfn-bootstrap\n",

        :

        "# Install the files and packages from the metadata\n",
        "/opt/aws/bin/cfn-init -v ",
        "         --stack ", { "Ref" : "AWS::StackName" },
        "         --resource WebServerInstance ",
        "         --region ", { "Ref" : "AWS::Region" }, "\n",
        :
      ]]}}
    }
  },
```

Die Anwendung selbst ist ein zweizeiliges Beispiel mit „Hello World”, das innerhalb der Vorlage vollständig definiert ist. Bei einer realen Anwendung können die Dateien auf Amazon S3 oder einem anderen Repository gespeichert und in der Vorlage referenziert werden. GitHub CloudFormation kann Pakete (wie RPMs oder RubyGems) herunterladen und auf einzelne Dateien verweisen und Dateien erweitern `.zip`, um die Anwendungsartefakte auf der EC2 Amazon-Instance zu erstellen. `.tar`

Die Vorlage aktiviert und konfiguriert den cfn-hup-Daemon so, dass er auf Änderungen an der in den Metadaten für die Amazon-Instance definierten Konfiguration wartet. EC2 Mithilfe des cfn-hup-

Daemons können Sie Anwendungssoftware aktualisieren, z. B. die Version von Apache oder PHP, oder Sie können die PHP-Anwendungsdatei selbst von aktualisieren. CloudFormation Der folgende Ausschnitt aus derselben EC2 Amazon-Ressource in der Vorlage zeigt die Teile, die erforderlich sind, um cfn-hup so zu konfigurieren, dass cfn-init aufgerufen wird, um die Software zu aktualisieren, falls Änderungen an den Metadaten erkannt werden:

```
  "WebServerInstance": {
    "Type" : "AWS::EC2::Instance",
    "Metadata" : {
      "AWS::CloudFormation::Init" : {
        "config" : {

            :

          "files" : {

            :

            "/etc/cfn/cfn-hup.conf" : {
              "content" : { "Fn::Join" : ["", [
                "[main]\n",
                "stack=", { "Ref" : "AWS::StackName" }, "\n",
                "region=", { "Ref" : "AWS::Region" }, "\n"
              ]]},
              "mode"    : "000400",
              "owner"   : "root",
              "group"   : "root"
            },

            "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
              "content": { "Fn::Join" : ["", [
                "[cfn-auto-reloader-hook]\n",
                "triggers=post.update\n",
                "path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
\n",
                "action=/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackId" }, " -r
 WebServerInstance ",
                " --region     ", { "Ref" : "AWS::Region" }, "\n",
                "runas=root\n"
              ]]}
            }
          },
```

```
          :
    },

    "Properties": {

         :

      "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [

        :

        "# Start up the cfn-hup daemon to listen for changes to the Web Server metadata
\n",
        "/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'\n",

        :
      ]]}}
    }
  },
```

Um den Stack zu vervollständigen, erstellt die Vorlage eine EC2 Amazon-Sicherheitsgruppe.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template: Sample template that can be used
 to test EC2 updates. **WARNING** This template creates an Amazon Ec2 Instance. You
 will be billed for the AWS resources used if you create a stack from this template.",

  "Parameters" : {

    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "t2.small",
      "AllowedValues" : [
        "t1.micro",
        "t2.nano",
        "t2.micro",
        "t2.small",
        "t2.medium",
        "t2.large",
        "m1.small",
```

```
        "m1.medium",
        "m1.large",
        "m1.xlarge",
        "m2.xlarge",
        "m2.2xlarge",
        "m2.4xlarge",
        "m3.medium",
        "m3.large",
        "m3.xlarge",
        "m3.2xlarge",
        "m4.large",
        "m4.xlarge",
        "m4.2xlarge",
        "m4.4xlarge",
        "m4.10xlarge",
        "c1.medium",
        "c1.xlarge",
        "c3.large",
        "c3.xlarge",
        "c3.2xlarge",
        "c3.4xlarge",
        "c3.8xlarge",
        "c4.large",
        "c4.xlarge",
        "c4.2xlarge",
        "c4.4xlarge",
        "c4.8xlarge",
        "g2.2xlarge",
        "g2.8xlarge",
        "r3.large",
        "r3.xlarge",
        "r3.2xlarge",
        "r3.4xlarge",
        "r3.8xlarge",
        "i2.xlarge",
        "i2.2xlarge",
        "i2.4xlarge",
        "i2.8xlarge",
        "d2.xlarge",
        "d2.2xlarge",
        "d2.4xlarge",
        "d2.8xlarge",
        "hi1.4xlarge",
        "hs1.8xlarge",
```

```
        "cr1.8xlarge",
        "cc2.8xlarge",
        "cg1.4xlarge"
      ],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "HVM64"  },
      "t2.nano"     : { "Arch" : "HVM64"  },
      "t2.micro"    : { "Arch" : "HVM64"  },
      "t2.small"    : { "Arch" : "HVM64"  },
      "t2.medium"   : { "Arch" : "HVM64"  },
      "t2.large"    : { "Arch" : "HVM64"  },
      "m1.small"    : { "Arch" : "HVM64"  },
      "m1.medium"   : { "Arch" : "HVM64"  },
      "m1.large"    : { "Arch" : "HVM64"  },
      "m1.xlarge"   : { "Arch" : "HVM64"  },
      "m2.xlarge"   : { "Arch" : "HVM64"  },
      "m2.2xlarge"  : { "Arch" : "HVM64"  },
      "m2.4xlarge"  : { "Arch" : "HVM64"  },
      "m3.medium"   : { "Arch" : "HVM64"  },
      "m3.large"    : { "Arch" : "HVM64"  },
      "m3.xlarge"   : { "Arch" : "HVM64"  },
      "m3.2xlarge"  : { "Arch" : "HVM64"  },
      "m4.large"    : { "Arch" : "HVM64"  },
      "m4.xlarge"   : { "Arch" : "HVM64"  },
      "m4.2xlarge"  : { "Arch" : "HVM64"  },
      "m4.4xlarge"  : { "Arch" : "HVM64"  },
      "m4.10xlarge" : { "Arch" : "HVM64"  },
      "c1.medium"   : { "Arch" : "HVM64"  },
      "c1.xlarge"   : { "Arch" : "HVM64"  },
      "c3.large"    : { "Arch" : "HVM64"  },
      "c3.xlarge"   : { "Arch" : "HVM64"  },
      "c3.2xlarge"  : { "Arch" : "HVM64"  },
      "c3.4xlarge"  : { "Arch" : "HVM64"  },
      "c3.8xlarge"  : { "Arch" : "HVM64"  },
      "c4.large"    : { "Arch" : "HVM64"  },
      "c4.xlarge"   : { "Arch" : "HVM64"  },
      "c4.2xlarge"  : { "Arch" : "HVM64"  },
      "c4.4xlarge"  : { "Arch" : "HVM64"  },
      "c4.8xlarge"  : { "Arch" : "HVM64"  },
```

```
      "g2.2xlarge"  : { "Arch" : "HVMG2"  },
      "g2.8xlarge"  : { "Arch" : "HVMG2"  },
      "r3.large"    : { "Arch" : "HVM64"  },
      "r3.xlarge"   : { "Arch" : "HVM64"  },
      "r3.2xlarge"  : { "Arch" : "HVM64"  },
      "r3.4xlarge"  : { "Arch" : "HVM64"  },
      "r3.8xlarge"  : { "Arch" : "HVM64"  },
      "i2.xlarge"   : { "Arch" : "HVM64"  },
      "i2.2xlarge"  : { "Arch" : "HVM64"  },
      "i2.4xlarge"  : { "Arch" : "HVM64"  },
      "i2.8xlarge"  : { "Arch" : "HVM64"  },
      "d2.xlarge"   : { "Arch" : "HVM64"  },
      "d2.2xlarge"  : { "Arch" : "HVM64"  },
      "d2.4xlarge"  : { "Arch" : "HVM64"  },
      "d2.8xlarge"  : { "Arch" : "HVM64"  },
      "hi1.4xlarge" : { "Arch" : "HVM64"  },
      "hs1.8xlarge" : { "Arch" : "HVM64"  },
      "cr1.8xlarge" : { "Arch" : "HVM64"  },
      "cc2.8xlarge" : { "Arch" : "HVM64"  }
    },

    "AWSRegionArch2AMI" : {
      "us-east-1"        : {"HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
 "ami-0a584ac55a7631c0c"},
      "us-west-2"        : {"HVM64" : "ami-a0cfeed8", "HVMG2" :
 "ami-0e09505bc235aa82d"},
      "us-west-1"        : {"HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
 "ami-066ee5fd4a9ef77f1"},
      "eu-west-1"        : {"HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
 "ami-0a7c483d527806435"},
      "eu-west-2"        : {"HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},
      "eu-west-3"        : {"HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
 "NOT_SUPPORTED"},
      "eu-central-1"     : {"HVM64" : "ami-0233214e13e500f77", "HVMG2" :
 "ami-06223d46a6d0661c7"},
      "ap-northeast-1"   : {"HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
 "ami-053cdd503598e4a9d"},
      "ap-northeast-2"   : {"HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
 "NOT_SUPPORTED"},
      "ap-northeast-3"   : {"HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
 "NOT_SUPPORTED"},
      "ap-southeast-1"   : {"HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
 "ami-0be9df32ae9f92309"},
```

```
      "ap-southeast-2"   : {"HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
 "ami-0a9ce9fecc3d1daf8"},
      "ap-south-1"       : {"HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
 "ami-097b15e89dbdcfcf4"},
      "us-east-2"        : {"HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
 "NOT_SUPPORTED"},
      "ca-central-1"     : {"HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
      "sa-east-1"        : {"HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
 "NOT_SUPPORTED"},
      "cn-north-1"       : {"HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
 "NOT_SUPPORTED"},
      "cn-northwest-1"   : {"HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
    }
  },

  "Resources" : {

    "WebServerInstance": {
      "Type" : "AWS::EC2::Instance",
      "Metadata" : {
        "Comment" : "Install a simple PHP application",
        "AWS::CloudFormation::Init" : {
          "config" : {
            "packages" : {
              "yum" : {
                "httpd"             : [],
                "php"               : []
              }
            },

            "files" : {

              "/var/www/html/index.php" : {
                "content" : { "Fn::Join" : ["", [
                  "<?php\n",
                  "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",
                  "?>\n"
                ]]},
                "mode"    : "000644",
                "owner"   : "apache",
                "group"   : "apache"
              },
```

```
              "/etc/cfn/cfn-hup.conf" : {
                "content" : { "Fn::Join" : ["", [
                  "[main]\n",
                  "stack=", { "Ref" : "AWS::StackId" }, "\n",
                  "region=", { "Ref" : "AWS::Region" }, "\n"
                ]]},
                "mode"    : "000400",
                "owner"   : "root",
                "group"   : "root"
              },

              "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
                "content": { "Fn::Join" : ["", [
                  "[cfn-auto-reloader-hook]\n",
                  "triggers=post.update\n",
                  "path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
\n",
                  "action=/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackId" }, " -r
 WebServerInstance ",
                                                        " --region     ", { "Ref" :
 "AWS::Region" }, "\n",
                  "runas=root\n"
                ]]}
              }
            },

            "services" : {
              "sysvinit" : {
                "httpd"    : { "enabled" : "true", "ensureRunning" : "true" },
                "cfn-hup" : { "enabled" : "true", "ensureRunning" : "true",
                    "files" : ["/etc/cfn/cfn-hup.conf", "/etc/cfn/hooks.d/cfn-auto-
reloader.conf"]}
              }
            }
          }
        }
      },

      "Properties": {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
 "AWS::Region" },
                          { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" :
 "InstanceType" }, "Arch" ] } ] },
        "InstanceType"   : { "Ref" : "InstanceType" },
```

```
        "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
        "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
             "#!/bin/bash -xe\n",
             "yum install -y aws-cfn-bootstrap\n",

             "# Install the files and packages from the metadata\n",
             "/opt/aws/bin/cfn-init -v ",
             "         --stack ", { "Ref" : "AWS::StackName" },
             "         --resource WebServerInstance ",
             "         --region ", { "Ref" : "AWS::Region" }, "\n",

             "# Start up the cfn-hup daemon to listen for changes to the Web Server
metadata\n",
             "/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'\n",

             "# Signal the status from cfn-init\n",
             "/opt/aws/bin/cfn-signal -e $? ",
             "         --stack ", { "Ref" : "AWS::StackName" },
             "         --resource WebServerInstance ",
             "         --region ", { "Ref" : "AWS::Region" }, "\n"
        ]]}}
      },
      "CreationPolicy" : {
        "ResourceSignal" : {
          "Timeout" : "PT5M"
        }
      }
    },

    "WebServerSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" :
"0.0.0.0/0"}
        ]
      }
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Description" : "Application URL",
```

```
      "Value" : { "Fn::Join" : [", ["http://", { "Fn::GetAtt" : [ "WebServerInstance",
 "PublicDnsName" ]}]] }
    }
  }
}
```

In diesem Beispiel wird eine einzelne EC2 Amazon-Instance verwendet, aber Sie können dieselben Mechanismen auch für komplexere Lösungen verwenden, die Elastic Load Balancing und Amazon EC2 Auto Scaling Gruppen verwenden, um eine Sammlung von Anwendungsservern zu verwalten. Es gibt jedoch einige spezielle Überlegungen für Auto-Scaling-Gruppen. Weitere Informationen finden Sie unter Aktualisieren von Auto-Scaling-Gruppen.

Erstellen des anfänglichen Stack

In diesem Beispiel verwenden wir die, AWS Management Console um einen ersten Stack aus der Beispielvorlage zu erstellen.

> **Warning**
>
> Wenn Sie dieses Verfahren abschließen, werden AWS Live-Dienste bereitgestellt. Ihnen werden die Standard-Nutzungspreise berechnet, solange diese Services ausgeführt werden.

So erstellen Sie den Stack aus AWS Management Console

1. Kopieren Sie die vorherige Vorlage und speichern Sie sie lokal auf Ihrem System als Textdatei. Notieren Sie sich den Speicherort, denn Sie benötigen die Datei in einem späteren Schritt.
2. Melden Sie sich bei der CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation an.
3. Wählen Sie Create New Stack.
4. Geben Sie im Assistenten Create New Stack auf dem Bildschirm Select Template **`UpdateTutorial`** in das Feld Name ein. Wählen Sie auf derselben Seite Upload a template to Amazon S3 (Eine Vorlage in Amazon S3 hochladen) aus, navigieren Sie zu der Datei, die Sie im ersten Schritt heruntergeladen haben, und klicken Sie dann auf Next (Weiter).
5. Geben Sie auf dem Bildschirm Specify Parameters in das Feld Instance Type **`t1.micro`** ein. Wählen Sie anschließend Weiter.
6. Wählen Sie auf dem Bildschirm Options (Optionen) die Option Next (Weiter) aus.

7. Überprüfen Sie auf dem Bildschirm Review (Überprüfen) die eingegebenen Einstellungen und wählen Sie anschließend Create (Erstellen) aus.

Nachdem der Status Ihres Stack zu CREATE_COMPLETE geändert wurde, zeigt die Ausgabe-Registerkarte die URL Ihrer Website an. Wenn Sie den Wert der WebsiteURL-Ausgabe auswählen, sehen Sie Ihre neue PHP-Anwendung im Einsatz.

Aktualisieren der Anwendung

Jetzt, da wir den Stack bereitgestellt haben, wollen wir die Anwendung aktualisieren. Wir nehmen eine einfache Änderung am Text vor, der durch die Anwendung ausgedruckt wird. Dazu fügen wir wie in diesem Vorlagenausschnitt dargestellt einen Echo-Befehl in die index.php-Datei ein:

```
"WebServerInstance": {
      "Type" : "AWS::EC2::Instance",
      "Metadata" : {
        "AWS::CloudFormation::Init" : {
          "config" : {
              :

            "files" : {

              "/var/www/html/index.php" : {
                "content" : { "Fn::Join" : ["", [
                  "<?php\n",
                  "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",
                  "echo '<p>Updated version via UpdateStack</p>';\n ",
                  "?>\n"
                ]]},
                "mode"    : "000644",
                "owner"   : "apache",
                "group"   : "apache"
              },

              :

      }
    },
```

Verwenden Sie einen Text-Editor, um die Vorlagendatei, die Sie lokal gespeichert haben, zu bearbeiten.

Nun aktualisieren wir den Stack.

Um den Stack von der zu aktualisieren AWS Management Console

1. Melden Sie sich bei der CloudFormation Konsole an unter: https://console.aws.amazon.com/cloudformation.
2. Wählen Sie im CloudFormation Dashboard den Stack aus, den Sie zuvor erstellt haben, und wählen Sie dann Stack aktualisieren aus.
3. Wählen Sie im Assistenten Update Stack (Stack aktualisieren) auf dem Bildschirm Select Template (Vorlage auswählen) Upload a template to Amazon S3 (Eine Vorlage in Amazon S3 hochloaden) aus, wählen Sie die geänderte Vorlage aus und dann Next (Weiter).
4. Wählen Sie auf dem Bildschirm Options (Optionen) die Option Next (Weiter) aus.
5. Wählen Sie Next (Weiter) aus, da der Stack über keine Richtlinie verfügt. Alle Ressourcen können ohne überschreibende Richtlinie aktualisiert werden.
6. Überprüfen Sie auf dem Bildschirm Review (Überprüfen) die eingegebenen Einstellungen, und wählen Sie anschließend Update (Aktualisieren) aus.

Wenn Sie den Stack von aus aktualisieren AWS Management Console, werden Sie feststellen, dass die Parameter, die zur Erstellung des ersten Stacks verwendet wurden, auf der Seite „Parameter“ des Assistenten „Stack aktualisieren“ bereits ausgefüllt sind. Wenn Sie den update-stack-Befehl verwenden, geben Sie dieselben Werte für die Parameter ein, die Sie ursprünglich für die Erstellung des Stack verwendet haben.

Sobald Ihr Stapel den Zustand UPDATE_COMPLETE erreicht, können Sie erneut den WebsiteURL-Ausgabewert auswählen und prüfen, ob die Änderungen an Ihrer Anwendung in Kraft getreten sind. Standardmäßig wird die „cfn-hup”-Daemon alle 15 Minuten ausgeführt. Es kann bis zu 15 Minuten bis zur Änderung der Anwendung dauern, nachdem der Stapel aktualisiert wurde.

Um zu sehen, welche Ressourcen aktualisiert wurden, rufen Sie die CloudFormation Konsole auf. Betrachten Sie auf der Registerkarte Events die Stack-Ereignisse. In diesem speziellen Fall wurden die Metadaten für die EC2 Amazon-Instance WebServerInstance aktualisiert, was CloudFormation dazu führte, dass auch die anderen Ressourcen (`WebServerSecurityGroup`) neu bewertet wurden, um sicherzustellen, dass keine weiteren Änderungen vorgenommen wurden. Keine der anderen Stack-Ressourcen wurde geändert. CloudFormation aktualisiert nur die Ressourcen im Stack, die von Änderungen am Stack betroffen sind. Solche Änderungen können direkt sein, wie etwa Eigenschafts- oder Metadatenänderungen, oder sie können auf Abhängigkeiten oder Datenflüsse

durch Ref GetAtt, oder andere systeminterne Vorlagenfunktionen zurückzuführen sein. Weitere Informationen finden Sie unter Referenz zu systeminternen Funktionen.

Dieses einfache Update veranschaulicht den Vorgang. Sie können jedoch viel komplexere Änderungen an den Dateien und Paketen vornehmen, die auf Ihren EC2 Amazon-Instances bereitgestellt werden. Sie können der Instance beispielsweise eine MySQL Instance und den PHP-Support für MySQL hinzufügen. Hierzu fügen Sie einfach die zusätzlichen Pakete und Dateien sowie die zusätzlichen Services für die Konfiguration ein und aktualisieren den Stack, um die Änderungen bereitzustellen. Im folgenden Vorlagenausschnitt sind die Änderungen rot hervorgehoben:

```
    "WebServerInstance": {
      "Type" : "AWS::EC2::Instance",
      "Metadata" : {
        "Comment" : "Install a simple PHP application",
        "AWS::CloudFormation::Init" : {
          "config" : {
            "packages" : {
              "yum" : {
                "httpd"             : [],
                "php"               : [],
                "php-mysql"         : [],
                "mysql-server"      : [],
                "mysql-libs"        : [],
                "mysql"             : []
              }
            },

            :

            "services" : {
              "sysvinit" : {
                "httpd"    : { "enabled" : "true", "ensureRunning" : "true" },
                "cfn-hup" : { "enabled" : "true", "ensureRunning" : "true",
                    "files" : ["/etc/cfn/cfn-hup.conf", "/etc/cfn/hooks.d/cfn-auto-
reloader.conf"]},
                "mysqld"   : { "enabled" : "true", "ensureRunning" : "true" }
              }
            }
          }
        }
      },

      "Properties": {
```

```
          :
      }
    }
```

Sie können die CloudFormation Metadaten aktualisieren, um auf neue Versionen der von der Anwendung verwendeten Pakete zu aktualisieren. In den vorherigen Beispielen ist die Versionseigenschaft der einzelnen Pakete leer. Dies bedeutet, dass „cfn-init" die neueste Version des Pakets installieren sollte.

```
     "packages" : {
       "yum" : {
         "httpd"             : [],
         "php"               : []
      }
```

Sie können optional eine Versionszeichenfolge für ein Paket festlegen. Wenn Sie die Versionszeichenfolge nachfolgenden Stack-Aktualisierungsaufrufen ändern, wird die neue Version des Pakets bereitgestellt. Hier ist ein Beispiel für die Verwendung von Versionsnummern für RubyGems Pakete. Alle Pakete, die Versioning unterstützen, können bestimmte Versionen haben.

```
  "packages" : {
    "rubygems" : {
      "mysql"           : [],
      "rubygems-update" : ["1.6.2"],
      "rake"            : ["0.8.7"],
      "rails"           : ["2.3.11"]
    }
    }
```

Aktualisieren von Auto-Scaling-Gruppen

Wenn Sie in Ihrer Vorlage Auto Scaling Scaling-Gruppen und nicht EC2 Amazon-Instance-Ressourcen verwenden, funktioniert die Aktualisierung der Anwendung genauso, bietet jedoch CloudFormation keine Synchronisation oder Serialisierung zwischen den EC2 Amazon-Instances in einer Auto Scaling Scaling-Gruppe. Der „cfn-hup"-Daemon auf den einzelnen Hosts wird unabhängig ausgeführt und aktualisiert die Anwendung nach einem eigenen Zeitplan. Wenn Sie „cfn-hup" verwenden, um die Konfiguration einer Instance zu aktualisieren, führt jede Instance die „cfn-hup"-Hooks nach einem eigenen Zeitplan aus; es gibt dann keine Koordination zwischen den Instances im Stack. Beachten Sie Folgendes:

- Wenn die cfn-hup-Änderungen auf allen EC2 Amazon-Instances in der Auto Scaling Scaling-Gruppe gleichzeitig ausgeführt werden, ist Ihr Service während des Updates möglicherweise nicht verfügbar.
- Wenn die „cfn-hup”-Änderungen zu verschiedenen Zeiten ausgeführt werden, werden möglicherweise alte und neue Versionen der Software gleichzeitig ausgeführt.

Um diese Probleme zu vermeiden, sollten Sie ein fortlaufendes Update für Ihre Instances in einer Auto-Scaling-Gruppe erzwingen. Weitere Informationen finden Sie unter UpdatePolicy-Attribut.

Ändern von Ressourceneigenschaften

Mit CloudFormation können Sie die Eigenschaften einer vorhandenen Ressource im Stack ändern. In den folgenden Abschnitten werden die verschiedenen Aktualisierungen beschrieben, die spezifische Probleme lösen. Es kann jedoch jede Eigenschaft jeder Ressource nach Bedarf geändert werden, die eine Aktualisierung im Stack unterstützt.

Aktualisieren des Instance-Typs

Der Stack, den wir bisher erstellt haben, verwendet eine EC2 t1.micro-Amazon-Instance. Nehmen wir an, dass Ihre neu erstellte Website mehr Traffic erhält, als eine t1.micro-Instance verarbeiten kann, und Sie möchten jetzt zu einem Amazon-Instance-Typ m1.small wechseln. EC2 Wenn sich die Architektur des Instance-Typs ändert, wird die Instance mit einer anderen AMI erstellt. Wenn Sie sich die Zuordnungen in der Vorlage ansehen, werden Sie feststellen, dass sowohl t1.micro als auch m1.small dieselben Architekturen haben und dasselbe Amazon Linux verwenden. AMIs

```
"Mappings" : {
  "AWSInstanceType2Arch" : {
    "t1.micro"    : { "Arch" : "HVM64"  },
    "t2.nano"     : { "Arch" : "HVM64"  },
    "t2.micro"    : { "Arch" : "HVM64"  },
    "t2.small"    : { "Arch" : "HVM64"  },
    "t2.medium"   : { "Arch" : "HVM64"  },
    "t2.large"    : { "Arch" : "HVM64"  },
    "m1.small"    : { "Arch" : "HVM64"  },
    "m1.medium"   : { "Arch" : "HVM64"  },
    "m1.large"    : { "Arch" : "HVM64"  },
    "m1.xlarge"   : { "Arch" : "HVM64"  },
    "m2.xlarge"   : { "Arch" : "HVM64"  },
    "m2.2xlarge"  : { "Arch" : "HVM64"  },
    "m2.4xlarge"  : { "Arch" : "HVM64"  },
```

```
      "m3.medium"   : { "Arch" : "HVM64"  },
      "m3.large"    : { "Arch" : "HVM64"  },
      "m3.xlarge"   : { "Arch" : "HVM64"  },
      "m3.2xlarge"  : { "Arch" : "HVM64"  },
      "m4.large"    : { "Arch" : "HVM64"  },
      "m4.xlarge"   : { "Arch" : "HVM64"  },
      "m4.2xlarge"  : { "Arch" : "HVM64"  },
      "m4.4xlarge"  : { "Arch" : "HVM64"  },
      "m4.10xlarge" : { "Arch" : "HVM64"  },
      "c1.medium"   : { "Arch" : "HVM64"  },
      "c1.xlarge"   : { "Arch" : "HVM64"  },
      "c3.large"    : { "Arch" : "HVM64"  },
      "c3.xlarge"   : { "Arch" : "HVM64"  },
      "c3.2xlarge"  : { "Arch" : "HVM64"  },
      "c3.4xlarge"  : { "Arch" : "HVM64"  },
      "c3.8xlarge"  : { "Arch" : "HVM64"  },
      "c4.large"    : { "Arch" : "HVM64"  },
      "c4.xlarge"   : { "Arch" : "HVM64"  },
      "c4.2xlarge"  : { "Arch" : "HVM64"  },
      "c4.4xlarge"  : { "Arch" : "HVM64"  },
      "c4.8xlarge"  : { "Arch" : "HVM64"  },
      "g2.2xlarge"  : { "Arch" : "HVMG2"  },
      "g2.8xlarge"  : { "Arch" : "HVMG2"  },
      "r3.large"    : { "Arch" : "HVM64"  },
      "r3.xlarge"   : { "Arch" : "HVM64"  },
      "r3.2xlarge"  : { "Arch" : "HVM64"  },
      "r3.4xlarge"  : { "Arch" : "HVM64"  },
      "r3.8xlarge"  : { "Arch" : "HVM64"  },
      "i2.xlarge"   : { "Arch" : "HVM64"  },
      "i2.2xlarge"  : { "Arch" : "HVM64"  },
      "i2.4xlarge"  : { "Arch" : "HVM64"  },
      "i2.8xlarge"  : { "Arch" : "HVM64"  },
      "d2.xlarge"   : { "Arch" : "HVM64"  },
      "d2.2xlarge"  : { "Arch" : "HVM64"  },
      "d2.4xlarge"  : { "Arch" : "HVM64"  },
      "d2.8xlarge"  : { "Arch" : "HVM64"  },
      "hi1.4xlarge" : { "Arch" : "HVM64"  },
      "hs1.8xlarge" : { "Arch" : "HVM64"  },
      "cr1.8xlarge" : { "Arch" : "HVM64"  },
      "cc2.8xlarge" : { "Arch" : "HVM64"  }
    },
    "AWSRegionArch2AMI" : {
      "us-east-1"        : {"HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
 "ami-0a584ac55a7631c0c"},
```

```
      "us-west-2"        : {"HVM64" : "ami-a0cfeed8", "HVMG2" :
 "ami-0e09505bc235aa82d"},
      "us-west-1"        : {"HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
 "ami-066ee5fd4a9ef77f1"},
      "eu-west-1"        : {"HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
 "ami-0a7c483d527806435"},
      "eu-west-2"        : {"HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},
      "eu-west-3"        : {"HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
 "NOT_SUPPORTED"},
      "eu-central-1"     : {"HVM64" : "ami-0233214e13e500f77", "HVMG2" :
 "ami-06223d46a6d0661c7"},
      "ap-northeast-1"   : {"HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
 "ami-053cdd503598e4a9d"},
      "ap-northeast-2"   : {"HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
 "NOT_SUPPORTED"},
      "ap-northeast-3"   : {"HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
 "NOT_SUPPORTED"},
      "ap-southeast-1"   : {"HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
 "ami-0be9df32ae9f92309"},
      "ap-southeast-2"   : {"HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
 "ami-0a9ce9fecc3d1daf8"},
      "ap-south-1"       : {"HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
 "ami-097b15e89dbdcfcf4"},
      "us-east-2"        : {"HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
 "NOT_SUPPORTED"},
      "ca-central-1"     : {"HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
      "sa-east-1"        : {"HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
 "NOT_SUPPORTED"},
      "cn-north-1"       : {"HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
 "NOT_SUPPORTED"},
      "cn-northwest-1"   : {"HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
    }
```

Verwenden wir die Vorlage, die wir im vorherigen Schritt geändert haben, um den Instance-Typ zu ändern. Da InstanceType es sich um einen Eingabeparameter für die Vorlage handelte, müssen wir die Vorlage nicht ändern. Wir können den Wert des Parameters im Stack-Update-Assistenten auf der Seite „Parameter angeben“ ändern.

So aktualisieren Sie den Stack aus AWS Management Console

1. Melden Sie sich bei der CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation an.

2. Wählen Sie im CloudFormation Dashboard den Stack aus, den Sie zuvor erstellt haben, und wählen Sie dann Stack aktualisieren aus.
3. Wählen Sie im Assistenten Update Stack (Stack aktualisieren) auf dem Bildschirm Select Template (Vorlage auswählen) Use current template (Aktuelle Vorlage verwenden) aus und dann Next (Weiter).

   Die Seite zum Festlegen der Details wird mit den Parametern angezeigt, die zum Erstellen des anfänglichen Stack verwendet wurden; die Werte sind im Abschnitt Specify Parameters bereits eingetragen.
4. Ändern Sie den Wert des InstanceTypeTextfeldes von `t1.micro` zum`t1.small`. Wählen Sie anschließend Weiter.
5. Wählen Sie auf dem Bildschirm Options (Optionen) die Option Next (Weiter) aus.
6. Wählen Sie Next (Weiter) aus, da der Stack über keine Richtlinie verfügt. Alle Ressourcen können ohne überschreibende Richtlinie aktualisiert werden.
7. Überprüfen Sie auf dem Bildschirm Review (Überprüfen) die eingegebenen Einstellungen, und wählen Sie anschließend Update (Aktualisieren) aus.

Sie können den Instance-Typ einer EBS-gestützten Amazon-Instance dynamisch ändern, indem Sie die EC2 Instance starten und stoppen. CloudFormation versucht, die Änderung zu optimieren, indem der Instance-Typ aktualisiert und die Instance neu gestartet wird, sodass sich die Instance-ID nicht ändert. Wird die Instance jedoch neu gestartet, ändert sich die öffentliche IP-Adresse der Instance. Um sicherzustellen, dass die Elastic IP-Adresse nach der Änderung korrekt gebunden ist, CloudFormation wird auch die Elastic IP-Adresse aktualisiert. Sie können die Änderungen in der CloudFormation Konsole auf der Registerkarte Ereignisse sehen.

Um den Instance-Typ von zu überprüfen AWS Management Console, öffnen Sie die EC2 Amazon-Konsole und suchen Sie dort nach Ihrer Instance.

Aktualisieren Sie das AMI auf einer EC2 Amazon-Instance

Betrachten wir nun, wie wir das Amazon Machine Image (AMI) ändern können, das auf der Instance ausgeführt wird. Wir werden die AMI-Änderung einleiten, indem wir den Stack so aktualisieren, dass er einen neuen EC2 Amazon-Instance-Typ verwendet, z. B. t2.medium, bei dem es sich um einen HVM64 Instance-Typ handelt.

Wir verwenden wie im vorherigen Abschnitt unsere vorhandene Vorlage, um den Instance-Typ mit unserem Beispiel-Stack zu ändern. Ändern Sie im Stack-Aktualisierungsassistenten auf der Seite „Specify Parameters” den Wert des Instance-Typs.

In diesem Fall können wir die Instance nicht einfach starten und beenden, um das AMI zu ändern. CloudFormation betrachtet dies als eine Änderung an einer unveränderlichen Eigenschaft der Ressource. Um eine unveränderliche Eigenschaft zu ändern, CloudFormation muss eine Ersatzressource gestartet werden, in diesem Fall eine neue EC2 Amazon-Instance, auf der das neue AMI ausgeführt wird.

Nachdem die neue Instance ausgeführt wurde, werden die anderen Ressourcen im Stack so CloudFormation aktualisiert, dass sie auf die neue Ressource verweisen. Wenn alle neuen Ressourcen erstellt wurden, wird die alte Ressource gelöscht. Dieser Vorgang wird als bezeichnet`UPDATE_CLEANUP`. Dieses Mal werden Sie feststellen, dass sich Instance-ID und Anwendungs-URL der Instance im Stack infolge der Aktualisierung geändert haben. Die Ereignisse in der Event-Tabelle enthalten die Beschreibung „Requested update has a change to an immutable property and hence creating a new physical resource”, um anzugeben, dass eine Ressource ersetzt wurde.

Wenn in das AMI, das Sie aktualisieren möchten, Anwendungscode geschrieben wurde, können Sie zum Aktualisieren des AMI und zum Laden Ihrer neuen Anwendung denselben Stack-Aktualisierungsmechanismus verwenden.

So aktualisieren Sie das AMI für eine Instance in Ihrem Stack

1. Erstellen Sie Ihre neue Datei AMIs , die Ihre Anwendungs- oder Betriebssystemänderungen enthält. Weitere Informationen finden Sie unter Erstellen eines Amazon EBS-backed AMI im EC2 Amazon-Benutzerhandbuch.
2. Aktualisieren Sie Ihre Vorlage, um das neue AMI zu integrieren IDs.
3. Aktualisieren Sie den Stack, entweder AWS Management Console wie unter beschrieben Aktualisieren der Anwendung oder mithilfe des AWS Befehls update-stack.

Wenn Sie den Stack aktualisieren, wird CloudFormation erkannt, dass sich die AMI-ID geändert hat, und löst dann ein Stack-Update auf die gleiche Weise aus, wie wir das obige initiiert haben.

Aktualisieren Sie die EC2 Amazon-Startkonfiguration für eine Auto Scaling Scaling-Gruppe

Wenn Sie Auto Scaling Scaling-Gruppen anstelle von EC2 Amazon-Instances verwenden, ist der Prozess der Aktualisierung der laufenden Instances etwas anders. Bei Auto Scaling Scaling-

Ressourcen ist die Konfiguration der EC2 Amazon-Instances, z. B. der Instance-Typ oder die AMI-ID, in der Auto Scaling Scaling-Startkonfiguration gekapselt. Sie können Änderungen an der Startkonfiguration auf die gleiche Weise vornehmen, wie wir in den vorherigen Abschnitten Änderungen an den EC2 Amazon-Instance-Ressourcen vorgenommen haben. Eine Änderung der Startkonfiguration hat jedoch keine Auswirkungen auf die laufenden EC2 Amazon-Instances in der Auto Scaling Scaling-Gruppe. Eine aktualisierte Startkonfiguration gilt nur für neue Instances, die nach der Aktualisierung erstellt werden.

Wenn Sie möchten, dass die Änderung für alle Instances in Ihrer Auto-Scaling-Gruppe auf Ihre Startkonfiguration übernommen wird, verwenden Sie ein Aktualisierungsattribut. Weitere Informationen finden Sie unter UpdatePolicy-Attribut.

Hinzufügen von Ressourceneigenschaften

Bis jetzt haben wir das Ändern von vorhandenen Eigenschaften einer Ressource in einer Vorlage angeschaut. Sie können auch Eigenschaften hinzufügen, die ursprünglich nicht in der Vorlage angegeben waren. Um dies zu veranschaulichen, fügen wir einer vorhandenen EC2 Instance ein EC2 Amazon-Schlüsselpaar hinzu und öffnen dann Port 22 in der EC2 Amazon-Sicherheitsgruppe, sodass Sie Secure Shell (SSH) für den Zugriff auf die Instance verwenden können.

Hinzufügen eines Schlüsselpaares zu einer Instance

Um SSH-Zugriff zu einer vorhandenen EC2 Amazon-Instance hinzuzufügen

1. Fügen Sie der Vorlage zwei zusätzliche Parameter hinzu, um den Namen eines vorhandenen EC2 Amazon-Schlüsselpaars und einen SSH-Standort zu übergeben.

```
 "Parameters" : {

   "KeyName" : {
     "Description" : "Name of an existing Amazon EC2 key pair for SSH access",
     "Type": "AWS::EC2::KeyPair::KeyName"
   },
   "SSHLocation" : {
     "Description" : " The IP address range that can be used to SSH to the EC2
instances",
     "Type": "String",
     "MinLength": "9",
     "MaxLength": "18",
     "Default": "0.0.0.0/0",
```

```
      "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\
\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/
x."
    }
    :
  },
```

2. Fügen Sie die KeyName Eigenschaft zur EC2 Amazon-Instance hinzu.

```
          "WebServerInstance": {
          "Type" : "AWS::EC2::Instance",
          :
          "Properties": {
          :
          "KeyName" : { "Ref" : "KeyName" },
          :
          }
          },
```

3. Fügen Sie Port 22 und den SSH-Standort zu den Eingangsregeln für die EC2 Amazon-Sicherheitsgruppe hinzu.

```
    "WebServerSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP and SSH",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" :
 { "Ref" : "SSHLocation"}},
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" :
 "0.0.0.0/0"}
        ]
      }
    },
```

4. Aktualisieren Sie den Stack entweder AWS Management Console wie unter erklärt Aktualisieren der Anwendung oder mithilfe des AWS Befehls. update-stack

Ändern der Stack-Ressourcen

Die Anforderungen der Anwendung können sich im Laufe der Zeit ändern, CloudFormation sodass Sie die Gruppe der Ressourcen, aus denen der Stack besteht, ändern können. Um dies zu veranschaulichen, nehmen wir die einzelne Instance-Anwendung aus Hinzufügen von Ressourceneigenschaften und konvertieren Sie durch Aktualisierung des Stacks in eine automatisch skalierte, lastenverteilte Anwendung.

So erstellen Sie eine einfache, einzelne Instance-PHP-Anwendung mit einer Elastic IP-Adresse. Wir werden die Anwendung nun in eine hochverfügbare, automatisch skalierte und lastverteilte Anwendung verwandeln, indem wir ihre Ressourcen während der Aktualisierung ändern.

1. Fügen Sie eine Elastic Load Balancer-Ressource hinzu.

```
"ElasticLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "CrossZone" : "true",
    "AvailabilityZones" : { "Fn::GetAZs" : "" },
    "LBCookieStickinessPolicy" : [ {
      "PolicyName" : "CookieBasedPolicy",
      "CookieExpirationPeriod" : "30"
    } ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP",
      "PolicyNames" : [ "CookieBasedPolicy" ]
    } ],
    "HealthCheck" : {
      "Target" : "HTTP:80/",
      "HealthyThreshold" : "2",
      "UnhealthyThreshold" : "5",
      "Interval" : "10",
      "Timeout" : "5"
    }
  }
}
```

2. Konvertieren Sie die EC2 Instanz in der Vorlage in eine Auto Scaling Launch Configuration. Die Eigenschaften sind identisch, sodass wir nur die Typennamen ändern müssen:

```
"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",
```

auf:

```
"LaunchConfig": {
  "Type" : "AWS::AutoScaling::LaunchConfiguration",
```

Aus Gründen der Übersichtlichkeit haben wir den Namen der Ressource von WebServerInstancein geändert LaunchConfig, sodass Sie den Ressourcennamen, auf den cfn-init und cfn-hup verweisen, aktualisieren müssen (suchen Sie einfach nach, WebServerInstance und ersetzen Sie ihn durch, außer cfn-signal). LaunchConfig Für cfn-signal müssen Sie die Auto Scaling Scaling-Gruppe (WebServerGroup) signalisieren, nicht die Instanz, wie im folgenden Snippet gezeigt:

```
            "# Signal the status from cfn-init\n",
            "/opt/aws/bin/cfn-signal -e $? ",
            "         --stack ", { "Ref" : "AWS::StackName" },
            "         --resource WebServerGroup ",
            "         --region ", { "Ref" : "AWS::Region" }, "\n"
```

3. Fügen Sie eine Auto-Scaling-Gruppe-Ressource hinzu.

```
    "WebServerGroup" : {
      "Type" : "AWS::AutoScaling::AutoScalingGroup",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
        "MinSize" : "1",
        "DesiredCapacity" : "1",
        "MaxSize" : "5",
        "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
      },
      "CreationPolicy" : {
        "ResourceSignal" : {
          "Timeout" : "PT15M"
        }
      },
```

```
      "UpdatePolicy": {
        "AutoScalingRollingUpdate": {
          "MinInstancesInService": "1",
          "MaxBatchSize": "1",
          "PauseTime" : "PT15M",
          "WaitOnResourceSignals": "true"
        }
      }
    }
```

4. Aktualisieren Sie die Sicherheitsgruppendefinition, um den Datenverkehr vom Load Balancer an die Instances zu sperren.

```
    "WebServerSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80 locked down to the ELB
 and SSH access",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80",
 "SourceSecurityGroupOwnerId" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
 "SourceSecurityGroup.OwnerAlias"]},
"SourceSecurityGroupName" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
 "SourceSecurityGroup.GroupName"]}},
          {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" :
 { "Ref" : "SSHLocation"}}
        ]
      }
    }
```

5. Aktualisieren Sie die Ausgabewerte, um den DNS-Namen des Elastic Load Balancer als Speicherort der Anwendung auszugeben:

```
"WebsiteURL" : {
  "Value" : { "Fn::Join" : ["", ["http://",
      { "Fn::GetAtt" : [ "WebServerInstance", "PublicDnsName" ]}]]},
  "Description" : "Application URL"
}
```

auf:

```
"WebsiteURL" : {
  "Value" : { "Fn::Join" : ["", ["http://",
      { "Fn::GetAtt" : [ "ElasticLoadBalancer", "DNSName" ]}]]},
  "Description" : "Application URL"
}
```

Das folgende Beispiel zeigt die vollständige Vorlage als Referenz. Wenn Sie diese Vorlage verwenden, um den Stack zu aktualisieren, verwandeln Sie Ihre einfache, einzelne Instance-Anwendung in eine hoch verfügbare, Multi-AZ-Anwendung mit automatischer Skalierung und Lastausgleich. Nur die Ressourcen, die aktualisiert werden müssen, werden geändert. Alle weiteren Datenspeicher für diese Anwendung bleiben unverändert. Jetzt können Sie es verwenden, um Ihre Stacks CloudFormation zu erweitern oder zu verbessern, wenn sich Ihre Anforderungen ändern.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template: Sample template that can be used
 to test EC2 updates. **WARNING** This template creates an Amazon Ec2 Instance. You
 will be billed for the AWS resources used if you create a stack from this template.",

  "Parameters" : {

    "KeyName": {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the
 instance",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription" : "must be the name of an existing EC2 KeyPair."
    },

    "SSHLocation" : {
      "Description" : " The IP address range that can be used to SSH to the EC2
 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    },
```

```
    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "t2.small",
      "AllowedValues" : [
        "t1.micro",
        "t2.nano",
        "t2.micro",
        "t2.small",
        "t2.medium",
        "t2.large",
        "m1.small",
        "m1.medium",
        "m1.large",
        "m1.xlarge",
        "m2.xlarge",
        "m2.2xlarge",
        "m2.4xlarge",
        "m3.medium",
        "m3.large",
        "m3.xlarge",
        "m3.2xlarge",
        "m4.large",
        "m4.xlarge",
        "m4.2xlarge",
        "m4.4xlarge",
        "m4.10xlarge",
        "c1.medium",
        "c1.xlarge",
        "c3.large",
        "c3.xlarge",
        "c3.2xlarge",
        "c3.4xlarge",
        "c3.8xlarge",
        "c4.large",
        "c4.xlarge",
        "c4.2xlarge",
        "c4.4xlarge",
        "c4.8xlarge",
        "g2.2xlarge",
        "g2.8xlarge",
        "r3.large",
        "r3.xlarge",
```

```
        "r3.2xlarge",
        "r3.4xlarge",
        "r3.8xlarge",
        "i2.xlarge",
        "i2.2xlarge",
        "i2.4xlarge",
        "i2.8xlarge",
        "d2.xlarge",
        "d2.2xlarge",
        "d2.4xlarge",
        "d2.8xlarge",
        "hi1.4xlarge",
        "hs1.8xlarge",
        "cr1.8xlarge",
        "cc2.8xlarge",
        "cg1.4xlarge"
      ],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "HVM64"  },
      "t2.nano"     : { "Arch" : "HVM64"  },
      "t2.micro"    : { "Arch" : "HVM64"  },
      "t2.small"    : { "Arch" : "HVM64"  },
      "t2.medium"   : { "Arch" : "HVM64"  },
      "t2.large"    : { "Arch" : "HVM64"  },
      "m1.small"    : { "Arch" : "HVM64"  },
      "m1.medium"   : { "Arch" : "HVM64"  },
      "m1.large"    : { "Arch" : "HVM64"  },
      "m1.xlarge"   : { "Arch" : "HVM64"  },
      "m2.xlarge"   : { "Arch" : "HVM64"  },
      "m2.2xlarge"  : { "Arch" : "HVM64"  },
      "m2.4xlarge"  : { "Arch" : "HVM64"  },
      "m3.medium"   : { "Arch" : "HVM64"  },
      "m3.large"    : { "Arch" : "HVM64"  },
      "m3.xlarge"   : { "Arch" : "HVM64"  },
      "m3.2xlarge"  : { "Arch" : "HVM64"  },
      "m4.large"    : { "Arch" : "HVM64"  },
      "m4.xlarge"   : { "Arch" : "HVM64"  },
      "m4.2xlarge"  : { "Arch" : "HVM64"  },
      "m4.4xlarge"  : { "Arch" : "HVM64"  },
```

```
      "m4.10xlarge" : { "Arch" : "HVM64"  },
      "c1.medium"   : { "Arch" : "HVM64"  },
      "c1.xlarge"   : { "Arch" : "HVM64"  },
      "c3.large"    : { "Arch" : "HVM64"  },
      "c3.xlarge"   : { "Arch" : "HVM64"  },
      "c3.2xlarge"  : { "Arch" : "HVM64"  },
      "c3.4xlarge"  : { "Arch" : "HVM64"  },
      "c3.8xlarge"  : { "Arch" : "HVM64"  },
      "c4.large"    : { "Arch" : "HVM64"  },
      "c4.xlarge"   : { "Arch" : "HVM64"  },
      "c4.2xlarge"  : { "Arch" : "HVM64"  },
      "c4.4xlarge"  : { "Arch" : "HVM64"  },
      "c4.8xlarge"  : { "Arch" : "HVM64"  },
      "g2.2xlarge"  : { "Arch" : "HVMG2"  },
      "g2.8xlarge"  : { "Arch" : "HVMG2"  },
      "r3.large"    : { "Arch" : "HVM64"  },
      "r3.xlarge"   : { "Arch" : "HVM64"  },
      "r3.2xlarge"  : { "Arch" : "HVM64"  },
      "r3.4xlarge"  : { "Arch" : "HVM64"  },
      "r3.8xlarge"  : { "Arch" : "HVM64"  },
      "i2.xlarge"   : { "Arch" : "HVM64"  },
      "i2.2xlarge"  : { "Arch" : "HVM64"  },
      "i2.4xlarge"  : { "Arch" : "HVM64"  },
      "i2.8xlarge"  : { "Arch" : "HVM64"  },
      "d2.xlarge"   : { "Arch" : "HVM64"  },
      "d2.2xlarge"  : { "Arch" : "HVM64"  },
      "d2.4xlarge"  : { "Arch" : "HVM64"  },
      "d2.8xlarge"  : { "Arch" : "HVM64"  },
      "hi1.4xlarge" : { "Arch" : "HVM64"  },
      "hs1.8xlarge" : { "Arch" : "HVM64"  },
      "cr1.8xlarge" : { "Arch" : "HVM64"  },
      "cc2.8xlarge" : { "Arch" : "HVM64"  }
    },
    "AWSRegionArch2AMI" : {
      "us-east-1"        : {"HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
 "ami-0a584ac55a7631c0c"},
      "us-west-2"        : {"HVM64" : "ami-a0cfeed8", "HVMG2" :
 "ami-0e09505bc235aa82d"},
      "us-west-1"        : {"HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
 "ami-066ee5fd4a9ef77f1"},
      "eu-west-1"        : {"HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
 "ami-0a7c483d527806435"},
      "eu-west-2"        : {"HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},
```

```
      "eu-west-3"        : {"HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
 "NOT_SUPPORTED"},
      "eu-central-1"     : {"HVM64" : "ami-0233214e13e500f77", "HVMG2" :
 "ami-06223d46a6d0661c7"},
      "ap-northeast-1"   : {"HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
 "ami-053cdd503598e4a9d"},
      "ap-northeast-2"   : {"HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
 "NOT_SUPPORTED"},
      "ap-northeast-3"   : {"HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
 "NOT_SUPPORTED"},
      "ap-southeast-1"   : {"HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
 "ami-0be9df32ae9f92309"},
      "ap-southeast-2"   : {"HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
 "ami-0a9ce9fecc3d1daf8"},
      "ap-south-1"       : {"HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
 "ami-097b15e89dbdcfcf4"},
      "us-east-2"        : {"HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
 "NOT_SUPPORTED"},
      "ca-central-1"     : {"HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
      "sa-east-1"        : {"HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
 "NOT_SUPPORTED"},
      "cn-north-1"       : {"HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
 "NOT_SUPPORTED"},
      "cn-northwest-1"   : {"HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
    }
  },

  "Resources" : {

    "ElasticLoadBalancer" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "CrossZone" : "true",
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "LBCookieStickinessPolicy" : [ {
          "PolicyName" : "CookieBasedPolicy",
          "CookieExpirationPeriod" : "30"
        } ],
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
          "InstancePort" : "80",
          "Protocol" : "HTTP",
          "PolicyNames" : [ "CookieBasedPolicy" ]
        } ],
```

```
        "HealthCheck" : {
          "Target" : "HTTP:80/",
          "HealthyThreshold" : "2",
          "UnhealthyThreshold" : "5",
          "Interval" : "10",
          "Timeout" : "5"
        }
      }
    },

    "WebServerGroup" : {
      "Type" : "AWS::AutoScaling::AutoScalingGroup",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
        "MinSize" : "1",
        "DesiredCapacity" : "1",
        "MaxSize" : "5",
        "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
      },
      "CreationPolicy" : {
        "ResourceSignal" : {
          "Timeout" : "PT15M"
        }
      },
      "UpdatePolicy": {
        "AutoScalingRollingUpdate": {
          "MinInstancesInService": "1",
          "MaxBatchSize": "1",
          "PauseTime" : "PT15M",
          "WaitOnResourceSignals": "true"
        }
      }
    },

    "LaunchConfig": {
      "Type" : "AWS::AutoScaling::LaunchConfiguration",
      "Metadata" : {
        "Comment" : "Install a simple PHP application",
        "AWS::CloudFormation::Init" : {
          "config" : {
            "packages" : {
              "yum" : {
                "httpd"             : [],
```

```
                "php"               : []
              }
            },

            "files" : {

              "/var/www/html/index.php" : {
                "content" : { "Fn::Join" : ["", [
                  "<?php\n",
                  "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",
                  "echo 'Updated version via UpdateStack';\n ",
                  "?>\n"
                ]]},
                "mode"    : "000644",
                "owner"   : "apache",
                "group"   : "apache"
              },


              "/etc/cfn/cfn-hup.conf" : {
                "content" : { "Fn::Join" : ["", [
                  "[main]\n",
                  "stack=", { "Ref" : "AWS::StackId" }, "\n",
                  "region=", { "Ref" : "AWS::Region" }, "\n"
                ]]},
                "mode"    : "000400",
                "owner"   : "root",
                "group"   : "root"
              },

              "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
                "content": { "Fn::Join" : ["", [
                  "[cfn-auto-reloader-hook]\n",
                  "triggers=post.update\n",
                  "path=Resources.LaunchConfig.Metadata.AWS::CloudFormation::Init\n",
                  "action=/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackId" }, " -r
LaunchConfig ",
                                                    " --region     ", { "Ref" :
"AWS::Region" }, "\n",
                  "runas=root\n"
                ]]}
              }
            },
```

```
            "services" : {
              "sysvinit" : {
                "httpd"    : { "enabled" : "true", "ensureRunning" : "true" },
                "cfn-hup" : { "enabled" : "true", "ensureRunning" : "true",
                    "files" : ["/etc/cfn/cfn-hup.conf", "/etc/cfn/hooks.d/cfn-auto-
reloader.conf"]}
              }
            }
          }
        }
      },

      "Properties": {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
 "AWS::Region" },
                          { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" :
 "InstanceType" }, "Arch" ] } ] },
        "InstanceType"   : { "Ref" : "InstanceType" },
        "KeyName"        : { "Ref" : "KeyName" },
        "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
        "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
             "#!/bin/bash -xe\n",
             "yum install -y aws-cfn-bootstrap\n",

             "# Install the files and packages from the metadata\n",
             "/opt/aws/bin/cfn-init -v ",
             "         --stack ", { "Ref" : "AWS::StackName" },
             "         --resource LaunchConfig ",
             "         --region ", { "Ref" : "AWS::Region" }, "\n",

             "# Start up the cfn-hup daemon to listen for changes to the Web Server
 metadata\n",
             "/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'\n",

             "# Signal the status from cfn-init\n",
             "/opt/aws/bin/cfn-signal -e $? ",
             "         --stack ", { "Ref" : "AWS::StackName" },
             "         --resource WebServerGroup ",
             "         --region ", { "Ref" : "AWS::Region" }, "\n"
        ]]}}
      }
    },

    "WebServerSecurityGroup" : {
```

```
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80 locked down to the ELB and
 SSH access",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80",
 "SourceSecurityGroupOwnerId" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
 "SourceSecurityGroup.OwnerAlias"]},"SourceSecurityGroupName" : {"Fn::GetAtt" :
 ["ElasticLoadBalancer", "SourceSecurityGroup.GroupName"]}},
          {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" :
 { "Ref" : "SSHLocation"}}
        ]
      }
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Description" : "Application URL",
      "Value" : { "Fn::Join" : ["", ["http://", { "Fn::GetAtt" :
 [ "ElasticLoadBalancer", "DNSName" ]}]] }
    }
  }
}
```

Verfügbarkeit und Auswirkungen

Verschiedene Eigenschaften haben unterschiedliche Auswirkungen auf die Ressourcen im Stack. Sie können CloudFormation verwenden, um eine beliebige Eigenschaft zu aktualisieren. Bevor Sie jedoch Änderungen vornehmen, sollten Sie sich folgende Fragen stellen:

1. Wie wirkt sich die Aktualisierung auf die Ressource selbst aus? Durch die Aktualisierung eines Schwellenwerts für einen Alarm wird beispielswiese der Alarm während der Aktualisierung inaktiv. Wie wir gesehen haben, erfordert die Änderung des Instance-Typs, dass die Instance gestoppt und neu gestartet wird. CloudFormation verwendet die Aktualisierungs- oder Änderungsaktionen für die zugrunde liegenden Ressourcen, um Änderungen an Ressourcen vorzunehmen. Um die Auswirkung von Updates zu verstehen, sollten Sie die Dokumentation für die spezifischen Ressourcen einsehen.
2. Ist die Änderung veränderlich oder unveränderlich? Einige Änderungen an den Ressourceneigenschaften, wie z. B. das Ändern des AMI auf einer EC2 Amazon-Instance, werden von den zugrunde liegenden Diensten nicht unterstützt. Bei veränderlichen Änderungen

CloudFormation wird der Typ Update oder Modify APIs für die zugrunde liegenden Ressourcen verwendet. Bei unveränderlichen Eigenschaftsänderungen CloudFormation werden neue Ressourcen mit den aktualisierten Eigenschaften erstellt und diese dann mit dem Stack verknüpft, bevor die alten Ressourcen gelöscht werden. Obwohl CloudFormation versucht wird, die Ausfallzeit der Stack-Ressourcen zu reduzieren, ist das Ersetzen einer Ressource ein mehrstufiger Prozess, der einige Zeit in Anspruch nimmt. Während der Stack-Neukonfiguration ist Ihre Anwendung nicht voll betriebsbereit. Beispielsweise kann sie keine Anforderungen erfüllen oder auf eine Datenbank zugreifen.

Zugehörige Ressourcen

Weitere Informationen zur Verwendung CloudFormation zum Starten von Anwendungen und zur Integration mit anderen Konfigurations- und Bereitstellungsdiensten wie Puppet und Opscode Chef finden Sie in den folgenden Whitepapers:

- Bootstrapping von Anwendungen über CloudFormation
- Integration CloudFormation mit Opscode Chef
- Integration CloudFormation mit Puppet

Bei der in diesem Abschnitt verwendeten Vorlage handelt es sich um eine „Hello World”-PHP-Anwendung. Die Vorlagenbibliothek enthält auch eine ElastiCache Amazon-Beispielvorlage, die zeigt, wie eine PHP-Anwendung ElasticCache mithilfe von cfn-hup und cfn-init integriert werden kann, um auf Änderungen in der Amazon ElastiCache Cache Cluster-Konfiguration zu reagieren, die alle von Update Stack ausgeführt werden können.

Führen Sie blue/green ECS-Bereitstellungen durch, indem Sie CodeDeploy CloudFormation

Um eine Anwendung zu aktualisieren, die auf Amazon Elastic Container Service (Amazon ECS) ausgeführt wird, können Sie eine CodeDeploy Blau/Grün-Bereitstellungsstrategie verwenden. Diese Strategie trägt dazu bei, Unterbrechungen zu minimieren, die durch wechselnde Anwendungsversionen verursacht werden.

In einer blue/green Bereitstellung erstellen Sie neben Ihrer aktuellen Live-Umgebung (als blau bezeichnet) eine neue Anwendungsumgebung (als grün bezeichnet). Auf diese Weise können Sie die grüne Umgebung überwachen und testen, bevor Sie Live-Verkehr von der blauen Umgebung in die

grüne Umgebung weiterleiten. Nachdem die grüne Umgebung Live-Verkehr bereitgestellt hat, können Sie die blaue Umgebung problemlos beenden.

Um CodeDeploy blaue/grüne Bereitstellungen auf ECS durchzuführen CloudFormation, fügen Sie Ihrer Stack-Vorlage die folgenden Informationen hinzu:

- Ein `Hooks` Abschnitt, der einen `AWS::CodeDeploy::BlueGreen` Hook beschreibt.
- Ein `Transform` Abschnitt, der die `AWS::CodeDeployBlueGreen` Transformation spezifiziert.

Die folgenden Themen führen Sie durch die Einrichtung einer CloudFormation Vorlage für eine blaue/grüne Bereitstellung auf ECS.

Themen

- Über blue/green Bereitstellungen
- Überlegungen bei der Verwaltung von ECS Blue/Green-Bereitstellungen mit CloudFormation
- AWS::CodeDeploy::BlueGreenHook-Syntax
- Beispiel für eine blaue/grüne Bereitstellungsvorlage

Über blue/green Bereitstellungen

Dieses Thema bietet einen Überblick darüber, wie die Durchführung von blue/green Bereitstellungen mit funktioniert. CloudFormation Außerdem wird erklärt, wie Sie Ihre CloudFormation Vorlage für Blau/Grün-Bereitstellungen vorbereiten.

Themen

- Funktionsweise
- Ressourcen-Updates, die umweltfreundliche Bereitstellungen einleiten
- Vorbereiten der Vorlage für die Durchführung von ECS-Blau/Grün-Bereitstellungen
- Modellieren Sie Ihre blue/green Bereitstellung mithilfe von CloudFormation Ressourcen
- Änderungssätze
- Überwachung von Stack-Ereignissen
- IAM-Berechtigungen für Bereitstellungen blue/green

Funktionsweise

Wenn Sie blue/green ECS-Bereitstellungen über durchführen CodeDeploy, erstellen Sie zunächst eine Stack-Vorlage, die die Ressourcen für Ihre blauen und grünen Anwendungsumgebungen definiert, einschließlich der Angabe der zu CloudFormation verwendenden Einstellungen für das Routing und die Stabilisierung des Datenverkehrs. Als Nächstes erstellen Sie einen Stack aus dieser Vorlage. Dadurch wird Ihre blaue (aktuelle) Anwendung generiert. CloudFormation erstellt nur die blauen Ressourcen während der Stack-Erstellung. Ressourcen für eine grüne Bereitstellung werden erst erstellt, wenn sie erforderlich sind.

Wenn Sie dann bei einem future Stack-Update die Aufgabendefinition oder die Taskset-Ressourcen in Ihrer blauen Anwendung aktualisieren, CloudFormation geht das Folgende vor:

- Generiert alle notwendigen Ressourcen für eine grüne Anwendungsumgebung.
- Verschiebt den Datenverkehr basierend auf den angegebenen Datenverkehrs-Weiterleitungsparametern.
- Löscht die blauen Ressourcen.

Wenn zu einem beliebigen Zeitpunkt ein Fehler auftritt, bevor die grüne Bereitstellung erfolgreich und abgeschlossen ist, führt CloudFormation den Stack in seinen Zustand zurück, bevor die gesamte grüne Bereitstellung initiiert wurde.

Ressourcen-Updates, die umweltfreundliche Bereitstellungen einleiten

Wenn Sie ein Stack-Update durchführen, das bestimmte Eigenschaften bestimmter ECS-Ressourcen aktualisiert, wird ein umweltfreundlicher Bereitstellungsprozess CloudFormation eingeleitet. Die Ressourcen, die diesen Prozess einleiten, sind:

- AWS::ECS::TaskDefinition
- AWS::ECS::TaskSet

Wenn die Aktualisierungen dieser Ressourcen jedoch keine Eigenschaftsänderungen beinhalten, die ersetzt werden müssen, wird keine umweltfreundliche Bereitstellung eingeleitet. Weitere Informationen finden Sie unter Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen.

Es ist wichtig zu beachten, dass Sie Aktualisierungen der oben genannten Ressourcen nicht mit Aktualisierungen anderer Ressourcen im selben Stack-Aktualisierungsvorgang kombinieren können.

Wenn Sie sowohl die aufgelisteten Ressourcen als auch andere Ressourcen innerhalb desselben Stacks aktualisieren müssen, haben Sie zwei Möglichkeiten:

- Führen Sie zwei separate Stack-Aktualisierungsvorgänge durch: einen, die nur die Aktualisierungen der obengenannten Ressourcen enthält, und eine separate Stack-Aktualisierung, die Änderungen an anderen Ressourcen enthält.
- Entfernen Sie die Abschnitte `Hooks` und `Transform` aus Ihrer Vorlage und führen Sie dann die Stack-Aktualisierung durch. In diesem Fall CloudFormation wird keine umweltfreundliche Bereitstellung durchgeführt.

Vorbereiten der Vorlage für die Durchführung von ECS-Blau/Grün-Bereitstellungen

Um blue/green Bereitstellungen auf Ihrem Stack zu aktivieren, nehmen Sie die folgenden Abschnitte in Ihre Stack-Vorlage auf, bevor Sie ein Stack-Update durchführen.

- Fügen Sie Ihrer Vorlage einen Verweis auf die Transformation `AWS::CodeDeployBlueGreen` hinzu:

```
"Transform": [
  "AWS::CodeDeployBlueGreen"
],
```

- Fügen Sie einen `Hooks`-Abschnitt hinzu, der den Hook `AWS::CodeDeploy::BlueGreen` aufruft und die Eigenschaften für Ihre Bereitstellung angibt. Weitere Informationen finden Sie unter `AWS::CodeDeploy::BlueGreenHook-Syntax`.
- Definieren Sie im Abschnitt `Resources` die blauen und grünen Ressourcen für Ihre Bereitstellung.

Sie können diese Abschnitte hinzufügen, wenn Sie die Vorlage zum ersten Mal erstellen (d. h. vor dem Erstellen des Stacks selbst), oder Sie können sie einer vorhandenen Vorlage hinzufügen, bevor Sie eine Stack-Aktualisierung durchführen. Wenn Sie das blue/green Deployment für einen neuen Stack angeben, werden bei der Stack-Erstellung CloudFormation nur die blauen Ressourcen erstellt — Ressourcen für das grüne Deployment werden erst erstellt, wenn sie während eines Stack-Updates benötigt werden.

Modellieren Sie Ihre blue/green Bereitstellung mithilfe von CloudFormation Ressourcen

Um eine CodeDeploy blaue/grüne Bereitstellung auf ECS durchzuführen, muss Ihre CloudFormation Vorlage die Ressourcen enthalten, die Ihre Bereitstellung modellieren, z. B. einen Amazon ECS-

Service und einen Load Balancer. Weitere Informationen dazu, was diese Ressourcen darstellen, finden Sie unter Vor einer Amazon ECS-Bereitstellung im AWS CodeDeploy -Benutzerhandbuch.

Anforderung	Ressource	Erforderlich/optional	Initiiert die blue/green Bereitstellung, falls sie ersetzt wird?
Amazon-ECS-Cluster	AWS::ECS::Cluster	Optional. Der Standard-Cluster kann verwendet werden.	Nein
Amazon-ECS-Service	AWS::ECS::Service	Erforderlich	Nein
Application oder Network Load Balancer	AWS::ECS::Service LoadBalancer	Erforderlich	Nein
Produktions-Listener	AWS::ElasticLoadBalancingV2::Listener	Erforderlich	Nein
Listener testen	AWS::ElasticLoadBalancingV2::Listener	Optional.	Nein
Zwei Zielgruppen	AWS::ElasticLoadBalancingV2::TargetGroup	Erforderlich	Nein
Amazon ECS-Aufgabendefinition	AWS::ECS::TaskDefinition	Erforderlich	Ja
Container für Ihre Amazon ECS-Anwendung	AWS::ECS::TaskDefinition Container Definition Name	Erforderlich	Nein
Ein Port für den Ersatzaufgabensatz	AWS::ECS::TaskDefinition PortMapping ContainerPort	Erforderlich	Nein

Änderungssätze

Es wird dringend empfohlen, einen Änderungssatz zu erstellen, bevor Sie eine Stack-Aktualisierung durchführen, die eine Grün-Bereitstellung initiiert. Auf diese Weise können Sie sehen, welche Änderungen tatsächlich an Ihrem Stack vorgenommen werden, bevor Sie das Stack-Update durchführen. Beachten Sie, dass Ressourcenänderungen möglicherweise nicht in der Reihenfolge aufgeführt werden, in der sie während der Stack-Aktualisierung durchgeführt werden. Weitere Informationen finden Sie unter CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren.

Überwachung von Stack-Ereignissen

Sie können die Stack-Ereignisse, die bei jedem Schritt der ECS-Bereitstellung generiert wurden, in der Registerkarte Ereignisse der Seite Stack und mithilfe der AWS CLI anzeigen. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

IAM-Berechtigungen für Bereitstellungen blue/green

CloudFormation Um die blaugrünen Bereitstellungen erfolgreich durchführen zu können, benötigen Sie die folgenden Berechtigungen: CodeDeploy

- `codedeploy:Get*`
- `codedeploy:CreateCloudFormationDeployment`

Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für CodeDeploy in der Service-Autorisierungs-Referenz.

Überlegungen bei der Verwaltung von ECS Blue/Green-Bereitstellungen mit CloudFormation

Das Verfahren CloudFormation zur Durchführung Ihrer blue/green ECS-Bereitstellungen unterscheidet CodeDeploy sich von einer standardmäßigen ECS-Bereitstellung mit nur. CodeDeploy Ein detailliertes Verständnis dieser Unterschiede finden Sie unter Unterschiede zwischen Amazon blue/green ECS-Bereitstellungen durch CodeDeploy und AWS CloudFormation im AWS CodeDeploy Benutzerhandbuch.

Bei der Verwaltung Ihrer blue/green Bereitstellung mithilfe von CloudFormation sind bestimmte Einschränkungen und Überlegungen zu beachten:

- Nur Updates für bestimmte Ressourcen führen zu einer umweltfreundlichen Bereitstellung. Weitere Informationen finden Sie unter Ressourcen-Updates, die umweltfreundliche Bereitstellungen einleiten.
- Sie können Updates für Ressourcen, die umweltfreundliche Bereitstellungen initiieren, und Updates für andere Ressourcen nicht in dasselbe Stack-Update aufnehmen. Weitere Informationen finden Sie unter Ressourcen-Updates, die umweltfreundliche Bereitstellungen einleiten.
- Sie können nur einen einzelnen ECS-Service als Bereitstellungsziel angeben.
- Parameter, deren Werte verschleiert sind, CloudFormation können CodeDeploy während einer umweltfreundlichen Bereitstellung nicht aktualisiert werden. Dies führt zu einem Fehler und einem Fehler beim Stack-Update. Dazu zählen:
  - Parameter, die mit dem Attribut `NoEcho` definiert sind.
  - Parameter, die dynamische Referenzen verwenden, um ihre Werte von externen Diensten abzurufen. Weitere Hinweise zu dynamischen Verweisen finden Sie unter. Ruft mithilfe dynamischer Verweise Werte ab, die in anderen Diensten gespeichert sind
- Um eine grüne Bereitstellung abzubrechen, die noch im Gange ist, brechen Sie das Stack-Update in CloudFormation, Not CodeDeploy oder ECS ab. Weitere Informationen finden Sie unter Ein Stack-Update abbrechen. Nach Abschluss einer Aktualisierung können Sie diese nicht mehr stornieren. Sie können einen Stack jedoch erneut mit allen vorherigen Einstellungen aktualisieren.
- Die folgenden CloudFormation Funktionen werden derzeit nicht für Vorlagen unterstützt, die blue/green ECS-Bereitstellungen definieren:
  - Ausgaben deklarieren oder Fn:: verwendenImportValue, um Werte aus anderen Stacks zu importieren.
  - Ressourcen importieren. Weitere Informationen zum Importieren von Ressourcen finden Sie unterAWS Ressourcen in einen CloudFormation Stack importieren.
  - Verwenden des `AWS::CodeDeploy::BlueGreen` Hooks in einer Vorlage, die verschachtelte Stack-Ressourcen enthält. Weitere Informationen zu verschachtelten Stacks finden Sie unter. Teilen Sie eine Vorlage mithilfe verschachtelter Stapel in wiederverwendbare Teile auf
  - Verwenden des `AWS::CodeDeploy::BlueGreen` Hooks in einem verschachtelten Stapel.

`AWS::CodeDeploy::BlueGreen`Hook-Syntax

Die folgende Syntax beschreibt die Struktur eines `AWS::CodeDeploy::BlueGreen` Hooks für blue/green ECS-Bereitstellungen.

Syntax

```
"Hooks": {
  "Logical ID": {
    "Type": "AWS::CodeDeploy::BlueGreen",
    "Properties": {
      "TrafficRoutingConfig": {
        "Type": "Traffic routing type",
        "TimeBasedCanary": {
          "StepPercentage": Integer,
          "BakeTimeMins": Integer
        },
        "TimeBasedLinear": {
          "StepPercentage": Integer,
          "BakeTimeMins": Integer
        }
      },
      "AdditionalOptions": {"TerminationWaitTimeInMinutes": Integer},
      "LifecycleEventHooks": {
        "BeforeInstall": "FunctionName",
        "AfterInstall": "FunctionName",
        "AfterAllowTestTraffic": "FunctionName",
        "BeforeAllowTraffic": "FunctionName",
        "AfterAllowTraffic": "FunctionName"
      },
      "ServiceRole": "CodeDeployServiceRoleName",
      "Applications": [
        {
          "Target": {
            "Type": "AWS::ECS::Service",
            "LogicalID": "Logical ID of AWS::ECS::Service"
          },
          "ECSAttributes": {
            "TaskDefinitions": [
              "Logical ID of AWS::ECS::TaskDefinition (Blue)",
              "Logical ID of AWS::ECS::TaskDefinition (Green)"
            ],
            "TaskSets": [
              "Logical ID of AWS::ECS::TaskSet (Blue)",
              "Logical ID of AWS::ECS::TaskSet (Green)"
            ],
            "TrafficRouting": {
              "ProdTrafficRoute": {
                "Type": "AWS::ElasticLoadBalancingV2::Listener",
```

```
                "LogicalID": "Logical ID of AWS::ElasticLoadBalancingV2::Listener
 (Production)"
              },
              "TestTrafficRoute": {
                "Type": "AWS::ElasticLoadBalancingV2::Listener",
                "LogicalID": "Logical ID of AWS::ElasticLoadBalancingV2::Listener
 (Test)"
              },
              "TargetGroups": [
                "Logical ID of AWS::ElasticLoadBalancingV2::TargetGroup (Blue)",
                "Logical ID of AWS::ElasticLoadBalancingV2::TargetGroup (Green)"
              ]
            }
          }
        }
      ]
    }
  }
}
```

Eigenschaften

Logische ID (auch logischer Name genannt)

Die logische ID eines Hooks, der im `Hooks` Abschnitt der Vorlage deklariert wurde. Die logische ID muss alphanumerisch (A-Za-z0-9) und innerhalb der Vorlage eindeutig sein.

Erforderlich: Ja

`Type`

Der Typ des Hooks. `AWS::CodeDeploy::BlueGreen`

Erforderlich: Ja

`Properties`

Eigenschaften des Hooks.

Erforderlich: Ja

`TrafficRoutingConfig`

Konfigurationseinstellungen für Datenverkehrsweiterleitungen

Required: No

Die Standardkonfiguration ist eine zeitbasierte Verschiebung des Canary-Verkehrs mit einem Schrittprozentsatz von 15 % und einer Bake-Zeit von fünf Minuten.

`Type`

Der Typ des Datenverkehrs, der von der Bereitstellungskonfiguration verwendet wird.

Gültige Werte: AllAtOnce | TimeBasedCanary | TimeBasedLinear

Erforderlich: Ja

`TimeBasedCanary`

Gibt eine Konfiguration an, die den Datenverkehr in zwei Schritten von einer Version der Bereitstellung auf eine andere verlagert.

Erforderlich: Bedingt: Wenn Sie `TimeBasedCanary` als Datenverkehrs-Weiterleitungstyp angeben, müssen Sie den Parameter `TimeBasedCanary` einschließen.

`StepPercentage`

Der Prozentsatz des Datenverkehrs, der im ersten Schritt einer `TimeBasedCanary`-Bereitstellung verschoben werden soll. Der Schrittprozentsatz muss 14 % oder höher sein.

Required: No

`BakeTimeMins`

Die Anzahl der Minuten zwischen der ersten und zweiten Verkehrsverschiebung einer `TimeBasedCanary`-Bereitstellung.

Required: No

`TimeBasedLinear`

Gibt eine Konfiguration an, die den Datenverkehr in gleichen Schritten von einer Version der Bereitstellung in ein anderes gleiches Inkrement verlagert, wobei zwischen jedem Inkrement eine gleiche Anzahl von Minuten liegt.

Erforderlich: Bedingt: Wenn Sie `TimeBasedLinear` als Datenverkehrs-Weiterleitungstyp angeben, müssen Sie den Parameter `TimeBasedLinear` einschließen.

`StepPercentage`

Der Prozentsatz des Datenverkehrs, der zu Beginn jedes Inkrements einer `TimeBasedLinear`-Bereitstellung verschoben wird. Der Schrittprozentsatz muss 14 % oder höher sein.

Required: No

`BakeTimeMins`

Die Anzahl der Minuten zwischen jeder inkrementellen Verkehrsverschiebung einer `TimeBasedLinear`-Bereitstellung.

Required: No

`AdditionalOptions`

Zusätzliche Optionen für die blue/green Bereitstellung.

Required: No

`TerminationWaitTimeInMinutes`

Gibt die Wartezeit in Minuten an, bevor die blauen Ressourcen beendet werden.

Required: No

`LifecycleEventHooks`

Verwenden Sie Lifecycle-Event-Hooks, um eine Lambda-Funktion anzugeben, die aufgerufen CodeDeploy werden kann, um eine Bereitstellung zu validieren. Sie können dieselbe Funktion oder eine andere für die Bereitstellungs-Lebenszyklusereignisse verwenden. Nach Abschluss der Validierungstests ruft die `AfterAllowTraffic` Lambda-Funktion zurück CodeDeploy und liefert das Ergebnis `Succeeded` oder`Failed`. Weitere Informationen finden Sie im Abschnitt AppSpec „Hooks“ im AWS CodeDeploy Benutzerhandbuch.

Required: No

`BeforeInstall`

Funktion zur Ausführung von Aufgaben, bevor der neue Aufgabensatz erstellt ist.

Required: No

`AfterInstall`

Funktion zur Ausführung von Aufgaben nach Erstellung des neuen Aufgabensatzes und Zuordnung einer der Zielgruppen.

Required: No

`AfterAllowTestTraffic`

Funktion zur Ausführung von Aufgaben, wenn der Test-Listener Datenverkehr an den neuen Aufgabensatz sendet.

Required: No

`BeforeAllowTraffic`

Funktion zur Ausführung von Aufgaben, wenn die zweite Zielgruppe dem neuen Aufgabensatz zugeordnet ist, aber bevor Datenverkehr an den neuen Aufgabensatz umgeleitet wurde.

Required: No

`AfterAllowTraffic`

Funktion zur Ausführung von Aufgaben, wenn die zweite Zielgruppe Datenverkehr an den neuen Aufgabensatz sendet.

Required: No

`ServiceRole`

Die Ausführungsrolle, die für CloudFormation die Ausführung der blaugrünen Bereitstellungen verwendet werden soll. Eine Liste der erforderlichen Berechtigungen finden Sie unter. IAM-Berechtigungen für Bereitstellungen blue/green

Required: No

`Applications`

Gibt die Eigenschaften der Amazon ECS-Anwendung an.

Erforderlich: Ja

`Target`

Erforderlich: Ja

`Type`

Der Ressourcentyp.

Erforderlich: Ja

`LogicalID`

Die logische ID der Ressource.

Erforderlich: Ja

`ECSAttributes`

Die Ressourcen, die die verschiedenen Anforderungen Ihrer Amazon ECS-Anwendungsbereitstellung darstellen.

Erforderlich: Ja

`TaskDefinitions`

Die logische ID der AWS::ECS::TaskDefinition-Ressource zum Ausführen des Docker-Containers, der Ihre Amazon ECS-Anwendung enthält.

Erforderlich: Ja

`TaskSets`

Die Logik IDs der AWS::ECS::TaskSetRessourcen, die als Tasksets für die Anwendung verwendet werden sollen.

Erforderlich: Ja

`TrafficRouting`

Gibt Ressourcen an, die für das Weiterleiten des Datenverkehrs verwendet werden.

Erforderlich: Ja

`ProdTrafficRoute`

Der Listener, der von Ihrem Load Balancer verwendet wird, um Datenverkehr an Ihre Zielgruppen weiterzuleiten.

Erforderlich: Ja

`Type`

Der Ressourcentyp. `AWS::ElasticLoadBalancingV2::Listener`

Erforderlich: Ja

`LogicalID`

Die logische ID der Ressource.

Erforderlich: Ja

`TestTrafficRoute`

Der Listener, der von Ihrem Load Balancer verwendet wird, um Datenverkehr an Ihre Zielgruppen weiterzuleiten.

Erforderlich: Ja

`Type`

Der Ressourcentyp. `AWS::ElasticLoadBalancingV2::Listener`

Erforderlich: Ja

`LogicalID`

Die logische ID der Ressource.

Required: No

`TargetGroups`

Logische ID der Ressourcen, die als Zielgruppen verwendet werden sollen, um den Datenverkehr an das registrierte Ziel weiterzuleiten.

Erforderlich: Ja

Beispiel für eine blaue/grüne Bereitstellungsvorlage

Die folgende Beispielvorlage richtet eine CodeDeploy blaue/grüne Bereitstellung auf ECS ein, mit einem Fortschritt bei der Weiterleitung des Datenverkehrs von 15 Prozent pro Schritt und einem Stabilisierungszeitraum von 5 Minuten zwischen den einzelnen Schritten.

Durch die Erstellung eines Stacks mit der Vorlage wird die Erstkonfiguration der Bereitstellung bereitgestellt. Wenn Sie dann Änderungen an den Eigenschaften der `BlueTaskSet` Ressource vorgenommen haben, die einen Austausch dieser Ressource erfordern, CloudFormation wird im Rahmen des Stack-Updates eine umweltfreundliche Bereitstellung initiiert.

JSON

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Parameters":{
    "Vpc":{ "Type":"AWS::EC2::VPC::Id" },
    "Subnet1":{ "Type":"AWS::EC2::Subnet::Id" },
    "Subnet2":{ "Type":"AWS::EC2::Subnet::Id" }
  },
  "Transform":[ "AWS::CodeDeployBlueGreen" ],
  "Hooks":{
    "CodeDeployBlueGreenHook":{
      "Type":"AWS::CodeDeploy::BlueGreen",
      "Properties":{
        "TrafficRoutingConfig":{
          "Type":"TimeBasedCanary",
          "TimeBasedCanary":{
            "StepPercentage":15,
            "BakeTimeMins":5
          }
        },
        "Applications":[
          {
            "Target":{
              "Type":"AWS::ECS::Service",
              "LogicalID":"ECSDemoService"
            },
            "ECSAttributes":{
              "TaskDefinitions":[ "BlueTaskDefinition","GreenTaskDefinition" ],
              "TaskSets":[ "BlueTaskSet","GreenTaskSet" ],
              "TrafficRouting":{
                "ProdTrafficRoute":{
                  "Type":"AWS::ElasticLoadBalancingV2::Listener",
                  "LogicalID":"ALBListenerProdTraffic"
                },
                "TargetGroups":[ "ALBTargetGroupBlue","ALBTargetGroupGreen" ]
              }
            }
          }
        ]
      }
    }
  },
  "Resources":{
```

```
    "ExampleSecurityGroup":{
      "Type":"AWS::EC2::SecurityGroup",
      "Properties":{
        "GroupDescription":"Security group for ec2 access",
        "VpcId":{ "Ref":"Vpc" },
        "SecurityGroupIngress":[
          {
            "IpProtocol":"tcp",
            "FromPort":80,
            "ToPort":80,
            "CidrIp":"0.0.0.0/0"
          },
          {
            "IpProtocol":"tcp",
            "FromPort":8080,
            "ToPort":8080,
            "CidrIp":"0.0.0.0/0"
          },
          {
            "IpProtocol":"tcp",
            "FromPort":22,
            "ToPort":22,
            "CidrIp":"0.0.0.0/0"
          }
        ]
      }
    },
    "ALBTargetGroupBlue":{
      "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
      "Properties":{
        "HealthCheckIntervalSeconds":5,
        "HealthCheckPath":"/",
        "HealthCheckPort":"80",
        "HealthCheckProtocol":"HTTP",
        "HealthCheckTimeoutSeconds":2,
        "HealthyThresholdCount":2,
        "Matcher":{ "HttpCode":"200" },
        "Port":80,
        "Protocol":"HTTP",
        "Tags":[{ "Key":"Group","Value":"Example" }],
        "TargetType":"ip",
        "UnhealthyThresholdCount":4,
        "VpcId":{ "Ref":"Vpc" }
      }
```

```
    },
    "ALBTargetGroupGreen":{
      "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
      "Properties":{
        "HealthCheckIntervalSeconds":5,
        "HealthCheckPath":"/",
        "HealthCheckPort":"80",
        "HealthCheckProtocol":"HTTP",
        "HealthCheckTimeoutSeconds":2,
        "HealthyThresholdCount":2,
        "Matcher":{ "HttpCode":"200" },
        "Port":80,
        "Protocol":"HTTP",
        "Tags":[{ "Key":"Group","Value":"Example" }],
        "TargetType":"ip",
        "UnhealthyThresholdCount":4,
        "VpcId":{ "Ref":"Vpc" }
      }
    },
    "ExampleALB":{
      "Type":"AWS::ElasticLoadBalancingV2::LoadBalancer",
      "Properties":{
        "Scheme":"internet-facing",
        "SecurityGroups":[{ "Ref":"ExampleSecurityGroup" }],
        "Subnets":[{ "Ref":"Subnet1" },{ "Ref":"Subnet2" }],
        "Tags":[{ "Key":"Group","Value":"Example" }],
        "Type":"application",
        "IpAddressType":"ipv4"
      }
    },
    "ALBListenerProdTraffic":{
      "Type":"AWS::ElasticLoadBalancingV2::Listener",
      "Properties":{
        "DefaultActions":[
          {
            "Type":"forward",
            "ForwardConfig":{
              "TargetGroups":[
                {
                  "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" },
                  "Weight":1
                }
              ]
            }
```

```
          }
        ],
        "LoadBalancerArn":{ "Ref":"ExampleALB" },
        "Port":80,
        "Protocol":"HTTP"
      }
    },
    "ALBListenerProdRule":{
      "Type":"AWS::ElasticLoadBalancingV2::ListenerRule",
      "Properties":{
        "Actions":[
          {
            "Type":"forward",
            "ForwardConfig":{
              "TargetGroups":[
                {
                  "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" },
                  "Weight":1
                }
              ]
            }
          }
        ],
        "Conditions":[
          {
            "Field":"http-header",
            "HttpHeaderConfig":{
              "HttpHeaderName":"User-Agent",
              "Values":[ "Mozilla" ]
            }
          }
        ],
        "ListenerArn":{ "Ref":"ALBListenerProdTraffic" },
        "Priority":1
      }
    },
    "ECSTaskExecutionRole":{
      "Type":"AWS::IAM::Role",
      "Properties":{
        "AssumeRolePolicyDocument":{
          "Version":"2012-10-17",
          "Statement":[
            {
              "Sid":"",
```

```
              "Effect":"Allow",
              "Principal":{
                "Service":"ecs-tasks.amazonaws.com"
              },
              "Action":"sts:AssumeRole"
            }
          ]
        },
        "ManagedPolicyArns":[ "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy" ]
      }
    },
    "BlueTaskDefinition":{
      "Type":"AWS::ECS::TaskDefinition",
      "Properties":{
        "ExecutionRoleArn":{
          "Fn::GetAtt":[ "ECSTaskExecutionRole","Arn" ]
        },
        "ContainerDefinitions":[
          {
            "Name":"DemoApp",
            "Image":"nginxdemos/hello:latest",
            "Essential":true,
            "PortMappings":[
              {
                "HostPort":80,
                "Protocol":"tcp",
                "ContainerPort":80
              }
            ]
          }
        ],
        "RequiresCompatibilities":[ "FARGATE" ],
        "NetworkMode":"awsvpc",
        "Cpu":"256",
        "Memory":"512",
        "Family":"ecs-demo"
      }
    },
    "ECSDemoCluster":{
      "Type":"AWS::ECS::Cluster",
      "Properties":{}
    },
    "ECSDemoService":{
```

```
      "Type":"AWS::ECS::Service",
      "Properties":{
        "Cluster":{ "Ref":"ECSDemoCluster" },
        "DesiredCount":1,
        "DeploymentController":{ "Type":"EXTERNAL" }
      }
    },
    "BlueTaskSet":{
      "Type":"AWS::ECS::TaskSet",
      "Properties":{
        "Cluster":{ "Ref":"ECSDemoCluster" },
        "LaunchType":"FARGATE",
        "NetworkConfiguration":{
          "AwsVpcConfiguration":{
            "AssignPublicIp":"ENABLED",
            "SecurityGroups":[{ "Ref":"ExampleSecurityGroup" }],
            "Subnets":[{ "Ref":"Subnet1" },{ "Ref":"Subnet2" }]
          }
        },
        "PlatformVersion":"1.4.0",
        "Scale":{
          "Unit":"PERCENT",
          "Value":100
        },
        "Service":{ "Ref":"ECSDemoService"},
        "TaskDefinition":{ "Ref":"BlueTaskDefinition" },
        "LoadBalancers":[
          {
            "ContainerName":"DemoApp",
            "ContainerPort":80,
            "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" }
          }
        ]
      }
    },
    "PrimaryTaskSet":{
      "Type":"AWS::ECS::PrimaryTaskSet",
      "Properties":{
        "Cluster":{ "Ref":"ECSDemoCluster" },
        "Service":{ "Ref":"ECSDemoService" },
        "TaskSetId":{ "Fn::GetAtt":[ "BlueTaskSet","Id" ]
        }
      }
    }
```

```
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  Vpc:
    Type: 'AWS::EC2::VPC::Id'
  Subnet1:
    Type: 'AWS::EC2::Subnet::Id'
  Subnet2:
    Type: 'AWS::EC2::Subnet::Id'
Transform:
  - 'AWS::CodeDeployBlueGreen'
Hooks:
  CodeDeployBlueGreenHook:
    Type: 'AWS::CodeDeploy::BlueGreen'
    Properties:
      TrafficRoutingConfig:
        Type: TimeBasedCanary
        TimeBasedCanary:
          StepPercentage: 15
          BakeTimeMins: 5
      Applications:
        - Target:
            Type: 'AWS::ECS::Service'
            LogicalID: ECSDemoService
          ECSAttributes:
            TaskDefinitions:
              - BlueTaskDefinition
              - GreenTaskDefinition
            TaskSets:
              - BlueTaskSet
              - GreenTaskSet
            TrafficRouting:
              ProdTrafficRoute:
                Type: 'AWS::ElasticLoadBalancingV2::Listener'
                LogicalID: ALBListenerProdTraffic
              TargetGroups:
                - ALBTargetGroupBlue
                - ALBTargetGroupGreen
Resources:
```

```
  ExampleSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Security group for ec2 access
      VpcId: !Ref Vpc
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 8080
          ToPort: 8080
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
  ALBTargetGroupBlue:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      HealthCheckIntervalSeconds: 5
      HealthCheckPath: /
      HealthCheckPort: '80'
      HealthCheckProtocol: HTTP
      HealthCheckTimeoutSeconds: 2
      HealthyThresholdCount: 2
      Matcher:
        HttpCode: '200'
      Port: 80
      Protocol: HTTP
      Tags:
        - Key: Group
          Value: Example
      TargetType: ip
      UnhealthyThresholdCount: 4
      VpcId: !Ref Vpc
  ALBTargetGroupGreen:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      HealthCheckIntervalSeconds: 5
      HealthCheckPath: /
      HealthCheckPort: '80'
      HealthCheckProtocol: HTTP
```

```
      HealthCheckTimeoutSeconds: 2
      HealthyThresholdCount: 2
      Matcher:
        HttpCode: '200'
      Port: 80
      Protocol: HTTP
      Tags:
        - Key: Group
          Value: Example
      TargetType: ip
      UnhealthyThresholdCount: 4
      VpcId: !Ref Vpc
  ExampleALB:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Scheme: internet-facing
      SecurityGroups:
        - !Ref ExampleSecurityGroup
      Subnets:
        - !Ref Subnet1
        - !Ref Subnet2
      Tags:
        - Key: Group
          Value: Example
      Type: application
      IpAddressType: ipv4
  ALBListenerProdTraffic:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      DefaultActions:
        - Type: forward
          ForwardConfig:
            TargetGroups:
              - TargetGroupArn: !Ref ALBTargetGroupBlue
                Weight: 1
      LoadBalancerArn: !Ref ExampleALB
      Port: 80
      Protocol: HTTP
  ALBListenerProdRule:
    Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
    Properties:
      Actions:
        - Type: forward
          ForwardConfig:
```

```
            TargetGroups:
              - TargetGroupArn: !Ref ALBTargetGroupBlue
                Weight: 1
      Conditions:
        - Field: http-header
          HttpHeaderConfig:
            HttpHeaderName: User-Agent
            Values:
              - Mozilla
      ListenerArn: !Ref ALBListenerProdTraffic
      Priority: 1
  ECSTaskExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Sid: ''
            Effect: Allow
            Principal:
              Service: ecs-tasks.amazonaws.com
            Action: 'sts:AssumeRole'
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy'
  BlueTaskDefinition:
    Type: 'AWS::ECS::TaskDefinition'
    Properties:
      ExecutionRoleArn: !GetAtt
        - ECSTaskExecutionRole
        - Arn
      ContainerDefinitions:
        - Name: DemoApp
          Image: 'nginxdemos/hello:latest'
          Essential: true
          PortMappings:
            - HostPort: 80
              Protocol: tcp
              ContainerPort: 80
      RequiresCompatibilities:
        - FARGATE
      NetworkMode: awsvpc
      Cpu: '256'
      Memory: '512'
      Family: ecs-demo
```

```
  ECSDemoCluster:
    Type: 'AWS::ECS::Cluster'
    Properties: {}
  ECSDemoService:
    Type: 'AWS::ECS::Service'
    Properties:
      Cluster: !Ref ECSDemoCluster
      DesiredCount: 1
      DeploymentController:
        Type: EXTERNAL
  BlueTaskSet:
    Type: 'AWS::ECS::TaskSet'
    Properties:
      Cluster: !Ref ECSDemoCluster
      LaunchType: FARGATE
      NetworkConfiguration:
        AwsVpcConfiguration:
          AssignPublicIp: ENABLED
          SecurityGroups:
            - !Ref ExampleSecurityGroup
          Subnets:
            - !Ref Subnet1
            - !Ref Subnet2
      PlatformVersion: 1.4.0
      Scale:
        Unit: PERCENT
        Value: 100
      Service: !Ref ECSDemoService
      TaskDefinition: !Ref BlueTaskDefinition
      LoadBalancers:
        - ContainerName: DemoApp
          ContainerPort: 80
          TargetGroupArn: !Ref ALBTargetGroupBlue
  PrimaryTaskSet:
    Type: 'AWS::ECS::PrimaryTaskSet'
    Properties:
      Cluster: !Ref ECSDemoCluster
      Service: !Ref ECSDemoService
      TaskSetId: !GetAtt
        - BlueTaskSet
        - Id
```

CloudFormation Schnipsel aus Vorlagen

Dieser Abschnitt enthält eine Reihe von Beispielszenarien, die Sie verwenden können, um zu verstehen, wie verschiedene AWS CloudFormation -Vorlagenteile deklariert werden. Sie können die Ausschnitte auch als Ausgangspunkt für Abschnitte von benutzerdefinierten Vorlagen verwenden.

> **Note**
>
> Da AWS CloudFormation Vorlagen JSON-konform sein müssen, ist kein Zeilenfortsetzungszeichen vorgesehen. Der Zeilenumbruch in diesem Dokument kann zufällig erfolgen, wenn die Zeile länger als 68 Zeichen ist.

Themen

- Allgemeine Vorlagenausschnitte
- CloudFormation Vorlagenausschnitte für automatische Skalierung
- AWS Vorlagenausschnitte für die Rechnungskonsole
- AWS CloudFormation Schnipsel aus Vorlagen
- CloudFront Amazon-Vorlagenschnipsel
- CloudWatch Amazon-Vorlagenschnipsel
- Amazon CloudWatch Logs-Vorlagenausschnitte
- Amazon DynamoDB-Vorlagenausschnitte
- EC2 CloudFormation Amazon-Vorlagenschnipsel
- Beispielvorlagen für Amazon Elastic Container Service
- Amazon Elastic File System-Beispielvorlage
- Elastic Beanstalk-Vorlagenausschnitte
- Elastic Load Balancing-Vorlagenausschnitte
- AWS Identity and Access Management Schnipsel aus Vorlagen
- AWS Lambda Vorlage
- Amazon Redshift-Vorlagenausschnitte
- Amazon RDS-Vorlagenausschnitte
- Route-53-Vorlagenausschnitte

- Amazon S3-Vorlagenausschnitte
- Amazon SNS-Vorlagenausschnitte
- Amazon SQS-Vorlagenausschnitte
- Amazon Timestream Timestream-Vorlagenausschnitte

Allgemeine Vorlagenausschnitte

Die folgenden Beispiele zeigen verschiedene CloudFormation Vorlagenfunktionen, die nicht spezifisch für einen AWS Dienst sind.

Themen

- Base64codierte Eigenschaft UserData
- Base64codierte UserData Eigenschaft mit und AccessKeySecretKey
- ParametersAbschnitt mit einem literalen Zeichenkettenparameter
- ParametersAbschnitt mit Zeichenkettenparameter mit Einschränkung für reguläre Ausdrücke
- ParametersAbschnitt mit Zahl, Parameter mit MinValue und MaxValue Einschränkungen
- ParametersAbschnitt mit Zahl, Parameter mit Einschränkung AllowedValues
- ParametersAbschnitt mit einem literalen Parameter CommaDelimitedList
- ParametersAbschnitt mit einem Parameterwert, der auf einem Pseudo-Parameter basiert
- MappingAbschnitt mit drei Zuordnungen
- Descriptionbasiert auf einer Literalzeichenfolge
- OutputsAbschnitt mit einer Literalzeichenfolge als Ausgabe
- OutputsAbschnitt mit einer Ressourcenreferenz und einer Pseudoreferenzausgabe
- OutputsAbschnitt mit einer Ausgabe, die auf einer Funktion, einer Literalzeichenfolge, einer Referenz und einem Pseudo-Parameter basiert
- Vorlagenformatversion
- AWSTags Eigenschaft

Base64codierte Eigenschaft UserData

Dieses Beispiel zeigt die Zusammenstellung einer `UserData` Eigenschaft mithilfe der `Fn::Join` Funktionen `Fn::Base64` und. Die Referenzen `MyValue` und `MyName` sind Parameter, die im

`Parameters` Abschnitt der Vorlage definiert werden müssen. Die Literalzeichenfolge `Hello World` ist einfach ein weiterer Wert, der in diesem Beispiel als Teil der `UserData`-Eigenschaft übergeben wird.

JSON

```
"UserData" : {
    "Fn::Base64" : {
        "Fn::Join" : [ ",", [
            { "Ref" : "MyValue" },
            { "Ref" : "MyName" },
            "Hello World" ] ]
    }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
     Ref: MyValue
     Ref: MyName
     Hello World
```

Base64codierte UserData Eigenschaft mit und AccessKeySecretKey

Dieses Beispiel zeigt die Zusammenstellung einer `UserData` Eigenschaft mithilfe der `Fn::Join` Funktionen `Fn::Base64` und. Sie enthält die `AccessKey`- und `SecretKey`-Informationen. Die Referenzen `AccessKey` und `SecretKey` sind Parameter, die im Abschnitt „Parameters“ der Vorlage definiert werden müssen.

JSON

```
"UserData" : {
    "Fn::Base64" : {
        "Fn::Join" : [ "", [
            "ACCESS_KEY=", { "Ref" : "AccessKey" },
            "SECRET_KEY=", { "Ref" : "SecretKey" } ]
        ]
    }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
     ACCESS_KEY=${AccessKey}
     SECRET_KEY=${SecretKey}
```

ParametersAbschnitt mit einem literalen Zeichenkettenparameter

Das folgende Beispiel beschreibt eine gültige Deklaration für den Abschnitt „Parameters“, in welcher ein einzelner Parameter vom Typ `String` deklariert wird.

JSON

```
"Parameters" : {
    "UserName" : {
        "Type" : "String",
        "Default" : "nonadmin",
        "Description" : "Assume a vanilla user if no command-line spec provided"
    }
}
```

YAML

```
Parameters:
  UserName:
    Type: String
    Default: nonadmin
    Description: Assume a vanilla user if no command-line spec provided
```

ParametersAbschnitt mit Zeichenkettenparameter mit Einschränkung für reguläre Ausdrücke

Das folgende Beispiel beschreibt eine gültige Deklaration für den Abschnitt „Parameters“, in welcher ein einzelner Parameter vom Typ `String` deklariert wird. Der `AdminUserAccount` Parameter hat den Standardwert`admin`. Der Parameterwert muss eine Mindestlänge von 1 und eine maximale Länge von 16 aufweisen und enthält alphabetische Zeichen und Zahlen, aber muss mit einem Buchstaben beginnen.

JSON

```
"Parameters" : {
    "AdminUserAccount": {
      "Default": "admin",
      "NoEcho": "true",
      "Description" : "The admin account user name",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "16",
      "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*"
    }
}
```

YAML

```
Parameters:
  AdminUserAccount:
    Default: admin
    NoEcho: true
    Description: The admin account user name
    Type: String
    MinLength: 1
    MaxLength: 16
    AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
```

ParametersAbschnitt mit Zahl, Parameter mit MinValue und MaxValue Einschränkungen

Das folgende Beispiel beschreibt eine gültige Deklaration für den Abschnitt „Parameters“, in welcher ein einzelner Parameter vom Typ `Number` deklariert wird. Der `WebServerPort` Parameter hat einen Standardwert von 80 und einen Minimalwert von 1 und einen Höchstwert von 65535.

JSON

```
"Parameters" : {
    "WebServerPort": {
      "Default": "80",
      "Description" : "TCP/IP port for the web server",
      "Type": "Number",
      "MinValue": "1",
```

```
      "MaxValue": "65535"
    }
}
```

YAML

```
Parameters:
  WebServerPort:
    Default: 80
    Description: TCP/IP port for the web server
    Type: Number
    MinValue: 1
    MaxValue: 65535
```

ParametersAbschnitt mit Zahl, Parameter mit Einschränkung AllowedValues

Das folgende Beispiel beschreibt eine gültige Deklaration für den Abschnitt „Parameters“, in welcher ein einzelner Parameter vom Typ `Number` deklariert wird. Der `WebServerPort` Parameter hat den Standardwert 80 und erlaubt nur Werte von 80 und 8888.

JSON

```
"Parameters" : {
    "WebServerPortLimited": {
      "Default": "80",
      "Description" : "TCP/IP port for the web server",
      "Type": "Number",
      "AllowedValues" : ["80", "8888"]
    }
}
```

YAML

```
Parameters:
  WebServerPortLimited:
    Default: 80
    Description: TCP/IP port for the web server
    Type: Number
    AllowedValues:
    - 80
    - 8888
```

ParametersAbschnitt mit einem literalen Parameter CommaDelimitedList

Das folgende Beispiel zeigt eine gültige `Parameters` Abschnittsdeklaration, in der ein einziger `CommaDelimitedList` Typparameter deklariert wird. Die `NoEcho` Eigenschaft ist auf gesetzt`TRUE`, wodurch ihr Wert in der describe-stacks Ausgabe mit Sternchen (*****) maskiert wird, mit Ausnahme von Informationen, die an den unten angegebenen Speicherorten gespeichert sind.

> **Important**
>
> Durch die Verwendung des `NoEcho`-Attributs werden keine Informationen maskiert, die im Folgenden gespeichert sind:
>
> - Der `Metadata` Vorlagenbereich. CloudFormation transformiert, modifiziert oder redigiert keine Informationen, die Sie in den `Metadata` Abschnitt aufnehmen. Weitere Informationen finden Sie unter Metadaten.
> - Der `Outputs`-Vorlagenabschnitt. Weitere Informationen finden Sie unter Ausgaben:
> - Das `Metadata`-Attribut einer Ressourcendefinition. Weitere Informationen finden Sie unter `Metadata`-Attribut.
>
> Es wird dringend empfohlen, diese Mechanismen nicht zu verwenden, um vertrauliche Informationen wie Passwörter oder Geheimnisse einzugeben.

> **Important**
>
> Anstatt vertrauliche Informationen direkt in Ihre CloudFormation Vorlagen einzubetten, empfehlen wir Ihnen, dynamische Parameter in der Stack-Vorlage zu verwenden, um auf vertrauliche Informationen zu verweisen, die außerhalb von gespeichert und verwaltet werden CloudFormation, z. B. im AWS Systems Manager Parameterspeicher oder. AWS Secrets Manager
> Weitere Informationen finden Sie in den bewährten Methoden zu Keine Anmeldeinformationen in Ihre Vorlagen einbetten.

JSON

```
"Parameters" : {
```

```
    "UserRoles" : {
        "Type" : "CommaDelimitedList",
        "Default" : "guest,newhire",
        "NoEcho" : "TRUE"
    }
}
```

YAML

```
Parameters:
  UserRoles:
    Type: CommaDelimitedList
    Default: "guest,newhire"
    NoEcho: true
```

ParametersAbschnitt mit einem Parameterwert, der auf einem Pseudo-Parameter basiert

Das folgende Beispiel zeigt Befehle in den EC2 Benutzerdaten, die die Pseudo-Parameter `AWS::StackName` und verwenden. `AWS::Region` Weitere Informationen zu Pseudoparametern finden Sie unter AWS Werte mithilfe von Pseudo-Parametern abrufen.

JSON

```
          "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
             "#!/bin/bash -xe\n",
             "yum install -y aws-cfn-bootstrap\n",

             "/opt/aws/bin/cfn-init -v ",
             "         --stack ", { "Ref" : "AWS::StackName" },
             "         --resource LaunchConfig ",
             "         --region ", { "Ref" : "AWS::Region" }, "\n",

             "/opt/aws/bin/cfn-signal -e $? ",
             "         --stack ", { "Ref" : "AWS::StackName" },
             "         --resource WebServerGroup ",
             "         --region ", { "Ref" : "AWS::Region" }, "\n"
        ]]}}
      }
```

YAML

```
UserData:
  Fn::Base64: !Sub |
     #!/bin/bash -xe
     yum update -y aws-cfn-bootstrap
     /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource LaunchConfig --
region ${AWS::Region}
     /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource WebServerGroup
 --region ${AWS::Region}
```

MappingAbschnitt mit drei Zuordnungen

Das folgende Beispiel zeigt eine gültige `Mapping` Abschnittsdeklaration, die drei Zuordnungen enthält. Die Zuweisung liefert bei Übereinstimmung mit dem Zuweisungsschlüssel `Stop`, `SlowDown` oder `Go` die RGB-Werte, die dem entsprechenden `RGBColor`-Attribut zugewiesen wurden.

JSON

```
"Mappings" : {
    "LightColor" : {
        "Stop" : {
            "Description" : "red",
            "RGBColor" : "RED 255 GREEN 0 BLUE 0"
        },
        "SlowDown" : {
            "Description" : "yellow",
            "RGBColor" : "RED 255 GREEN 255 BLUE 0"
        },
        "Go" : {
            "Description" : "green",
            "RGBColor" : "RED 0 GREEN 128 BLUE 0"
        }
    }
}
```

YAML

```
Mappings:
  LightColor:
    Stop:
      Description: red
```

```
      RGBColor: "RED 255 GREEN 0 BLUE 0"
    SlowDown:
      Description: yellow
      RGBColor: "RED 255 GREEN 255 BLUE 0"
    Go:
      Description: green
      RGBColor: "RED 0 GREEN 128 BLUE 0"
```

Descriptionbasiert auf einer Literalzeichenfolge

Das folgende Beispiel zeigt eine gültige `Description` Abschnittsdeklaration, bei der der Wert auf einer Literalzeichenfolge basiert. Dieses Codefragment kann für Vorlagen, Parameter, Ressourcen, Eigenschaften oder Ausgaben verwendet werden.

JSON

```
"Description" : "Replace this value"
```

YAML

```
Description: "Replace this value"
```

OutputsAbschnitt mit einer Literalzeichenfolge als Ausgabe

Dieses Beispiel zeigt eine Ausgabenzuweisung basierend auf einer Literalzeichenfolge.

JSON

```
"Outputs" : {
    "MyPhone" : {
        "Value" : "Please call 555-5555",
        "Description" : "A random message for aws cloudformation describe-stacks"
    }
}
```

YAML

```
Outputs:
  MyPhone:
    Value: Please call 555-5555
```

```
    Description: A random message for aws cloudformation describe-stacks
```

OutputsAbschnitt mit einer Ressourcenreferenz und einer Pseudoreferenzausgabe

Dieses Beispiel zeigt einen `Outputs` Abschnitt mit zwei Ausgabezuweisungen. Eine basiert auf einer Ressource und die andere auf einer Pseudoreferenz.

JSON

```
"Outputs" : {
   "SNSTopic" : { "Value" : { "Ref" : "MyNotificationTopic" } },
   "StackName" : { "Value" : { "Ref" : "AWS::StackName" } }
}
```

YAML

```
Outputs:
  SNSTopic:
    Value: !Ref MyNotificationTopic
  StackName:
    Value: !Ref AWS::StackName
```

OutputsAbschnitt mit einer Ausgabe, die auf einer Funktion, einer Literalzeichenfolge, einer Referenz und einem Pseudo-Parameter basiert

Dieses Beispiel zeigt einen Outputs-Abschnitt mit einer Ausgabenzuweisung. Die Join-Funktion wird verwendet, um den Wert zu verketten, wobei ein Prozentzeichen als Begrenzungszeichen verwendet wird.

JSON

```
"Outputs" : {
    "MyOutput" : {
        "Value" : { "Fn::Join" :
            [ "%", [ "A-string", {"Ref" : "AWS::StackName" } ] ]
        }
    }
}
```

YAML

```
Outputs:
  MyOutput:
    Value: !Join [ %, [ 'A-string', !Ref 'AWS::StackName' ]]
```

Vorlagenformatversion

Der folgende Ausschnitt zeigt eine gültige Abschnittsdeklaration. `AWSTemplateFormatVersion`

JSON

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
```

AWSTags Eigenschaft

Dieses Beispiel zeigt eine Eigenschaft. AWS `Tags` Sie würden diese Eigenschaft im Abschnitt „Properties" einer Ressource angeben. Wenn die Ressource erstellt wird, wird sie mit den Tags markiert, die Sie deklarieren.

JSON

```
"Tags" : [
      {
        "Key" : "keyname1",
        "Value" : "value1"
      },
      {
        "Key" : "keyname2",
        "Value" : "value2"
      }
    ]
```

YAML

```
Tags:
```

```
  -
    Key: "keyname1"
    Value: "value1"
  -
    Key: "keyname2"
    Value: "value2"
```

CloudFormation Vorlagenausschnitte für automatische Skalierung

Mit Amazon EC2 Auto Scaling können Sie EC2 Amazon-Instances automatisch skalieren, entweder mit Skalierungsrichtlinien oder mit geplanter Skalierung. Auto Scaling-Gruppen sind Sammlungen von EC2 Amazon-Instances, die Auto Scaling und Flottenmanagementfunktionen wie Skalierungsrichtlinien, geplante Aktionen, Integritätsprüfungen, Lifecycle-Hooks und Load Balancing ermöglichen.

Application Auto Scaling ermöglicht die automatische Skalierung verschiedener Ressourcen außerhalb von Amazon EC2, entweder mit Skalierungsrichtlinien oder mit geplanter Skalierung.

Mithilfe von AWS CloudFormation Vorlagen können Sie Auto Scaling Scaling-Gruppen, Skalierungsrichtlinien, geplante Aktionen und andere Auto Scaling-Ressourcen als Teil Ihrer Infrastruktur erstellen und konfigurieren. Vorlagen erleichtern die wiederholbare und konsistente Verwaltung und Automatisierung der Bereitstellung von Auto Scaling-Ressourcen.

Die folgenden Beispielvorlagen beschreiben AWS CloudFormation Ressourcen oder Komponenten für Amazon EC2 Auto Scaling und Application Auto Scaling. Diese Ausschnitte sind für die Integration in eine Vorlage gedacht, nicht für eine unabhängige Ausführung.

Kategorien von Ausschnitten

- Konfigurieren Sie Amazon EC2 Auto Scaling Scaling-Ressourcen mit AWS CloudFormation
- Konfigurieren Sie Auto Scaling-Ressourcen für Anwendungen mit AWS CloudFormation

Konfigurieren Sie Amazon EC2 Auto Scaling Scaling-Ressourcen mit AWS CloudFormation

Die folgenden Beispiele zeigen verschiedene Ausschnitte, die in Vorlagen zur Verwendung mit Amazon EC2 Auto Scaling aufgenommen werden können.

Kategorien von Ausschnitten

- Eine Auto Scaling Scaling-Gruppe für eine einzelne Instanz erstellen
- Erstellen Sie eine Auto Scaling Scaling-Gruppe mit einem angeschlossenen Load Balancer
- Erstellen Sie eine Auto Scaling Scaling-Gruppe mit Benachrichtigungen
- Erstellen Sie eine Auto Scaling Scaling-Gruppe, die a CreationPolicy und a verwendet UpdatePolicy
- Erstellen Sie eine Skalierungsrichtlinie
- Gemischte Instances-Gruppen
- Beispiele für Launch-Konfigurationen

Eine Auto Scaling Scaling-Gruppe für eine einzelne Instanz erstellen

Dieses Beispiel zeigt eine `AWS::AutoScaling::AutoScalingGroup`Ressource mit einer einzigen Instanz, um Ihnen den Einstieg zu erleichtern. Die `VPCZoneIdentifier`-Eigenschaft der Auto-Scaling-Gruppe gibt eine Liste vorhandener Subnetze in drei verschiedenen Availability Zones an. Sie müssen das entsprechende Subnetz IDs aus Ihrem Konto angeben, bevor Sie Ihren Stack erstellen. Die `LaunchTemplate` Eigenschaft verweist auf eine `AWS::EC2::LaunchTemplate`Ressource mit dem logischen Namen`myLaunchTemplate`, der an anderer Stelle in Ihrer Vorlage definiert ist.

> **Note**
>
> Beispiele für Startvorlagen finden Sie Erstellen Sie Startvorlagen mit CloudFormation im Abschnitt EC2 Amazon-Snippets und im Abschnitt Beispiele in der `AWS::EC2::LaunchTemplate` Ressource.

JSON

```
"myASG" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
      "LaunchTemplate" : {
        "LaunchTemplateId" : {
          "Ref" : "myLaunchTemplate"
        },
        "Version" : {
```

```
          "Fn::GetAtt" : [
            "myLaunchTemplate",
            "LatestVersionNumber"
          ]
        }
      },
      "MaxSize" : "1",
      "MinSize" : "1"
   }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MaxSize: '1'
    MinSize: '1'
```

Erstellen Sie eine Auto Scaling Scaling-Gruppe mit einem angeschlossenen Load Balancer

Dieses Beispiel zeigt eine `AWS::AutoScaling::AutoScalingGroup`Ressource für den Lastenausgleich über mehrere Server. Es gibt die logischen Namen von AWS Ressourcen an, die an anderer Stelle in derselben Vorlage deklariert wurden.

1. Die `VPCZoneIdentifier` Eigenschaft gibt die logischen Namen von zwei `AWS::EC2::Subnet`Ressourcen an, auf denen die EC2 Instances der Auto Scaling Scaling-Gruppe erstellt werden: `myPublicSubnet1` und`myPublicSubnet2`.
2. Die `LaunchTemplate` Eigenschaft gibt eine `AWS::EC2::LaunchTemplate`Ressource mit dem logischen Namen an`myLaunchTemplate`.
3. Die `TargetGroupARNs`-Eigenschaft listet die Zielgruppen für einen Application Load Balancer oder einen Network Load Balancer auf, der verwendet wird, um den Datenverkehr an die Auto-Scaling-Gruppe weiterzuleiten. In diesem Beispiel wird eine Zielgruppe angegeben,

eine `AWS::ElasticLoadBalancingV2::TargetGroup`Ressource mit dem logischen Namen`myTargetGroup`.

JSON

```
"myServerGroup" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "VPCZoneIdentifier" : [ { "Ref" : "myPublicSubnet1" }, { "Ref" :
 "myPublicSubnet2" } ],
      "LaunchTemplate" : {
        "LaunchTemplateId" : {
          "Ref" : "myLaunchTemplate"
        },
        "Version" : {
          "Fn::GetAtt" : [
            "myLaunchTemplate",
            "LatestVersionNumber"
          ]
        }
      },
      "MaxSize" : "5",
      "MinSize" : "1",
      "TargetGroupARNs" : [ { "Ref" : "myTargetGroup" } ]
   }
}
```

YAML

```
myServerGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - !Ref myPublicSubnet1
      - !Ref myPublicSubnet2
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MaxSize: '5'
    MinSize: '1'
    TargetGroupARNs:
```

```
      - !Ref myTargetGroup
```

Weitere Informationen finden Sie auch unter

Ein detailliertes Beispiel, in dem eine Auto-Scaling-Gruppe mit einer Skalierungsrichtlinie zur Zielverfolgung-basierend auf der vordefinierten `ALBRequestCountPerTarget`-Metrik für Ihren Application Load Balancer erstellt wird, finden Sie im Abschnitt Beispiele in der `AWS::AutoScaling::ScalingPolicy`-Ressource.

Erstellen Sie eine Auto Scaling Scaling-Gruppe mit Benachrichtigungen

Dieses Beispiel zeigt eine `AWS::AutoScaling::AutoScalingGroup`Ressource, die Amazon SNS SNS-Benachrichtigungen sendet, wenn die angegebenen Ereignisse eintreten. Die `NotificationConfigurations` Eigenschaft gibt das SNS-Thema an, an das eine Benachrichtigung AWS CloudFormation gesendet wird, und die Ereignisse, die dazu führen, dass Benachrichtigungen AWS CloudFormation gesendet werden. Wenn die von angegebenen Ereignisse `NotificationTypes` eintreten, AWS CloudFormation wird eine Benachrichtigung an das von angegebene SNS-Thema gesendet. `TopicARN` Wenn Sie den Stack starten, AWS CloudFormation wird eine `AWS::SNS::Subscription`Ressource (`snsTopicForAutoScalingGroup`) erstellt, die in derselben Vorlage deklariert ist.

Die `VPCZoneIdentifier`-Eigenschaft der Auto-Scaling-Gruppe gibt eine Liste vorhandener Subnetze in drei verschiedenen Availability Zones an. Sie müssen das entsprechende Subnetz IDs aus Ihrem Konto angeben, bevor Sie Ihren Stack erstellen. Die `LaunchTemplate` Eigenschaft verweist auf den logischen Namen einer `AWS::EC2::LaunchTemplate`Ressource, die an anderer Stelle in derselben Vorlage deklariert wurde.

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "DependsOn": [
    "snsTopicForAutoScalingGroup"
  ],
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "logicalName"
      },
```

```
      "Version" : {
        "Fn::GetAtt" : [
          "logicalName",
          "LatestVersionNumber"
        ]
      }
    },
    "MaxSize" : "5",
    "MinSize" : "1",
    "NotificationConfigurations" : [
      {
        "TopicARN" : { "Ref" : "snsTopicForAutoScalingGroup" },
        "NotificationTypes" : [
          "autoscaling:EC2_INSTANCE_LAUNCH",
          "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
          "autoscaling:EC2_INSTANCE_TERMINATE",
          "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
          "autoscaling:TEST_NOTIFICATION"
        ]
      }
    ]
  }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    - snsTopicForAutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchTemplate:
      LaunchTemplateId: !Ref logicalName
      Version: !GetAtt logicalName.LatestVersionNumber
    MaxSize: '5'
    MinSize: '1'
    NotificationConfigurations:
      - TopicARN: !Ref snsTopicForAutoScalingGroup
        NotificationTypes:
```

```
          - autoscaling:EC2_INSTANCE_LAUNCH
          - autoscaling:EC2_INSTANCE_LAUNCH_ERROR
          - autoscaling:EC2_INSTANCE_TERMINATE
          - autoscaling:EC2_INSTANCE_TERMINATE_ERROR
          - autoscaling:TEST_NOTIFICATION
```

Erstellen Sie eine Auto Scaling Scaling-Gruppe, die a **`CreationPolicy`** und a verwendet **`UpdatePolicy`**

Das folgende Beispiel zeigt, wie einer `AWS::AutoScaling::AutoScalingGroup`Ressource `UpdatePolicy` Attribute hinzugefügt `CreationPolicy` werden.

Die Richtlinie zur Probenerstellung verhindert, dass die Auto Scaling Scaling-Gruppe `CREATE_COMPLETE` den Status erreicht, bis sie AWS CloudFormation eine `Count` Reihe von Erfolgssignalen erhält, wenn die Gruppe bereit ist. Um zu signalisieren, dass die Auto-Scaling-Gruppe bereit ist, wird ein `cfn-signal`-Hilfsskript, das den Benutzerdaten der Einführungsvorlage hinzugefügt wird (nicht gezeigt), auf den Instances ausgeführt. Wenn die Instances innerhalb des angegebenen Zeitraums kein Signal senden`Timeout`, CloudFormation wird davon ausgegangen, dass die Instances nicht erstellt wurden, CloudFormation schlägt die Ressourcenerstellung fehl und der Stack wird zurückgesetzt.

Die Beispielaktualisierungsrichtlinie weist CloudFormation an, mithilfe der `AutoScalingRollingUpdate` Eigenschaft ein fortlaufendes Update durchzuführen. Die fortlaufende Aktualisierung führt Änderungen an der Auto-Scaling-Gruppe in kleinen Stapeln (für dieses Beispiel Instance nach Instance) basierend auf `MaxBatchSize` durch und eine Pausenzeit zwischen Stapeln von Updates auf der Grundlage von `PauseTime`. Das `MinInstancesInService` Attribut gibt die Mindestanzahl von Instances an, die innerhalb der Auto Scaling Scaling-Gruppe in Betrieb sein müssen, während alte Instances CloudFormation aktualisiert werden.

Das `WaitOnResourceSignals`-Attribut ist auf `true` festgelegt. CloudFormation muss von jeder neuen Instance innerhalb der angegebenen `PauseTime` ein Signal erhalten, bevor die Aktualisierung fortgesetzt wird. Während das Stack-Update läuft, werden die folgenden EC2 Auto Scaling Scaling-Prozesse ausgesetzt: `HealthCheck` `ReplaceUnhealthyAZRebalance`,,`AlarmNotification`, und`ScheduledActions`. Hinweis: Unterbrechen Sie die Prozesstypen `Launch`, `Terminate` oder `AddToLoadBalancer` (wenn die Auto-Scaling-Gruppe mit Elastic Load Balancing verwendet wird) nicht, da dies die ordnungsgemäße Funktion der fortlaufenden Aktualisierung verhindern kann.

Die `VPCZoneIdentifier`-Eigenschaft der Auto-Scaling-Gruppe gibt eine Liste vorhandener Subnetze in drei verschiedenen Availability Zones an. Sie müssen das entsprechende Subnetz

IDs aus Ihrem Konto angeben, bevor Sie Ihren Stack erstellen. Die `LaunchTemplate` Eigenschaft verweist auf den logischen Namen einer `AWS::EC2::LaunchTemplate`Ressource, die an anderer Stelle in derselben Vorlage deklariert wurde.

Weitere Informationen zu den `UpdatePolicy` Attributen `CreationPolicy` und finden Sie unter Referenz zu Ressourcenattributen.

JSON

```
{
  "Resources":{
    "myASG":{
      "CreationPolicy":{
        "ResourceSignal":{
          "Count":"3",
          "Timeout":"PT15M"
        }
      },
      "UpdatePolicy":{
        "AutoScalingRollingUpdate":{
          "MinInstancesInService":"3",
          "MaxBatchSize":"1",
          "PauseTime":"PT12M5S",
          "WaitOnResourceSignals":"true",
          "SuspendProcesses":[
            "HealthCheck",
            "ReplaceUnhealthy",
            "AZRebalance",
            "AlarmNotification",
            "ScheduledActions",
            "InstanceRefresh"
          ]
        }
      },
      "Type":"AWS::AutoScaling::AutoScalingGroup",
      "Properties":{
        "VPCZoneIdentifier":[ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
        "LaunchTemplate":{
          "LaunchTemplateId":{
            "Ref":"logicalName"
          },
          "Version":{
            "Fn::GetAtt":[
```

```
              "logicalName",
              "LatestVersionNumber"
            ]
          }
        },
        "MaxSize":"5",
        "MinSize":"3"
      }
    }
  }
}
```

YAML

```
---
Resources:
  myASG:
    CreationPolicy:
      ResourceSignal:
        Count: '3'
        Timeout: PT15M
    UpdatePolicy:
      AutoScalingRollingUpdate:
        MinInstancesInService: '3'
        MaxBatchSize: '1'
        PauseTime: PT12M5S
        WaitOnResourceSignals: true
        SuspendProcesses:
          - HealthCheck
          - ReplaceUnhealthy
          - AZRebalance
          - AlarmNotification
          - ScheduledActions
          - InstanceRefresh
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      VPCZoneIdentifier:
        - subnetIdAz1
        - subnetIdAz2
        - subnetIdAz3
      LaunchTemplate:
        LaunchTemplateId: !Ref logicalName
        Version: !GetAtt logicalName.LatestVersionNumber
```

```
      MaxSize: '5'
      MinSize: '3'
```

Erstellen Sie eine Skalierungsrichtlinie

Dieses Beispiel zeigt eine `AWS::AutoScaling::ScalingPolicy`Ressource, die die Auto Scaling Scaling-Gruppe mithilfe einer Step Scaling-Richtlinie skaliert. Die Eigenschaft `AdjustmentType` ist auf `ChangeInCapacity` festgelegt. Das bedeutet, dass der Parameter `ScalingAdjustment` angibt, wie viele Instances hinzugefügt werden sollen (bei positivem Wert für `ScalingAdjustment`) oder gelöscht werden sollen (bei negativem Wert). In diesem Beispiel `ScalingAdjustment` ist es 1; daher erhöht die Richtlinie die Anzahl der EC2 Instanzen in der Gruppe um 1, wenn der Alarmschwellenwert überschritten wird.

Die `AWS::CloudWatch::Alarm`Ressource `CPUAlarmHigh` gibt die Skalierungsrichtlinie `ASGScalingPolicyHigh` als die Aktion an, die ausgeführt werden soll, wenn sich der Alarm im ALARM-Status () `AlarmActions` befindet. Die `Dimensions` Eigenschaft verweist auf den logischen Namen einer `AWS::AutoScaling::AutoScalingGroup`Ressource, die an anderer Stelle in derselben Vorlage deklariert wurde.

JSON

```
{
  "Resources":{
    "ASGScalingPolicyHigh":{
      "Type":"AWS::AutoScaling::ScalingPolicy",
      "Properties":{
        "AutoScalingGroupName":{ "Ref":"logicalName" },
        "PolicyType":"StepScaling",
        "AdjustmentType":"ChangeInCapacity",
        "StepAdjustments":[
          {
            "MetricIntervalLowerBound":0,
            "ScalingAdjustment":1
          }
        ]
      }
    },
    "CPUAlarmHigh":{
      "Type":"AWS::CloudWatch::Alarm",
      "Properties":{
        "EvaluationPeriods":"2",
```

```
        "Statistic":"Average",
        "Threshold":"90",
        "AlarmDescription":"Scale out if CPU > 90% for 2 minutes",
        "Period":"60",
        "AlarmActions":[ { "Ref":"ASGScalingPolicyHigh" } ],
        "Namespace":"AWS/EC2",
        "Dimensions":[
          {
            "Name":"AutoScalingGroupName",
            "Value":{ "Ref":"logicalName" }
          }
        ],
        "ComparisonOperator":"GreaterThanThreshold",
        "MetricName":"CPUUtilization"
      }
    }
  }
}
```

YAML

```
---
Resources:
  ASGScalingPolicyHigh:
    Type: AWS::AutoScaling::ScalingPolicy
    Properties:
      AutoScalingGroupName: !Ref logicalName
      PolicyType: StepScaling
      AdjustmentType: ChangeInCapacity
      StepAdjustments:
        - MetricIntervalLowerBound: 0
          ScalingAdjustment: 1
  CPUAlarmHigh:
    Type: AWS::CloudWatch::Alarm
    Properties:
      EvaluationPeriods: 2
      Statistic: Average
      Threshold: 90
      AlarmDescription: 'Scale out if CPU > 90% for 2 minutes'
      Period: 60
      AlarmActions:
        - !Ref ASGScalingPolicyHigh
      Namespace: AWS/EC2
```

```
      Dimensions:
        - Name: AutoScalingGroupName
          Value:
            !Ref logicalName
      ComparisonOperator: GreaterThanThreshold
      MetricName: CPUUtilization
```

Weitere Informationen finden Sie auch unter

Weitere Beispielvorlagen für Skalierungsrichtlinien finden Sie im Abschnitt Beispiele in der `AWS::AutoScaling::ScalingPolicy`-Ressource.

Gemischte Instances-Gruppen

Erstellen einer Auto-Scaling-Gruppe mit attributbasierter Auswahl des Instance-Typs

Dieses Beispiel zeigt eine `AWS::AutoScaling::AutoScalingGroup`Ressource, die die Informationen zum Starten einer gemischten Instanzgruppe mithilfe der attributbasierten Instanztypauswahl enthält. Sie geben die Mindest- und Höchstwerte für die `VCpuCount`-Eigenschaft und der Mindestwert für die `MemoryMiB`-Eigenschaft ein. Alle Instance-Typen, die von der Auto-Scaling-Gruppe verwendet werden, müssen mit Ihren erforderlichen Instance-Attributen übereinstimmen.

Die `VPCZoneIdentifier`-Eigenschaft der Auto-Scaling-Gruppe gibt eine Liste vorhandener Subnetze in drei verschiedenen Availability Zones an. Sie müssen das entsprechende Subnetz IDs aus Ihrem Konto angeben, bevor Sie Ihren Stack erstellen. Die `LaunchTemplate` Eigenschaft verweist auf den logischen Namen einer `AWS::EC2::LaunchTemplate`Ressource, die an anderer Stelle in derselben Vorlage deklariert wurde.

JSON

```
{
  "Resources":{
    "myASG":{
      "Type":"AWS::AutoScaling::AutoScalingGroup",
      "Properties":{
        "VPCZoneIdentifier":[
          "subnetIdAz1",
          "subnetIdAz2",
          "subnetIdAz3"
        ],
        "MixedInstancesPolicy":{
```

```
          "LaunchTemplate":{
            "LaunchTemplateSpecification":{
              "LaunchTemplateId":{
                "Ref":"logicalName"
              },
              "Version":{
                "Fn::GetAtt":[
                  "logicalName",
                  "LatestVersionNumber"
                ]
              }
            },
            "Overrides":[
              {
                "InstanceRequirements":{
                  "VCpuCount":{
                    "Min":2,
                    "Max":4
                  },
                  "MemoryMiB":{
                    "Min":2048
                  }
                }
              }
            ]
          }
        },
        "MaxSize":"5",
        "MinSize":"1"
      }
    }
  }
}
```

YAML

```
---
Resources:
  myASG:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      VPCZoneIdentifier:
        - subnetIdAz1
```

```
        - subnetIdAz2
        - subnetIdAz3
      MixedInstancesPolicy:
        LaunchTemplate:
          LaunchTemplateSpecification:
            LaunchTemplateId: !Ref logicalName
            Version: !GetAtt logicalName.LatestVersionNumber
          Overrides:
            - InstanceRequirements:
                VCpuCount:
                  Min: 2
                  Max: 4
                MemoryMiB:
                  Min: 2048
      MaxSize: '5'
      MinSize: '1'
```

Beispiele für Launch-Konfigurationen

Erstellen einer Startkonfiguration

Dieses Beispiel zeigt eine `AWS::AutoScaling::LaunchConfiguration`Ressource für eine Auto Scaling Scaling-Gruppe, in der Sie Werte für die `SecurityGroups` Eigenschaften `ImageIdInstanceType`, und angeben. Die `SecurityGroups` Eigenschaft gibt sowohl den logischen Namen einer `AWS::EC2::SecurityGroup`Ressource an, die an anderer Stelle in der Vorlage angegeben ist, als auch den Namen einer vorhandenen EC2 Sicherheitsgruppe`myExistingEC2SecurityGroup`.

JSON

```
"mySimpleConfig" : {
   "Type" : "AWS::AutoScaling::LaunchConfiguration",
   "Properties" : {
      "ImageId" : "ami-02354e95b3example",
      "InstanceType" : "t3.micro",
      "SecurityGroups" : [ { "Ref" : "logicalName" }, "myExistingEC2SecurityGroup" ]
   }
}
```

YAML

```
mySimpleConfig:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId: ami-02354e95b3example
    InstanceType: t3.micro
    SecurityGroups:
      - !Ref logicalName
      - myExistingEC2SecurityGroup
```

Erstellen Sie eine Auto Scaling Scaling-Gruppe, die eine Startkonfiguration verwendet

Dieses Beispiel zeigt eine `AWS::AutoScaling::AutoScalingGroup`Ressource mit einer einzigen Instanz. Die `VPCZoneIdentifier`-Eigenschaft der Auto-Scaling-Gruppe gibt eine Liste vorhandener Subnetze in drei verschiedenen Availability Zones an. Sie müssen das entsprechende Subnetz IDs aus Ihrem Konto angeben, bevor Sie Ihren Stack erstellen. Die `LaunchConfigurationName` Eigenschaft verweist auf eine `AWS::AutoScaling::LaunchConfiguration`Ressource mit dem logischen Namen`mySimpleConfig`, der in Ihrer Vorlage definiert ist.

JSON

```
"myASG" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
      "LaunchConfigurationName" : { "Ref" : "mySimpleConfig" },
      "MaxSize" : "1",
      "MinSize" : "1"
   }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
```

```
    LaunchConfigurationName: !Ref mySimpleConfig
    MaxSize: '1'
    MinSize: '1'
```

Konfigurieren Sie Auto Scaling-Ressourcen für Anwendungen mit AWS CloudFormation

Dieser Abschnitt enthält AWS CloudFormation Vorlagenbeispiele für Richtlinien zur Skalierung von Application Auto Scaling und geplante Aktionen für verschiedene AWS Ressourcen.

> **Important**
>
> Wenn ein Application Auto Scaling-Snippet in der Vorlage enthalten ist, müssen Sie möglicherweise mithilfe des Attributs eine Abhängigkeit von der spezifischen skalierbaren Ressource deklarieren, die über die Vorlage erstellt wurde. `DependsOn` Dies überschreibt die Standardparallelität und weist AWS CloudFormation darauf hin, Ressourcen in einer bestimmten Reihenfolge zu verwenden. Andernfalls kann die Skalierungskonfiguration angewendet werden, bevor die Ressource vollständig eingerichtet wurde.
> Weitere Informationen finden Sie unter DependsOn-Attribut.

Kategorien von Ausschnitten

- Erstellen Sie eine Skalierungsrichtlinie für eine Flotte AppStream
- Erstellen Sie eine Skalierungsrichtlinie für einen Aurora-DB-Cluster
- Erstellen Sie eine Skalierungsrichtlinie für eine DynamoDB-Tabelle
- Erstellen Sie eine Skalierungsrichtlinie für einen Amazon ECS-Service (Metriken: durchschnittliche CPU und Arbeitsspeicher)
- Erstellen Sie eine Skalierungsrichtlinie für einen Amazon ECS-Service (Metrik: durchschnittliche Anzahl von Anfragen pro Ziel)
- Erstellen Sie eine geplante Aktion mit einem Cron-Ausdruck für eine Lambda-Funktion
- Erstellen Sie eine geplante Aktion mit einem at Ausdruck für eine Spot-Flotte

Erstellen Sie eine Skalierungsrichtlinie für eine Flotte AppStream

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie erstellen und sie auf eine Ressource anwenden, die die `AWS::AppStream::FleetAWS::ApplicationAutoScaling::ScalingPolicy`Ressource

verwendet. Die `AWS::ApplicationAutoScaling::ScalableTarget`Ressource deklariert ein skalierbares Ziel, auf das diese Richtlinie angewendet wird. Das Auto Scaling von Anwendungen kann die Anzahl der Flotten-Instances auf mindestens 1 Instance und maximal 20 skalieren. Die Richtlinie hält die durchschnittliche Kapazitätsauslastung der Flotte bei 75 Prozent, mit Aufskalierungs- und Abskalierungs-Ruhephasen von 300 Sekunden (5 Minuten).

Es verwendet die `Fn::Join` `Rev` systemeigenen Funktionen und, um die `ResourceId` Eigenschaft mit dem logischen Namen der `AWS::AppStream::Fleet` Ressource zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter Referenz zu systeminternen Funktionen.

JSON

```
{
  "Resources" : {
    "ScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 20,
        "MinCapacity" : 1,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet" },
        "ServiceNamespace" : "appstream",
        "ScalableDimension" : "appstream:fleet:DesiredCapacity",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "fleet",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        }
      }
    },
    "ScalingPolicyAppStreamFleet" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu75" },
```

```
        "PolicyType" : "TargetTrackingScaling",
        "ServiceNamespace" : "appstream",
        "ScalableDimension" : "appstream:fleet:DesiredCapacity",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "fleet",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        },
        "TargetTrackingScalingPolicyConfiguration" : {
          "TargetValue" : 75,
          "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "AppStreamAverageCapacityUtilization"
          },
          "ScaleInCooldown" : 300,
          "ScaleOutCooldown" : 300
        }
      }
    }
  }
}
```

YAML

```
---
Resources:
  ScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 20
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet'
      ServiceNamespace: appstream
      ScalableDimension: appstream:fleet:DesiredCapacity
      ResourceId: !Join
```

```
        - /
        - - fleet
          - !Ref logicalName
  ScalingPolicyAppStreamFleet:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu75
      PolicyType: TargetTrackingScaling
      ServiceNamespace: appstream
      ScalableDimension: appstream:fleet:DesiredCapacity
      ResourceId: !Join
        - /
        - - fleet
          - !Ref logicalName
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 75
        PredefinedMetricSpecification:
          PredefinedMetricType: AppStreamAverageCapacityUtilization
        ScaleInCooldown: 300
        ScaleOutCooldown: 300
```

Erstellen Sie eine Skalierungsrichtlinie für einen Aurora-DB-Cluster

In diesem Snippet registrieren Sie eine `AWS::RDS::DBCluster`Ressource. Die `AWS::ApplicationAutoScaling::ScalableTarget`Ressource gibt an, dass der DB-Cluster dynamisch auf ein bis acht Aurora Replicas skaliert werden sollte. Sie wenden außerdem eine Skalierungsrichtlinie für die Zielverfolgung auf den Cluster an, der die `AWS::ApplicationAutoScaling::ScalingPolicy`Ressource verwendet.

In dieser Konfiguration wird die vordefinierte Metrik `RDSReaderAverageCPUUtilization` verwendet, um einen Aurora DB-Cluster anzupassen, basierend auf einer durchschnittlichen CPU-Auslastung von 40 Prozent über alle Aurora -Repliken hinweg in diesem Aurora DB-Cluster. Die Konfiguration sieht eine Scale-In-Ruhephase von 10 Minuten und eine Scale-Out-Ruhephase von 5 Minuten vor.

In diesem Beispiel wird die `Fn::Sub` systeminterne Funktion verwendet, um die `ResourceId` Eigenschaft mit dem logischen Namen der `AWS::RDS::DBCluster` Ressource zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter Referenz zu systeminternen Funktionen.

JSON

```
{
  "Resources" : {
    "ScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 8,
        "MinCapacity" : 1,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/rds.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_RDSCluster" },
        "ServiceNamespace" : "rds",
        "ScalableDimension" : "rds:cluster:ReadReplicaCount",
        "ResourceId" : { "Fn::Sub" : "cluster:${logicalName}" }
      }
    },
    "ScalingPolicyDBCluster" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu40" },
        "PolicyType" : "TargetTrackingScaling",
        "ServiceNamespace" : "rds",
        "ScalableDimension" : "rds:cluster:ReadReplicaCount",
        "ResourceId" : { "Fn::Sub" : "cluster:${logicalName}" },
        "TargetTrackingScalingPolicyConfiguration" : {
          "TargetValue" : 40,
          "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "RDSReaderAverageCPUUtilization"
          },
          "ScaleInCooldown" : 600,
          "ScaleOutCooldown" : 300
        }
      }
    }
  }
}
```

YAML

```
---
Resources:
  ScalableTarget:
```

```
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 8
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/rds.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_RDSCluster'
      ServiceNamespace: rds
      ScalableDimension: rds:cluster:ReadReplicaCount
      ResourceId: !Sub cluster:${logicalName}
  ScalingPolicyDBCluster:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu40
      PolicyType: TargetTrackingScaling
      ServiceNamespace: rds
      ScalableDimension: rds:cluster:ReadReplicaCount
      ResourceId: !Sub cluster:${logicalName}
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 40
        PredefinedMetricSpecification:
          PredefinedMetricType: RDSReaderAverageCPUUtilization
        ScaleInCooldown: 600
        ScaleOutCooldown: 300
```

Erstellen Sie eine Skalierungsrichtlinie für eine DynamoDB-Tabelle

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie mit dem `TargetTrackingScaling` Richtlinientyp erstellen und sie auf eine `AWS::DynamoDB::Table`Ressource anwenden, die die Ressource verwendet. `AWS::ApplicationAutoScaling::ScalingPolicy` Die `AWS::ApplicationAutoScaling::ScalableTarget`Ressource deklariert ein skalierbares Ziel, auf das diese Richtlinie angewendet wird, mit mindestens fünf Schreibkapazitätseinheiten und maximal 15. Die Skalierungsrichtlinie skaliert den Schreibkapazitätsdurchsatz der Tabelle, um die Zielauslastung basierend auf der vordefinierten `DynamoDBWriteCapacityUtilization`-Metrik auf 50 Prozent zu halten.

Es verwendet die `Fn::Join` `Ref` systemeigenen Funktionen und, um die `ResourceId` Eigenschaft mit dem logischen Namen der `AWS::DynamoDB::Table` Ressource zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter Referenz zu systeminternen Funktionen.

> **Note**
>
> Weitere Informationen zum Erstellen einer AWS CloudFormation Vorlage für DynamoDB-Ressourcen finden Sie im Blogbeitrag How to use AWS CloudFormation to configure auto scaling for Amazon DynamoDB tables and indexes im Datenbank-Blog. AWS

JSON

```
{
  "Resources" : {
    "WriteCapacityScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 15,
        "MinCapacity" : 5,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable" },
        "ServiceNamespace" : "dynamodb",
        "ScalableDimension" : "dynamodb:table:WriteCapacityUnits",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "table",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        }
      }
    },
    "WriteScalingPolicy" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : "WriteScalingPolicy",
        "PolicyType" : "TargetTrackingScaling",
        "ScalingTargetId" : { "Ref" : "WriteCapacityScalableTarget" },
        "TargetTrackingScalingPolicyConfiguration" : {
          "TargetValue" : 50.0,
```

```
          "ScaleInCooldown" : 60,
          "ScaleOutCooldown" : 60,
          "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "DynamoDBWriteCapacityUtilization"
          }
        }
      }
    }
  }
}
```

YAML

```
---
Resources:
  WriteCapacityScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 15
      MinCapacity: 5
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable'
      ServiceNamespace: dynamodb
      ScalableDimension: dynamodb:table:WriteCapacityUnits
      ResourceId: !Join
        - /
        - - table
          - !Ref logicalName
  WriteScalingPolicy:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: WriteScalingPolicy
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref WriteCapacityScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 50.0
        ScaleInCooldown: 60
        ScaleOutCooldown: 60
        PredefinedMetricSpecification:
          PredefinedMetricType: DynamoDBWriteCapacityUtilization
```

Erstellen Sie eine Skalierungsrichtlinie für einen Amazon ECS-Service (Metriken: durchschnittliche CPU und Arbeitsspeicher)

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie erstellen und sie auf eine Ressource anwenden, die die `AWS::ECS::Service``AWS::ApplicationAutoScaling::ScalingPolicy`Ressource verwendet. Die `AWS::ApplicationAutoScaling::ScalableTarget`Ressource deklariert ein skalierbares Ziel, auf das diese Richtlinie angewendet wird. Application Auto Scaling kann die Anzahl der Aufgaben mit mindestens 1 und maximal 6 Aufgaben skalieren.

Es werden zwei Skalierungsrichtlinien mit dem `TargetTrackingScaling`-Richtlinientyp erstellt. Die Richtlinien werden verwendet, um den ECS-Service basierend auf der durchschnittlichen CPU- und Speicherauslastung des Service zu skalieren. Sie verwendet die `Ref` systemeigenen Funktionen `Fn::Join` und, um die `ResourceId` Eigenschaft mit den logischen Namen der Ressourcen `AWS::ECS::Cluster`(`myContainerCluster`) und `AWS::ECS::Service` (`myService`) zu erstellen, die in derselben Vorlage angegeben sind. Weitere Informationen finden Sie unter Referenz zu systeminternen Funktionen.

JSON

```
{
  "Resources" : {
    "ECSScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : "6",
        "MinCapacity" : "1",
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService" },
        "ServiceNamespace" : "ecs",
        "ScalableDimension" : "ecs:service:DesiredCount",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "service",
              {
                "Ref" : "myContainerCluster"
              },
              {
                "Fn::GetAtt" : [
                  "myService",
```

```
                    "Name"
                  ]
                }
              ]
            ]
          }
        }
      },
      "ServiceScalingPolicyCPU" : {
        "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
        "Properties" : {
          "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu70" },
          "PolicyType" : "TargetTrackingScaling",
          "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
          "TargetTrackingScalingPolicyConfiguration" : {
            "TargetValue" : 70.0,
            "ScaleInCooldown" : 180,
            "ScaleOutCooldown" : 60,
            "PredefinedMetricSpecification" : {
              "PredefinedMetricType" : "ECSServiceAverageCPUUtilization"
            }
          }
        }
      },
      "ServiceScalingPolicyMem" : {
        "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
        "Properties" : {
          "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-mem90" },
          "PolicyType" : "TargetTrackingScaling",
          "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
          "TargetTrackingScalingPolicyConfiguration" : {
            "TargetValue" : 90.0,
            "ScaleInCooldown" : 180,
            "ScaleOutCooldown" : 60,
            "PredefinedMetricSpecification" : {
              "PredefinedMetricType" : "ECSServiceAverageMemoryUtilization"
            }
          }
        }
      }
    }
}
```

YAML

```
---
Resources:
  ECSScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 6
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ecs.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService'
      ServiceNamespace: ecs
      ScalableDimension: 'ecs:service:DesiredCount'
      ResourceId: !Join
        - /
        - - service
          - !Ref myContainerCluster
          - !GetAtt myService.Name
  ServiceScalingPolicyCPU:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu70
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref ECSScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 70.0
        ScaleInCooldown: 180
        ScaleOutCooldown: 60
        PredefinedMetricSpecification:
          PredefinedMetricType: ECSServiceAverageCPUUtilization
  ServiceScalingPolicyMem:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: !Sub ${AWS::StackName}-target-tracking-mem90
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref ECSScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 90.0
        ScaleInCooldown: 180
        ScaleOutCooldown: 60
        PredefinedMetricSpecification:
          PredefinedMetricType: ECSServiceAverageMemoryUtilization
```

Erstellen Sie eine Skalierungsrichtlinie für einen Amazon ECS-Service (Metrik: durchschnittliche Anzahl von Anfragen pro Ziel)

Im folgenden Beispiel wird eine Skalierungsrichtlinie zur Zielverfolgung mit der vordefinierten `ALBRequestCountPerTarget` Metrik auf einen ECS-Service angewendet. Die Richtlinie wird verwendet, um dem ECS-Service Kapazität hinzuzufügen, wenn die Anforderungsanzahl pro Ziel (pro Minute) den Zielwert überschreitet. Da der Wert von `DisableScaleIn` auf „`true`“ festgelegt ist, entfernt die Richtlinie für die Ziel-Nachverfolgung keine Kapazität vom skalierbaren Ziel.

Es verwendet die `Fn::GetAtt` systemeigenen Funktionen `Fn::Join` und, um die `ResourceLabel` Eigenschaft mit den logischen Namen der Ressourcen `AWS::ElasticLoadBalancingV2::LoadBalancer`(`myLoadBalancer`) und `AWS::ElasticLoadBalancingV2::TargetGroup`(`myTargetGroup`) zu erstellen, die in derselben Vorlage angegeben sind. Weitere Informationen finden Sie unter Referenz zu systeminternen Funktionen.

Die Eigenschaften `MaxCapacity` und `MinCapacity` des skalierbaren Ziels und die `TargetValue`-Eigenschaft der Referenzparameterwerte der Skalierungsrichtlinie, die Sie beim Erstellen oder Aktualisieren eines Stacks an die Vorlage übergeben.

JSON

```
{
  "Resources" : {
    "ECSScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : { "Ref" : "MaxCount" },
        "MinCapacity" : { "Ref" : "MinCount" },
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService" },
        "ServiceNamespace" : "ecs",
        "ScalableDimension" : "ecs:service:DesiredCount",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "service",
              {
                "Ref" : "myContainerCluster"
              },
```

```
              {
                "Fn::GetAtt" : [
                  "myService",
                  "Name"
                ]
              }
            ]
          ]
        }
      }
    },
    "ServiceScalingPolicyALB" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : "alb-requests-per-target-per-minute",
        "PolicyType" : "TargetTrackingScaling",
        "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
        "TargetTrackingScalingPolicyConfiguration" : {
          "TargetValue" : { "Ref" : "ALBPolicyTargetValue" },
          "ScaleInCooldown" : 180,
          "ScaleOutCooldown" : 30,
          "DisableScaleIn" : true,
          "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "ALBRequestCountPerTarget",
            "ResourceLabel" : {
              "Fn::Join" : [
                "/",
                [
                  {
                    "Fn::GetAtt" : [
                      "myLoadBalancer",
                      "LoadBalancerFullName"
                    ]
                  },
                  {
                    "Fn::GetAtt" : [
                      "myTargetGroup",
                      "TargetGroupFullName"
                    ]
                  }
                ]
              ]
            }
          }
```

```
        }
      }
    }
  }
}
```

YAML

```
---
Resources:
  ECSScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: !Ref MaxCount
      MinCapacity: !Ref MinCount
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ecs.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService'
      ServiceNamespace: ecs
      ScalableDimension: 'ecs:service:DesiredCount'
      ResourceId: !Join
        - /
        - - service
          - !Ref myContainerCluster
          - !GetAtt myService.Name
  ServiceScalingPolicyALB:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: alb-requests-per-target-per-minute
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref ECSScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: !Ref ALBPolicyTargetValue
        ScaleInCooldown: 180
        ScaleOutCooldown: 30
        DisableScaleIn: true
        PredefinedMetricSpecification:
          PredefinedMetricType: ALBRequestCountPerTarget
          ResourceLabel: !Join
            - '/'
            - - !GetAtt myLoadBalancer.LoadBalancerFullName
              - !GetAtt myTargetGroup.TargetGroupFullName
```

Erstellen Sie eine geplante Aktion mit einem Cron-Ausdruck für eine Lambda-Funktion

Dieses Snippet registriert die bereitgestellte Parallelität für eine Funktion alias () `AWS::Lambda::Alias`, die mithilfe der Ressource benannt wurde. `BLUE` `AWS::ApplicationAutoScaling::ScalableTarget` Erstellt auch eine geplante Aktion mit einem wiederkehrenden Zeitplan unter Verwendung eines Cron-Ausdrucks. Die Zeitzone für den wiederkehrenden Zeitplan ist UTC.

Es verwendet die `Fn::Join` `Ref` systemeigenen Funktionen und in der `RoleARN` Eigenschaft, um den ARN der dienstverknüpften Rolle anzugeben. Sie verwendet die `Fn::Sub` systeminterne Funktion, um die `ResourceId` Eigenschaft mit dem logischen Namen der `AWS::Serverless::Function`Ressource `AWS::Lambda::Function`OR zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter Referenz zu systeminternen Funktionen.

> **Note**
>
> Sie können einem Alias, der auf die unveröffentlichte Version () verweist, keine bereitgestellte Parallelität zuweisen. `$LATEST`
> Weitere Informationen zum Erstellen einer AWS CloudFormation Vorlage für Lambda-Ressourcen finden Sie im Blogbeitrag Scheduling AWS Lambda Provisioned Concurrency for Recurring Peak Usage auf dem Compute-Blog. AWS

JSON

```
{
  "ScalableTarget" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties" : {
      "MaxCapacity" : 250,
      "MinCapacity" : 0,
      "RoleARN" : {
        "Fn::Join" : [
          ":",
          [
            "arn:aws:iam:",
            {
              "Ref" : "AWS::AccountId"
            },
```

```
            "role/aws-service-role/lambda.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency"
          ]
        ]
      },
      "ServiceNamespace" : "lambda",
      "ScalableDimension" : "lambda:function:ProvisionedConcurrency",
      "ResourceId" : { "Fn::Sub" : "function:${logicalName}:BLUE" },
      "ScheduledActions" : [
        {
          "ScalableTargetAction" : {
            "MinCapacity" : "250"
          },
          "ScheduledActionName" : "my-scale-out-scheduled-action",
          "Schedule" : "cron(0 18 * * ? *)",
          "EndTime" : "2022-12-31T12:00:00.000Z"
        }
      ]
    }
  }
}
```

YAML

```
ScalableTarget:
  Type: AWS::ApplicationAutoScaling::ScalableTarget
  Properties:
    MaxCapacity: 250
    MinCapacity: 0
    RoleARN: !Join
      - ':'
      - - 'arn:aws:iam:'
        - !Ref 'AWS::AccountId'
        - role/aws-service-role/lambda.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency
    ServiceNamespace: lambda
    ScalableDimension: lambda:function:ProvisionedConcurrency
    ResourceId: !Sub function:${logicalName}:BLUE
    ScheduledActions:
      - ScalableTargetAction:
          MinCapacity: 250
        ScheduledActionName: my-scale-out-scheduled-action
        Schedule: 'cron(0 18 * * ? *)'
```

```
        EndTime: '2022-12-31T12:00:00.000Z'
```

Erstellen Sie eine geplante Aktion mit einem **at** Ausdruck für eine Spot-Flotte

Dieser Ausschnitt zeigt, wie zwei geplante Aktionen erstellt werden, die für eine `AWS::EC2::SpotFleet`Ressource, die die Ressource verwendet, nur einmal ausgeführt werden. `AWS::ApplicationAutoScaling::ScalableTarget` Die Zeitzone für jede einmalig geplante Aktion ist UTC.

Es verwendet die `Fn::Join` `Ref` systemeigenen Funktionen und, um die `ResourceId` Eigenschaft mit dem logischen Namen der `AWS::EC2::SpotFleet` Ressource zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter Referenz zu systeminternen Funktionen.

> **Note**
>
> Die Spot-Flottenanforderung muss den Anforderungstyp `maintain` aufweisen. Auto Scaling wird für einmalige Anforderungen oder Spot-Blöcke nicht unterstützt.

JSON

```
{
  "Resources" : {
    "SpotFleetScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 0,
        "MinCapacity" : 0,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ec2.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest" },
        "ServiceNamespace" : "ec2",
        "ScalableDimension" : "ec2:spot-fleet-request:TargetCapacity",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "spot-fleet-request",
              {
                "Ref" : "logicalName"
```

```
              }
            ]
          ]
        },
        "ScheduledActions" : [
          {
            "ScalableTargetAction" : {
              "MaxCapacity" : "10",
              "MinCapacity" : "10"
            },
            "ScheduledActionName" : "my-scale-out-scheduled-action",
            "Schedule" : "at(2022-05-20T13:00:00)"
          },
          {
            "ScalableTargetAction" : {
              "MaxCapacity" : "0",
              "MinCapacity" : "0"
            },
            "ScheduledActionName" : "my-scale-in-scheduled-action",
            "Schedule" : "at(2022-05-20T21:00:00)"
          }
        ]
      }
    }
  }
}
```

YAML

```
---
Resources:
  SpotFleetScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 0
      MinCapacity: 0
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ec2.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest'
      ServiceNamespace: ec2
      ScalableDimension: 'ec2:spot-fleet-request:TargetCapacity'
      ResourceId: !Join
        - /
```

```
        - - spot-fleet-request
          - !Ref logicalName
      ScheduledActions:
        - ScalableTargetAction:
            MaxCapacity: 10
            MinCapacity: 10
          ScheduledActionName: my-scale-out-scheduled-action
          Schedule: 'at(2022-05-20T13:00:00)'
        - ScalableTargetAction:
            MaxCapacity: 0
            MinCapacity: 0
          ScheduledActionName: my-scale-in-scheduled-action
          Schedule: 'at(2022-05-20T21:00:00)'
```

AWS Vorlagenausschnitte für die Rechnungskonsole

In diesem Beispiel wird ein Preisplan mit einer globalen Aufschlagspreisregel von 10 % erstellt. Dieser Preisplan ist der Abrechnungsgruppe beigefügt. Die Abrechnungsgruppe verfügt außerdem über zwei benutzerdefinierte Einzelposten, für die zusätzlich zu den Gesamtkosten der Abrechnungsgruppe eine Gebühr von 10 USD und eine Gebühr von 10 % erhoben werden.

JSON

```
{
   "Parameters": {
      "LinkedAccountIds": {
         "Type": "ListNumber"
      },
      "PrimaryAccountId": {
         "Type": "Number"
      }
   },
   "Resources": {
      "TestPricingRule": {
         "Type": "AWS::BillingConductor::PricingRule",
         "Properties": {
            "Name": "TestPricingRule",
            "Description": "Test pricing rule created through Cloudformation. Mark
 everything by 10%.",
            "Type": "MARKUP",
            "Scope": "GLOBAL",
            "ModifierPercentage": 10
         }
```

```
      },
      "TestPricingPlan": {
         "Type": "AWS::BillingConductor::PricingPlan",
         "Properties": {
            "Name": "TestPricingPlan",
            "Description": "Test pricing plan created through Cloudformation.",
            "PricingRuleArns": [
               {"Fn::GetAtt": ["TestPricingRule", "Arn"]}
            ]
         }
      },
      "TestBillingGroup": {
         "Type": "AWS::BillingConductor::BillingGroup",
         "Properties": {
            "Name": "TestBillingGroup",
            "Description": "Test billing group created through Cloudformation with 1
 linked account. The linked account is also the primary account.",
            "PrimaryAccountId": {
               "Ref": "PrimaryAccountId"
            },
            "AccountGrouping": {
               "LinkedAccountIds": null
            },
            "ComputationPreference": {
               "PricingPlanArn": {
                 "Fn::GetAtt": ["TestPricingPlan", "Arn"]
               }
            }
         }
      },
      "TestFlatCustomLineItem": {
         "Type": "AWS::BillingConductor::CustomLineItem",
         "Properties": {
            "Name": "TestFlatCustomLineItem",
            "Description": "Test flat custom line item created through Cloudformation
 for a $10 charge.",
            "BillingGroupArn": {
              "Fn::GetAtt": ["TestBillingGroup", "Arn"]
            },
            "CustomLineItemChargeDetails": {
               "Flat": {
                  "ChargeValue": 10
               },
               "Type": "FEE"
```

```
            }
         }
      },
      "TestPercentageCustomLineItem": {
         "Type": "AWS::BillingConductor::CustomLineItem",
         "Properties": {
            "Name": "TestPercentageCustomLineItem",
            "Description": "Test percentage custom line item created through
 Cloudformation for a %10 additional charge on the overall total bill of the billing
 group.",
            "BillingGroupArn": {
              "Fn::GetAtt": ["TestBillingGroup", "Arn"]
            },
            "CustomLineItemChargeDetails": {
               "Percentage": {
                  "PercentageValue": 10,
                  "ChildAssociatedResources": [
                     {"Fn::GetAtt": ["TestBillingGroup", "Arn"]}
                  ]
               },
               "Type": "FEE"
            }
         }
      }
   }
}
```

YAML

```
Parameters:
  LinkedAccountIds:
    Type: ListNumber
  PrimaryAccountId:
    Type: Number
Resources:
  TestPricingRule:
    Type: 'AWS::BillingConductor::PricingRule'
    Properties:
      Name: 'TestPricingRule'
      Description: 'Test pricing rule created through Cloudformation. Mark everything
 by 10%.'
      Type: 'MARKUP'
      Scope: 'GLOBAL'
```

```
      ModifierPercentage: 10
  TestPricingPlan:
    Type: 'AWS::BillingConductor::PricingPlan'
    Properties:
      Name: 'TestPricingPlan'
      Description: 'Test pricing plan created through Cloudformation.'
      PricingRuleArns:
        - !GetAtt TestPricingRule.Arn
  TestBillingGroup:
    Type: 'AWS::BillingConductor::BillingGroup'
    Properties:
      Name: 'TestBillingGroup'
      Description: 'Test billing group created through Cloudformation with 1 linked
 account. The linked account is also the primary account.'
      PrimaryAccountId: !Ref PrimaryAccountId
      AccountGrouping:
        LinkedAccountIds: !Ref LinkedAccountIds
      ComputationPreference:
        PricingPlanArn: !GetAtt TestPricingPlan.Arn
  TestFlatCustomLineItem:
    Type: 'AWS::BillingConductor::CustomLineItem'
    Properties:
      Name: 'TestFlatCustomLineItem'
      Description: 'Test flat custom line item created through Cloudformation for a $10
 charge.'
      BillingGroupArn: !GetAtt TestBillingGroup.Arn
      CustomLineItemChargeDetails:
        Flat:
          ChargeValue: 10
        Type: 'FEE'
  TestPercentageCustomLineItem:
    Type: 'AWS::BillingConductor::CustomLineItem'
    Properties:
      Name: 'TestPercentageCustomLineItem'
      Description: 'Test percentage custom line item created through Cloudformation for
 a %10 additional charge on the overall total bill of the billing group.'
      BillingGroupArn: !GetAtt TestBillingGroup.Arn
      CustomLineItemChargeDetails:
        Percentage:
          PercentageValue: 10
          ChildAssociatedResources:
            - !GetAtt TestBillingGroup.Arn
        Type: 'FEE'
```

AWS CloudFormation Schnipsel aus Vorlagen

Themen

- Verschachtelte Stacks
- Wartebedingung

Verschachtelte Stacks

Verschachteln eines Stacks in einer Vorlage

Diese Beispielvorlage enthält eine geschachtelte Stack-Ressource mit dem Namen `myStack`. Wenn AWS CloudFormation aus der Vorlage ein Stapel erstellt wird, wird der erstelltmy`Stack`, dessen Vorlage in der `TemplateURL` Eigenschaft angegeben ist. Der Ausgabewert `StackRef` gibt die Stack-ID für `myStack` zurück und der Wert `OutputFromNestedStack` gibt den Ausgabewert `BucketName` innerhalb der `myStack`-Ressource zurück. Das `Outputs.`*`nestedstackoutputname`* Format ist für die Angabe von Ausgabewerten aus verschachtelten Stapeln reserviert und kann überall in der enthaltenden Vorlage verwendet werden.

Weitere Informationen finden Sie unter AWS::CloudFormation::Stack.

JSON

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myStack" : {
        "Type" : "AWS::CloudFormation::Stack",
        "Properties" : {
         "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-1/
S3_Bucket.template",
              "TimeoutInMinutes" : "60"
        }
        }
    },
    "Outputs": {
       "StackRef": {"Value": { "Ref" : "myStack"}},
       "OutputFromNestedStack" : {
             "Value" : { "Fn::GetAtt" : [ "myStack", "Outputs.BucketName" ] }
       }
    }
```

```
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: https://s3.amazonaws.com/cloudformation-templates-us-east-1/
S3_Bucket.template
      TimeoutInMinutes: '60'
Outputs:
  StackRef:
    Value: !Ref myStack
  OutputFromNestedStack:
    Value: !GetAtt myStack.Outputs.BucketName
```

Verschachteln eines Stacks mit Eingabeparametern in einer Vorlage

Diese Beispielvorlage enthält eine Stack-Ressource, die Eingabeparameter angibt. Wenn aus dieser Vorlage ein Stapel AWS CloudFormation erstellt wird, werden die in der `Parameters` Eigenschaft deklarierten Wertepaare als Eingabeparameter für die Vorlage verwendet, mit der der Stapel erstellt wurde. `myStackWithParams` In diesem Beispiel wurden die Parameter `InstanceType` und `KeyName` angegeben.

Weitere Informationen finden Sie unter AWS::CloudFormation::Stack.

JSON

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myStackWithParams" : {
          "Type" : "AWS::CloudFormation::Stack",
        "Properties" : {
            "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-
east-1/EC2ChooseAMI.template",
            "Parameters" : {
                "InstanceType" : "t2.micro",
                "KeyName" : "mykey"
            }
```

```
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myStackWithParams:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: https://s3.amazonaws.com/cloudformation-templates-us-east-1/
EC2ChooseAMI.template
      Parameters:
        InstanceType: t2.micro
        KeyName: mykey
```

Wartebedingung

Verwenden einer Wartebedingung mit einer EC2 Amazon-Instance

> **Important**
>
> Für Amazon EC2 - und Auto Scaling Scaling-Ressourcen empfehlen wir, anstelle von Wartebedingungen ein CreationPolicy Attribut zu verwenden. Fügen Sie diesen Ressourcen ein CreationPolicy Attribut hinzu und verwenden Sie das cfn-signal-Hilfsskript, um zu signalisieren, dass ein Instance-Erstellungsprozess erfolgreich abgeschlossen wurde.

Wenn Sie keine Erstellungsrichtlinie verwenden können, sehen Sie sich die folgende Beispielvorlage an, die eine EC2 Amazon-Instance mit einer Wartebedingung deklariert. Die `myWaitCondition` Wartebedingung wird `myWaitConditionHandle` zur Signalisierung verwendet, verwendet das `DependsOn` Attribut, um anzugeben, dass die Wartebedingung ausgelöst wird, nachdem die EC2 Amazon-Instance-Ressource erstellt wurde, und verwendet die `Timeout` Eigenschaft, um eine Dauer von 4500 Sekunden für die Wartebedingung anzugeben. Darüber hinaus wird die vorsignierte URL, die die Wartebedingung signalisiert, mit der `UserData` Eigenschaft der `Ec2Instance` Ressource an die EC2 Amazon-Instance übergeben, sodass eine Anwendung oder ein Skript, das auf dieser EC2 Amazon-Instance ausgeführt wird, die vorsignierte URL abrufen und verwenden kann, um der Wartebedingung einen Erfolg oder Misserfolg zu signalisieren. Sie müssen die Anwendung oder das

Skript verwenden `cfn-signal` oder erstellen, das die Wartebedingung signalisiert. Der Ausgabewert `ApplicationData` enthält die Daten, die vom Wartebedingungssignal zurückgegeben wurden.

Weitere Informationen finden Sie unter Wartebedingungen in einer CloudFormation Vorlage erstellen.

JSON

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Mappings" : {
        "RegionMap" : {
            "us-east-1" : {
                "AMI" : "ami-0ff8a91507f77f867"
            },
            "us-west-1" : {
                "AMI" : "ami-0bdb828fd58c52235"
            },
            "eu-west-1" : {
                "AMI" : "ami-047bb4163c506cd98"
            },
            "ap-northeast-1" : {
                "AMI" : "ami-06cd52961ce9f0d85"
            },
            "ap-southeast-1" : {
                "AMI" : "ami-08569b978cc4dfa10"
            }
        }
    },
    "Resources" : {
        "Ec2Instance" : {
            "Type" : "AWS::EC2::Instance",
            "Properties" : {
                "UserData" : { "Fn::Base64" : {"Ref" : "myWaitHandle"}},
                "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" :
 "AWS::Region" }, "AMI" ]}
            }
        },
        "myWaitHandle" : {
            "Type" : "AWS::CloudFormation::WaitConditionHandle",
            "Properties" : {
            }
        },
        "myWaitCondition" : {
            "Type" : "AWS::CloudFormation::WaitCondition",
```

```
            "DependsOn" : "Ec2Instance",
            "Properties" : {
                "Handle" : { "Ref" : "myWaitHandle" },
                "Timeout" : "4500"
            }
        }
    },
    "Outputs" : {
        "ApplicationData" : {
            "Value" : { "Fn::GetAtt" : [ "myWaitCondition", "Data" ]},
            "Description" : "The data passed back as part of signalling the
 WaitCondition."
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-0ff8a91507f77f867
    us-west-1:
      AMI: ami-0bdb828fd58c52235
    eu-west-1:
      AMI: ami-047bb4163c506cd98
    ap-northeast-1:
      AMI: ami-06cd52961ce9f0d85
    ap-southeast-1:
      AMI: ami-08569b978cc4dfa10
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      UserData:
        Fn::Base64: !Ref myWaitHandle
      ImageId:
        Fn::FindInMap:
        - RegionMap
        - Ref: AWS::Region
        - AMI
  myWaitHandle:
```

```
    Type: AWS::CloudFormation::WaitConditionHandle
    Properties: {}
  myWaitCondition:
    Type: AWS::CloudFormation::WaitCondition
    DependsOn: Ec2Instance
    Properties:
      Handle: !Ref myWaitHandle
      Timeout: '4500'
Outputs:
  ApplicationData:
    Value: !GetAtt myWaitCondition.Data
    Description: The data passed back as part of signalling the WaitCondition.
```

Verwenden des cfn-signal-Hilfsskript zum Signalisieren einer Wartebedingung

Dieses Beispiel zeigt eine `cfn-signal` Befehlszeile, die einer Wartebedingung den Erfolg signalisiert. Sie müssen die Befehlszeile in der `UserData` Eigenschaft der EC2 Instanz definieren.

JSON

```
"UserData": {
  "Fn::Base64": {
    "Fn::Join": [
      "",
      [
         "#!/bin/bash -xe\n",
         "/opt/aws/bin/cfn-signal --exit-code 0 '",
         {
           "Ref": "myWaitHandle"
         },
         "'\n"
      ]
    ]
  }
}
```

YAML

```
UserData:
  'Fn::Base64':
    'Fn::Join':
      - ''
```

```
      - - |
          #!/bin/bash -xe
        - /opt/aws/bin/cfn-signal --exit-code 0 '
        - Ref: myWaitHandle
        - |
          '
```

Erstellen eines Signal für eine Wartebedingung mit Curl

Das folgende Beispiel zeigt eine Curl-Befehlszeile, die für eine Wartebedingung einen Erfolg signalisiert.

```
curl -T /tmp/a "https://cloudformation-waitcondition-test.s3.amazonaws.com/
arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack
%2Fstack-gosar-20110427004224-test-stack-with-WaitCondition--VEYW
%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wyOoRejVoo
%3D"
```

Wobei die Datei /tmp/a die folgenden JSON-Struktur enthält:

```
{
  "Status" : "SUCCESS",
  "Reason" : "Configuration Complete",
  "UniqueId" : "ID1234",
  "Data" : "Application has completed configuration."
}
```

Das folgende Beispiel zeigt eine Curl-Befehlszeile, die dasselbe Erfolgssignal sendet, mit der Ausnahme, dass die JSON als Parameter in der Befehlszeile gesendet wird.

```
curl -X PUT -H 'Content-Type:' --data-binary '{"Status" : "SUCCESS","Reason" :
 "Configuration Complete","UniqueId" : "ID1234","Data" : "Application
 has completed configuration."}' "https://cloudformation-waitcondition-
test.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack
%2Fstack-gosar-20110427004224-test-stack-with-WaitCondition--VEYW
%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wyOoRejVoo
%3D"
```

CloudFront Amazon-Vorlagenschnipsel

Verwenden Sie diese Beispielvorlagenausschnitte zusammen mit Ihrer CloudFront Amazon-Vertriebsressource in. AWS CloudFormation Weitere Informationen finden Sie in der Referenz zum CloudFront Amazon-Ressourcentyp.

Themen

- CloudFront Amazon-Vertriebsressource mit Amazon S3-Ursprung
- CloudFront Amazon-Vertriebsressource mit benutzerdefiniertem Ursprung
- CloudFront Amazon-Vertrieb mit Unterstützung mehrerer Absender
- CloudFront Amazon-Vertrieb mit einer Lambda-Funktion als Ursprung
- Weitere Informationen finden Sie auch unter

CloudFront Amazon-Vertriebsressource mit Amazon S3-Ursprung

Die folgende Beispielvorlage zeigt eine CloudFront Amazon-Distribution, die eine S3Origin und eine ältere Origin-Zugriffsidentität (OAI) verwendet. Informationen zur Verwendung von Origin Access Control (OAC) stattdessen finden Sie unter Beschränken des Zugriffs auf einen Amazon Simple Storage Service-Ursprung im Amazon CloudFront Developer Guide.

JSON

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myDistribution" : {
            "Type" : "AWS::CloudFront::Distribution",
            "Properties" : {
                "DistributionConfig" : {
                    "Origins" : [ {
                        "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "Id" : "myS3Origin",
                        "S3OriginConfig" : {
                            "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
                        }
                    }],
                    "Enabled" : "true",
                    "Comment" : "Some comment",
```

```
                    "DefaultRootObject" : "index.html",
                    "Logging" : {
                        "IncludeCookies" : "false",
                        "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
                        "Prefix" : "myprefix"
                    },
                    "Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
                    "DefaultCacheBehavior" : {
                        "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
 "PATCH", "POST", "PUT" ],
                        "TargetOriginId" : "myS3Origin",
                        "ForwardedValues" : {
                            "QueryString" : "false",
                            "Cookies" : { "Forward" : "none" }
                        },
                        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
                        "ViewerProtocolPolicy" : "allow-all"
                    },
                   "PriceClass" : "PriceClass_200",
                   "Restrictions" : {
                       "GeoRestriction" : {
                           "RestrictionType" : "whitelist",
                           "Locations" : [ "AQ", "CV" ]
                       }
                   },
                   "ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }
               }
           }
       }
   }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
        - DomainName: amzn-s3-demo-bucket.s3.amazonaws.com
          Id: myS3Origin
```

```
          S3OriginConfig:
            OriginAccessIdentity: origin-access-identity/cloudfront/E127EXAMPLE51Z
        Enabled: 'true'
        Comment: Some comment
        DefaultRootObject: index.html
        Logging:
          IncludeCookies: 'false'
          Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
          Prefix: myprefix
        Aliases:
        - mysite.example.com
        - yoursite.example.com
        DefaultCacheBehavior:
          AllowedMethods:
          - DELETE
          - GET
          - HEAD
          - OPTIONS
          - PATCH
          - POST
          - PUT
          TargetOriginId: myS3Origin
          ForwardedValues:
            QueryString: 'false'
            Cookies:
              Forward: none
          TrustedSigners:
          - 1234567890EX
          - 1234567891EX
          ViewerProtocolPolicy: allow-all
        PriceClass: PriceClass_200
        Restrictions:
          GeoRestriction:
            RestrictionType: whitelist
            Locations:
            - AQ
            - CV
        ViewerCertificate:
          CloudFrontDefaultCertificate: 'true'
```

CloudFront Amazon-Vertriebsressource mit benutzerdefiniertem Ursprung

Die folgende Beispielvorlage zeigt eine CloudFront Amazon-Distribution mit einem CustomOrigin.

JSON

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myDistribution" : {
            "Type" : "AWS::CloudFront::Distribution",
            "Properties" : {
                "DistributionConfig" : {
                    "Origins" : [ {
                            "DomainName" : "www.example.com",
                            "Id" : "myCustomOrigin",
                            "CustomOriginConfig" : {
                                "HTTPPort" : "80",
                                "HTTPSPort" : "443",
                                "OriginProtocolPolicy" : "http-only"
                            }
                    } ],
                    "Enabled" : "true",
                    "Comment" : "Somecomment",
                    "DefaultRootObject" : "index.html",
                    "Logging" : {
                        "IncludeCookies" : "true",
                        "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
                        "Prefix": "myprefix"
                    },
                    "Aliases" : [
                        "mysite.example.com",
                        "*.yoursite.example.com"
                    ],
                    "DefaultCacheBehavior" : {
                        "TargetOriginId" : "myCustomOrigin",
                        "SmoothStreaming" : "false",
                        "ForwardedValues" : {
                            "QueryString" : "false",
                            "Cookies" : { "Forward" : "all" }
                        },
                        "TrustedSigners" : [
                            "1234567890EX",
                            "1234567891EX"
                        ],
                        "ViewerProtocolPolicy" : "allow-all"
                    },
                    "CustomErrorResponses" : [ {
```

```
                        "ErrorCode" : "404",
                        "ResponsePagePath" : "/error-pages/404.html",
                        "ResponseCode" : "200",
                        "ErrorCachingMinTTL" : "30"
                    } ],
                    "PriceClass" : "PriceClass_200",
                    "Restrictions" : {
                        "GeoRestriction" : {
                            "RestrictionType" : "whitelist",
                            "Locations" : [ "AQ", "CV" ]
                        }
                    },
                    "ViewerCertificate": { "CloudFrontDefaultCertificate" : "true" }
                }
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
        - DomainName: www.example.com
          Id: myCustomOrigin
          CustomOriginConfig:
            HTTPPort: '80'
            HTTPSPort: '443'
            OriginProtocolPolicy: http-only
        Enabled: 'true'
        Comment: Somecomment
        DefaultRootObject: index.html
        Logging:
          IncludeCookies: 'true'
          Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
          Prefix: myprefix
        Aliases:
        - mysite.example.com
```

```
        - "*.yoursite.example.com"
        DefaultCacheBehavior:
          TargetOriginId: myCustomOrigin
          SmoothStreaming: 'false'
          ForwardedValues:
            QueryString: 'false'
            Cookies:
              Forward: all
          TrustedSigners:
          - 1234567890EX
          - 1234567891EX
          ViewerProtocolPolicy: allow-all
        CustomErrorResponses:
        - ErrorCode: '404'
          ResponsePagePath: "/error-pages/404.html"
          ResponseCode: '200'
          ErrorCachingMinTTL: '30'
        PriceClass: PriceClass_200
        Restrictions:
          GeoRestriction:
            RestrictionType: whitelist
            Locations:
            - AQ
            - CV
        ViewerCertificate:
          CloudFrontDefaultCertificate: 'true'
```

CloudFront Amazon-Vertrieb mit Unterstützung mehrerer Absender

Die folgende Beispielvorlage zeigt, wie Sie eine CloudFront Distribution mit Unterstützung mehrerer Herkunft deklarieren. In der DistributionConfigwird eine Liste von Ursprüngen bereitgestellt und DefaultCacheBehaviorein festgelegt.

JSON

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myDistribution" : {
            "Type" : "AWS::CloudFront::Distribution",
            "Properties" : {
                "DistributionConfig" : {
                    "Origins" : [ {
```

```
                        "Id" : "myS3Origin",
                        "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "S3OriginConfig" : {
                            "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
                        }
                      },
                      {
                          "Id" : "myCustomOrigin",
                          "DomainName" : "www.example.com",
                          "CustomOriginConfig" : {
                              "HTTPPort" : "80",
                              "HTTPSPort" : "443",
                              "OriginProtocolPolicy" : "http-only"
                          }
                      }
                    ],
                    "Enabled" : "true",
                    "Comment" : "Some comment",
                    "DefaultRootObject" : "index.html",
                    "Logging" : {
                        "IncludeCookies" : "true",
                        "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
                        "Prefix" : "myprefix"
                    },
                    "Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
                    "DefaultCacheBehavior" : {
                        "TargetOriginId" : "myS3Origin",
                        "ForwardedValues" : {
                            "QueryString" : "false",
                            "Cookies" : { "Forward" : "all" }
                         },
                        "TrustedSigners" : [ "1234567890EX", "1234567891EX"  ],
                        "ViewerProtocolPolicy" : "allow-all",
                        "MinTTL" : "100",
                        "SmoothStreaming" : "true"
                    },
                    "CacheBehaviors" : [ {
                             "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
 "PATCH", "POST", "PUT" ],
                             "TargetOriginId" : "myS3Origin",
                             "ForwardedValues" : {
                                 "QueryString" : "true",
                                 "Cookies" : { "Forward" : "none" }
```

```
                            },
                            "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
                            "ViewerProtocolPolicy" : "allow-all",
                            "MinTTL" : "50",
                            "PathPattern" : "images1/*.jpg"
                        },
                        {
                            "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
 "PATCH", "POST", "PUT" ],
                            "TargetOriginId" : "myCustomOrigin",
                            "ForwardedValues" : {
                                "QueryString" : "true",
                                "Cookies" : { "Forward" : "none" }
                            },
                            "TrustedSigners" : [ "1234567890EX", "1234567891EX"  ],
                            "ViewerProtocolPolicy" : "allow-all",
                            "MinTTL" : "50",
                            "PathPattern" : "images2/*.jpg"
                        }
                    ],
                    "CustomErrorResponses" : [ {
                        "ErrorCode" : "404",
                        "ResponsePagePath" : "/error-pages/404.html",
                        "ResponseCode" : "200",
                        "ErrorCachingMinTTL" : "30"
                    } ],
                    "PriceClass" : "PriceClass_All",
                    "ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }
                }
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
```

```
        - Id: myS3Origin
          DomainName: amzn-s3-demo-bucket.s3.amazonaws.com
          S3OriginConfig:
            OriginAccessIdentity: origin-access-identity/cloudfront/E127EXAMPLE51Z
        - Id: myCustomOrigin
          DomainName: www.example.com
          CustomOriginConfig:
            HTTPPort: '80'
            HTTPSPort: '443'
            OriginProtocolPolicy: http-only
        Enabled: 'true'
        Comment: Some comment
        DefaultRootObject: index.html
        Logging:
          IncludeCookies: 'true'
          Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
          Prefix: myprefix
        Aliases:
        - mysite.example.com
        - yoursite.example.com
        DefaultCacheBehavior:
          TargetOriginId: myS3Origin
          ForwardedValues:
            QueryString: 'false'
            Cookies:
              Forward: all
          TrustedSigners:
          - 1234567890EX
          - 1234567891EX
          ViewerProtocolPolicy: allow-all
          MinTTL: '100'
          SmoothStreaming: 'true'
        CacheBehaviors:
        - AllowedMethods:
          - DELETE
          - GET
          - HEAD
          - OPTIONS
          - PATCH
          - POST
          - PUT
          TargetOriginId: myS3Origin
          ForwardedValues:
            QueryString: 'true'
```

```
            Cookies:
              Forward: none
          TrustedSigners:
          - 1234567890EX
          - 1234567891EX
          ViewerProtocolPolicy: allow-all
          MinTTL: '50'
          PathPattern: images1/*.jpg
        - AllowedMethods:
          - DELETE
          - GET
          - HEAD
          - OPTIONS
          - PATCH
          - POST
          - PUT
          TargetOriginId: myCustomOrigin
          ForwardedValues:
            QueryString: 'true'
            Cookies:
              Forward: none
          TrustedSigners:
          - 1234567890EX
          - 1234567891EX
          ViewerProtocolPolicy: allow-all
          MinTTL: '50'
          PathPattern: images2/*.jpg
        CustomErrorResponses:
        - ErrorCode: '404'
          ResponsePagePath: "/error-pages/404.html"
          ResponseCode: '200'
          ErrorCachingMinTTL: '30'
        PriceClass: PriceClass_All
        ViewerCertificate:
          CloudFrontDefaultCertificate: 'true'
```

CloudFront Amazon-Vertrieb mit einer Lambda-Funktion als Ursprung

Im folgenden Beispiel wird eine CloudFront Distribution erstellt, die einer angegebenen Lambda-Funktions-URL (als Parameter bereitgestellt) vorangestellt wird, wodurch reiner HTTPS-Zugriff, Caching, Komprimierung und globale Bereitstellung ermöglicht werden. Es konfiguriert die Lambda-URL als benutzerdefinierten HTTPS-Ursprung und wendet eine AWS Standard-Caching-Richtlinie an. Die Distribution ist für die Leistung mit HTTP/2 und IPv6 Unterstützung optimiert und gibt den

CloudFront Domainnamen aus, sodass Benutzer über einen sicheren, CDN-gestützten Endpunkt auf die Lambda-Funktion zugreifen können. Weitere Informationen finden Sie im AWS Blog unter Verwenden von Amazon CloudFront mit AWS Lambda als Ursprung zur Beschleunigung Ihrer Webanwendungen.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Parameters": {
        "LambdaEndpoint": {
            "Type": "String",
            "Description": "The Lambda function URL endpoint without the 'https://'"
        }
    },
    "Resources": {
        "MyDistribution": {
            "Type": "AWS::CloudFront::Distribution",
            "Properties": {
                "DistributionConfig": {
                    "PriceClass": "PriceClass_All",
                    "HttpVersion": "http2",
                    "IPV6Enabled": true,
                    "Origins": [
                        {
                            "DomainName": {
                                "Ref": "LambdaEndpoint"
                            },
                            "Id": "LambdaOrigin",
                            "CustomOriginConfig": {
                                "HTTPSPort": 443,
                                "OriginProtocolPolicy": "https-only"
                            }
                        }
                    ],
                    "Enabled": "true",
                    "DefaultCacheBehavior": {
                        "TargetOriginId": "LambdaOrigin",
                        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e58f6",
                        "ViewerProtocolPolicy": "redirect-to-https",
                        "SmoothStreaming": "false",
                        "Compress": "true"
                    }
```

```
                }
            }
        }
    },
    "Outputs": {
        "CloudFrontDomain": {
            "Description": "CloudFront default domain name configured",
            "Value": {
                "Fn::Sub": "https://${MyDistribution.DomainName}/"
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  LambdaEndpoint:
    Type: String
    Description: The Lambda function URL endpoint without the 'https://'
Resources:
  MyDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        PriceClass: PriceClass_All
        HttpVersion: http2
        IPV6Enabled: true
        Origins:
        - DomainName: !Ref LambdaEndpoint
          Id: LambdaOrigin
          CustomOriginConfig:
            HTTPSPort: 443
            OriginProtocolPolicy: https-only
        Enabled: 'true'
        DefaultCacheBehavior:
          TargetOriginId: LambdaOrigin
          CachePolicyId: '658327ea-f89d-4fab-a63d-7e88639e58f6'
          ViewerProtocolPolicy: redirect-to-https
          SmoothStreaming: 'false'
          Compress: 'true'
Outputs:
```

```
  CloudFrontDomain:
    Description: CloudFront default domain name configured
    Value: !Sub https://${MyDistribution.DomainName}/
```

Weitere Informationen finden Sie auch unter

Ein Beispiel für das Hinzufügen eines benutzerdefinierten Alias zu einem Route 53-Datensatz, um einen benutzerfreundlichen Namen für eine CloudFront Distribution zu erstellen, finden Sie unterAlias-Ressourcendatensatz für eine Verteilung CloudFront .

CloudWatch Amazon-Vorlagenschnipsel

Verwenden Sie diese Beispielvorlagenausschnitte, um Ihre CloudWatch Amazon-Ressourcen in zu beschreiben. AWS CloudFormation Weitere Informationen finden Sie in der Referenz zum CloudWatch Amazon-Ressourcentyp.

Themen

- Fakturierungsbenachrichtigung
- CPU-Nutzungsalarm
- Wiederherstellung einer Amazon Elastic Compute Cloud-Instance
- Erstellen eines grundlegenden Dashboards
- Erstellen Sie ein Dashboard mit side-by-side Widgets

Fakturierungsbenachrichtigung

Im folgenden Beispiel CloudWatch sendet Amazon eine E-Mail-Benachrichtigung, wenn Belastungen auf Ihrem AWS Konto den Alarmschwellenwert überschreiten. Um Nutzungsbenachrichtigungen zu erhalten, aktivieren Sie Fakturierungsbenachrichtigungen. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch unter Einen Abrechnungsalarm erstellen, um Ihre geschätzten AWS Gebühren zu überwachen. >

JSON

```
"SpendingAlarm": {
  "Type": "AWS::CloudWatch::Alarm",
  "Properties": {
    "AlarmDescription": { "Fn::Join": ["", [
      "Alarm if AWS spending is over $",
```

```
      { "Ref": "AlarmThreshold" }
    ]]},
    "Namespace": "AWS/Billing",
    "MetricName": "EstimatedCharges",
    "Dimensions": [{
      "Name": "Currency",
      "Value" : "USD"
    }],
    "Statistic": "Maximum",
    "Period": "21600",
    "EvaluationPeriods": "1",
    "Threshold": { "Ref": "AlarmThreshold" },
    "ComparisonOperator": "GreaterThanThreshold",
    "AlarmActions": [{
      "Ref": "BillingAlarmNotification"
    }],
    "InsufficientDataActions": [{
      "Ref": "BillingAlarmNotification"
    }]
  }
}
```

YAML

```
SpendingAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription:
      'Fn::Join':
        - ''
        - - Alarm if AWS spending is over $
          - !Ref: AlarmThreshold
    Namespace: AWS/Billing
    MetricName: EstimatedCharges
    Dimensions:
    - Name: Currency
      Value: USD
    Statistic: Maximum
    Period: '21600'
    EvaluationPeriods: '1'
    Threshold:
      !Ref: "AlarmThreshold"
    ComparisonOperator: GreaterThanThreshold
```

```
    AlarmActions:
    - !Ref: "BillingAlarmNotification"
    InsufficientDataActions:
    - !Ref: "BillingAlarmNotification"
```

CPU-Nutzungsalarm

Der folgende Beispielausschnitt erzeugt einen Alarm, der eine Benachrichtigung sendet, wenn die durchschnittliche CPU-Auslastung einer EC2 Amazon-Instance in drei Testzeiträumen mehr als 60 Sekunden lang 90 Prozent überschreitet.

JSON

```
"CPUAlarm" : {
  "Type" : "AWS::CloudWatch::Alarm",
  "Properties" : {
    "AlarmDescription" : "CPU alarm for my instance",
    "AlarmActions" : [ { "Ref" : "logical name of an AWS::SNS::Topic resource" } ],
    "MetricName" : "CPUUtilization",
    "Namespace" : "AWS/EC2",
    "Statistic" : "Average",
    "Period" : "60",
    "EvaluationPeriods" : "3",
    "Threshold" : "90",
    "ComparisonOperator" : "GreaterThanThreshold",
    "Dimensions" : [ {
      "Name" : "InstanceId",
      "Value" : { "Ref" : "logical name of an AWS::EC2::Instance resource" }
    } ]
  }
}
```

YAML

```
CPUAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: CPU alarm for my instance
    AlarmActions:
    - !Ref: "logical name of an AWS::SNS::Topic resource"
    MetricName: CPUUtilization
    Namespace: AWS/EC2
    Statistic: Average
```

```
    Period: '60'
    EvaluationPeriods: '3'
    Threshold: '90'
    ComparisonOperator: GreaterThanThreshold
    Dimensions:
    - Name: InstanceId
      Value: !Ref: "logical name of an AWS::EC2::Instance resource"
```

Wiederherstellung einer Amazon Elastic Compute Cloud-Instance

Der folgende CloudWatch Alarm stellt eine EC2 Instance wieder her, wenn die Statusüberprüfung 15 Minuten hintereinander fehlschlägt. Weitere Informationen zu Alarmaktionen finden Sie unter Erstellen von Alarmen zum Stoppen, Beenden, Neustarten oder Wiederherstellen einer EC2 Instance im CloudWatch Amazon-Benutzerhandbuch.

JSON

```
{
  "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": "Trigger a recovery 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:aws:automate:", { "Ref" :
 "AWS::Region" }, ":ec2:recover" ]]} ],
        "Dimensions": [{"Name": "InstanceId","Value": {"Ref": "RecoveryInstance"}}]
      }
    }
```

```
  }
}
```

YAML

```
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: Trigger a recovery 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: [ !Sub "arn:aws:automate:${AWS::Region}:ec2:recover" ]
      Dimensions:
      - Name: InstanceId
        Value: !Ref: RecoveryInstance
```

Erstellen eines grundlegenden Dashboards

Das folgende Beispiel erstellt ein einfaches CloudWatch Dashboard mit einem Metrik-Widget, das die CPU-Auslastung anzeigt, und einem Text-Widget, das eine Meldung anzeigt.

JSON

```
{
    "BasicDashboard": {
        "Type": "AWS::CloudWatch::Dashboard",
        "Properties": {
            "DashboardName": "Dashboard1",
            "DashboardBody": "{\"widgets\":[{\"type\":\"metric\",\"x\":0,\"y\":0,
\"width\":12,\"height\":6,\"properties\":{\"metrics\":[[\"AWS/EC2\",\"CPUUtilization
\",\"InstanceId\",\"i-012345\"]],\"period\":300,\"stat\":\"Average\",\"region\":\"us-
```

```
east-1\",\"title\":\"EC2 Instance CPU\"}},{\"type\":\"text\",\"x\":0,\"y\":7,\"width
\":3,\"height\":3,\"properties\":{\"markdown\":\"Hello world\"}}]}"
        }
    }
}
```

YAML

```
BasicDashboard:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
    DashboardBody: '{"widgets":
[{"type":"metric","x":0,"y":0,"width":12,"height":6,"properties":{"metrics":[["AWS/
EC2","CPUUtilization","InstanceId","i-012345"]],"period":300,"stat":"Average","region":"us-
east-1","title":"EC2 Instance CPU"}},
{"type":"text","x":0,"y":7,"width":3,"height":3,"properties":{"markdown":"Hello
 world"}}]}'
```

Erstellen Sie ein Dashboard mit side-by-side Widgets

Im folgenden Beispiel wird ein Dashboard mit zwei Metrik-Widgets erstellt, die nebeneinander angezeigt werden.

JSON

```
{
    "DashboardSideBySide": {
        "Type": "AWS::CloudWatch::Dashboard",
        "Properties": {
            "DashboardName": "Dashboard1",
            "DashboardBody": "{\"widgets\":[{\"type\":\"metric\",\"x\":0,\"y\":0,
\"width\":12,\"height\":6,\"properties\":{\"metrics\":[[\"AWS/EC2\",\"CPUUtilization
\",\"InstanceId\",\"i-012345\"]],\"period\":300,\"stat\":\"Average\",\"region\":
\"us-east-1\",\"title\":\"EC2 Instance CPU\"}},{\"type\":\"metric\",\"x\":12,\"y\":0,
\"width\":12,\"height\":6,\"properties\":{\"metrics\":[[\"AWS/S3\",\"BucketSizeBytes\",
\"BucketName\",\"amzn-s3-demo-bucket\"]],\"period\":86400,\"stat\":\"Maximum\",\"region
\":\"us-east-1\",\"title\":\"amzn-s3-demo-bucket bytes\"}}]}"
        }
    }
}
```

YAML

```
DashboardSideBySide:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
    DashboardBody: '{"widgets":
[{"type":"metric","x":0,"y":0,"width":12,"height":6,"properties":{"metrics":[["AWS/
EC2","CPUUtilization","InstanceId","i-012345"]],"period":300,"stat":"Average","region":"us-
east-1","title":"EC2 Instance CPU"}},
{"type":"metric","x":12,"y":0,"width":12,"height":6,"properties":
{"metrics":[["AWS/S3","BucketSizeBytes","BucketName","amzn-s3-demo-
bucket"]],"period":86400,"stat":"Maximum","region":"us-east-1","title":"amzn-s3-demo-
bucket bytes"}}]}'
```

Amazon CloudWatch Logs-Vorlagenausschnitte

Amazon CloudWatch Logs kann Ihr System, Ihre Anwendung und Ihre benutzerdefinierten Protokolldateien von EC2 Amazon-Instances oder anderen Quellen überwachen. Sie können AWS CloudFormation es verwenden, um Protokollgruppen und Metrikfilter bereitzustellen und zu verwalten. Weitere Informationen zu Amazon CloudWatch Logs finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

Themen

- Senden Sie Logs von einer Linux-Instance an CloudWatch Logs
- Sendet CloudWatch Protokolle von einer Windows-Instanz an Logs
- Weitere Informationen finden Sie auch unter

Senden Sie Logs von einer Linux-Instance an CloudWatch Logs

In der folgenden Vorlage werden ein Webserver und dessen benutzerdefinierte Metriken beschrieben. Protokollereignisse aus dem Protokoll des Webservers liefern die Daten für die benutzerdefinierten Metriken. Um Protokollereignisse an eine benutzerdefinierte Metrik zu senden, installiert das `UserData` Feld einen CloudWatch Logs-Agenten auf der EC2 Amazon-Instance. Die Konfigurationsinformationen für den Agenten, wie z. B. der Speicherort der Serverprotokolldatei, der Protokollgruppenname und der Name des Protokollstreams, werden in der Datei `/tmp/cwlogs/`

`apacheaccess.conf` definiert. Der Protokollstream wird erstellt, nachdem der Webserver mit dem Senden von Protokollereignissen an die Datei `/var/log/httpd/access_log` begonnen hat.

> **Note**
>
> Ein Hinweis zu Berechtigungen: Die `WebServerHost` Instance referenziert das `LogRoleInstanceProfile` Instance-Profil, das wiederum die `LogRole` Rolle referenziert. `LogRole` gibt die `s3:GetObject`-Berechtigung für arn:aws:s3:::* an.
> Diese Berechtigung ist erforderlich, da der CloudWatch Logs-Agent (`awslogs-agent-setup.py`) von Amazon S3 im `UserData` Abschnitt `WebServerHost` heruntergeladen wird.

Die beiden Metrikfilter beschreiben, wie die Protokollinformationen in CloudWatch-Metriken verwandelt werden. Die 404-Metrik zählt die Zahl von 404-Vorkommnissen. Die Größenmetrik verfolgt die Größe einer Anforderung. Die beiden CloudWatch Alarme senden Benachrichtigungen, wenn innerhalb von 2 Minuten mehr als zwei 404-Sekunden eintreffen oder wenn die durchschnittliche Anforderungsgröße innerhalb von 10 Minuten über 3500 KB liegt.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "AWS CloudFormation Sample Template for CloudWatch Logs.",
    "Parameters": {
        "KeyName": {
            "Description": "Name of an existing EC2 KeyPair to enable SSH access to the
 instances",
            "Type": "AWS::EC2::KeyPair::KeyName",
            "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
        },
        "SSHLocation": {
            "Description": "The IP address range that can be used to SSH to the EC2
 instances",
            "Type": "String",
            "MinLength": "9",
            "MaxLength": "18",
            "Default": "0.0.0.0/0",
            "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\
\d{1,2})",
            "ConstraintDescription": "must be a valid IP CIDR range of the form
 x.x.x.x/x."
```

```
        },
        "OperatorEmail": {
            "Description": "Email address to notify if there are any scaling
 operations",
            "Type": "String"
        }
    },
    "Mappings": {
        "RegionMap": {
            "us-east-1": {
                "AMI": "ami-0ff8a91507f77f867"
            },
            "us-west-1": {
                "AMI": "ami-0bdb828fd58c52235"
            },
            "us-west-2": {
                "AMI": "ami-a0cfeed8"
            },
            "eu-west-1": {
                "AMI": "ami-047bb4163c506cd98"
            },
            "ap-southeast-1": {
                "AMI": "ami-08569b978cc4dfa10"
            },
            "ap-southeast-2": {
                "AMI": "ami-09b42976632b27e9b"
            },
            "ap-northeast-1": {
                "AMI": "ami-06cd52961ce9f0d85"
            },
            "sa-east-1": {
                "AMI": "ami-07b14488da8ea02a0"
            },
            "eu-central-1": {
                "AMI": "ami-0233214e13e500f77"
            }
        }
    },
    "Resources": {
        "LogRole": {
            "Type": "AWS::IAM::Role",
            "Properties": {
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
```

```
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": [
                                "ec2.amazonaws.com"
                            ]
                        },
                        "Action": [
                            "sts:AssumeRole"
                        ]
                    }
                ]
            },
            "Path": "/",
            "Policies": [
                {
                    "PolicyName": "LogRolePolicy",
                    "PolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Action": [
                                    "logs:Create*",
                                    "logs:PutLogEvents",
                                    "s3:GetObject"
                                ],
                                "Resource": [
                                    "arn:aws:logs:*:*:*",
                                    "arn:aws:s3:::*"
                                ]
                            }
                        ]
                    }
                }
            ]
        }
    },
    "LogRoleInstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",
        "Properties": {
            "Path": "/",
            "Roles": [
```

```
                    {
                        "Ref": "LogRole"
                    }
                ]
            }
        },
        "WebServerSecurityGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "Enable HTTP access via port 80 and SSH access via
 port 22",
                "SecurityGroupIngress": [
                    {
                        "IpProtocol": "tcp",
                        "FromPort": 80,
                        "ToPort": 80,
                        "CidrIp": "0.0.0.0/0"
                    },
                    {
                        "IpProtocol": "tcp",
                        "FromPort": 22,
                        "ToPort": 22,
                        "CidrIp": {
                            "Ref": "SSHLocation"
                        }
                    }
                ]
            }
        },
        "WebServerHost": {
            "Type": "AWS::EC2::Instance",
            "Metadata": {
                "Comment": "Install a simple PHP application",
                "AWS::CloudFormation::Init": {
                    "config": {
                        "packages": {
                            "yum": {
                                "httpd": [],
                                "php": []
                            }
                        },
                        "files": {
                            "/tmp/cwlogs/apacheaccess.conf": {
                                "content": {
```

```
                                    "Fn::Join": [
                                        "",
                                        [
                                            "[general]\n",
                                            "state_file= /var/awslogs/agent-state\n",
                                            "[/var/log/httpd/access_log]\n",
                                            "file = /var/log/httpd/access_log\n",
                                            "log_group_name = ",
                                            {
                                                "Ref": "WebServerLogGroup"
                                            },
                                            "\n",
                                            "log_stream_name = {instance_id}/apache.log
\n",
                                            "datetime_format = %d/%b/%Y:%H:%M:%S"
                                        ]
                                    ]
                                },
                                "mode": "000400",
                                "owner": "apache",
                                "group": "apache"
                            },
                            "/var/www/html/index.php": {
                                "content": {
                                    "Fn::Join": [
                                        "",
                                        [
                                            "<?php\n",
                                            "echo '<h1>AWS CloudFormation sample PHP
 application</h1>';\n",
                                            "?>\n"
                                        ]
                                    ]
                                },
                                "mode": "000644",
                                "owner": "apache",
                                "group": "apache"
                            },
                            "/etc/cfn/cfn-hup.conf": {
                                "content": {
                                    "Fn::Join": [
                                        "",
                                        [
                                            "[main]\n",
```

```
                                                "stack=",
                                                {
                                                    "Ref": "AWS::StackId"
                                                },
                                                "\n",
                                                "region=",
                                                {
                                                    "Ref": "AWS::Region"
                                                },
                                                "\n"
                                            ]
                                        ]
                                    },
                                    "mode": "000400",
                                    "owner": "root",
                                    "group": "root"
                                },
                                "/etc/cfn/hooks.d/cfn-auto-reloader.conf": {
                                    "content": {
                                        "Fn::Join": [
                                            "",
                                            [
                                                "[cfn-auto-reloader-hook]\n",
                                                "triggers=post.update\n",

"path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init\n",
                                                "action=/opt/aws/bin/cfn-init -s ",
                                                {
                                                    "Ref": "AWS::StackId"
                                                },
                                                " -r WebServerHost ",
                                                " --region     ",
                                                {
                                                    "Ref": "AWS::Region"
                                                },
                                                "\n",
                                                "runas=root\n"
                                            ]
                                        ]
                                    }
                                }
                            },
                            "services": {
                                "sysvinit": {
```

```
                                "httpd": {
                                    "enabled": "true",
                                    "ensureRunning": "true"
                                },
                                "sendmail": {
                                    "enabled": "false",
                                    "ensureRunning": "false"
                                }
                            }
                        }
                    }
                }
            },
            "CreationPolicy": {
                "ResourceSignal": {
                    "Timeout": "PT5M"
                }
            },
            "Properties": {
                "ImageId": {
                    "Fn::FindInMap": [
                        "RegionMap",
                        {
                            "Ref": "AWS::Region"
                        },
                        "AMI"
                    ]
                },
                "KeyName": {
                    "Ref": "KeyName"
                },
                "InstanceType": "t1.micro",
                "SecurityGroups": [
                    {
                        "Ref": "WebServerSecurityGroup"
                    }
                ],
                "IamInstanceProfile": {
                    "Ref": "LogRoleInstanceProfile"
                },
                "UserData": {
                    "Fn::Base64": {
                        "Fn::Join": [
                            "",
```

```
                                [
                                    "#!/bin/bash -xe\n",
                                    "# Get the latest CloudFormation package\n",
                                    "yum install -y aws-cfn-bootstrap\n",
                                    "# Start cfn-init\n",
                                    "/opt/aws/bin/cfn-init -s ",
                                    {
                                        "Ref": "AWS::StackId"
                                    },
                                    " -r WebServerHost ",
                                    " --region ",
                                    {
                                        "Ref": "AWS::Region"
                                    },
                                    " || error_exit 'Failed to run cfn-init'\n",
                                    "# Start up the cfn-hup daemon to listen for changes to
 the EC2 instance metadata\n",
                                    "/opt/aws/bin/cfn-hup || error_exit 'Failed to start
 cfn-hup'\n",
                                    "# Get the CloudWatch Logs agent\n",
                                    "wget https://s3.amazonaws.com/aws-cloudwatch/
downloads/latest/awslogs-agent-setup.py\n",
                                    "# Install the CloudWatch Logs agent\n",
                                    "python awslogs-agent-setup.py -n -r ",
                                    {
                                        "Ref": "AWS::Region"
                                    },
                                    " -c /tmp/cwlogs/apacheaccess.conf || error_exit
 'Failed to run CloudWatch Logs agent setup'\n",
                                    "# All done so signal success\n",
                                    "/opt/aws/bin/cfn-signal -e $? ",
                                    "         --stack ",
                                    {
                                        "Ref": "AWS::StackName"
                                    },
                                    "         --resource WebServerHost ",
                                    "         --region ",
                                    {
                                        "Ref": "AWS::Region"
                                    },
                                    "\n"
                                ]
                            ]
                        }
```

```
                }
            }
        },
        "WebServerLogGroup": {
            "Type": "AWS::Logs::LogGroup",
            "Properties": {
                "RetentionInDays": 7
            }
        },
        "404MetricFilter": {
            "Type": "AWS::Logs::MetricFilter",
            "Properties": {
                "LogGroupName": {
                    "Ref": "WebServerLogGroup"
                },
                "FilterPattern": "[ip, identity, user_id, timestamp, request,
 status_code = 404, size, ...]",
                "MetricTransformations": [
                    {
                        "MetricValue": "1",
                        "MetricNamespace": "test/404s",
                        "MetricName": "test404Count"
                    }
                ]
            }
        },
        "BytesTransferredMetricFilter": {
            "Type": "AWS::Logs::MetricFilter",
            "Properties": {
                "LogGroupName": {
                    "Ref": "WebServerLogGroup"
                },
                "FilterPattern": "[ip, identity, user_id, timestamp, request,
 status_code, size, ...]",
                "MetricTransformations": [
                    {
                        "MetricValue": "$size",
                        "MetricNamespace": "test/BytesTransferred",
                        "MetricName": "testBytesTransferred"
                    }
                ]
            }
        },
        "404Alarm": {
```

```
            "Type": "AWS::CloudWatch::Alarm",
            "Properties": {
                "AlarmDescription": "The number of 404s is greater than 2 over 2
 minutes",
                "MetricName": "test404Count",
                "Namespace": "test/404s",
                "Statistic": "Sum",
                "Period": "60",
                "EvaluationPeriods": "2",
                "Threshold": "2",
                "AlarmActions": [
                    {
                        "Ref": "AlarmNotificationTopic"
                    }
                ],
                "ComparisonOperator": "GreaterThanThreshold"
            }
        },
        "BandwidthAlarm": {
            "Type": "AWS::CloudWatch::Alarm",
            "Properties": {
                "AlarmDescription": "The average volume of traffic is greater 3500 KB
 over 10 minutes",
                "MetricName": "testBytesTransferred",
                "Namespace": "test/BytesTransferred",
                "Statistic": "Average",
                "Period": "300",
                "EvaluationPeriods": "2",
                "Threshold": "3500",
                "AlarmActions": [
                    {
                        "Ref": "AlarmNotificationTopic"
                    }
                ],
                "ComparisonOperator": "GreaterThanThreshold"
            }
        },
        "AlarmNotificationTopic": {
            "Type": "AWS::SNS::Topic",
            "Properties": {
                "Subscription": [
                    {
                        "Endpoint": {
                            "Ref": "OperatorEmail"
```

```
                        },
                        "Protocol": "email"
                    }
                ]
            }
        }
    },
    "Outputs": {
        "InstanceId": {
            "Description": "The instance ID of the web server",
            "Value": {
                "Ref": "WebServerHost"
            }
        },
        "WebsiteURL": {
            "Value": {
                "Fn::Join": [
                    "",
                    [
                        "http://",
                        {
                            "Fn::GetAtt": [
                                "WebServerHost",
                                "PublicDnsName"
                            ]
                        }
                    ]
                ]
            },
            "Description": "URL for newly created LAMP stack"
        },
        "PublicIP": {
            "Description": "Public IP address of the web server",
            "Value": {
                "Fn::GetAtt": [
                    "WebServerHost",
                    "PublicIp"
                ]
            }
        },
        "CloudWatchLogGroupName": {
            "Description": "The name of the CloudWatch log group",
            "Value": {
                "Ref": "WebServerLogGroup"
```

```
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: AWS CloudFormation Sample Template for CloudWatch Logs.
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: 'AWS::EC2::KeyPair::KeyName'
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  OperatorEmail:
    Description: Email address to notify if there are any scaling operations
    Type: String
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-0ff8a91507f77f867
    us-west-1:
      AMI: ami-0bdb828fd58c52235
    us-west-2:
      AMI: ami-a0cfeed8
    eu-west-1:
      AMI: ami-047bb4163c506cd98
    ap-southeast-1:
      AMI: ami-08569b978cc4dfa10
    ap-southeast-2:
      AMI: ami-09b42976632b27e9b
    ap-northeast-1:
      AMI: ami-06cd52961ce9f0d85
    sa-east-1:
      AMI: ami-07b14488da8ea02a0
```

```
    eu-central-1:
      AMI: ami-0233214e13e500f77
Resources:
  LogRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      Path: /
      Policies:
        - PolicyName: LogRolePolicy
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action:
                  - 'logs:Create*'
                  - 'logs:PutLogEvents'
                  - 's3:GetObject'
                Resource:
                  - 'arn:aws:logs:*:*:*'
                  - 'arn:aws:s3:::*'
  LogRoleInstanceProfile:
    Type: 'AWS::IAM::InstanceProfile'
    Properties:
      Path: /
      Roles:
        - !Ref LogRole
  WebServerSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
```

```
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref SSHLocation
  WebServerHost:
    Type: 'AWS::EC2::Instance'
    Metadata:
      Comment: Install a simple PHP application
      'AWS::CloudFormation::Init':
        config:
          packages:
            yum:
              httpd: []
              php: []
          files:
            /tmp/cwlogs/apacheaccess.conf:
              content: !Join
                - ''
                - - |
                    [general]
                  - |
                    state_file= /var/awslogs/agent-state
                  - |
                    [/var/log/httpd/access_log]
                  - |
                    file = /var/log/httpd/access_log
                  - 'log_group_name = '
                  - !Ref WebServerLogGroup
                  - |+

                  - |
                    log_stream_name = {instance_id}/apache.log
                  - 'datetime_format = %d/%b/%Y:%H:%M:%S'
              mode: '000400'
              owner: apache
              group: apache
            /var/www/html/index.php:
              content: !Join
                - ''
                - - |
                    <?php
                  - |
                    echo '<h1>AWS CloudFormation sample PHP application</h1>';
                  - |
```

```
                ?>
            mode: '000644'
            owner: apache
            group: apache
          /etc/cfn/cfn-hup.conf:
            content: !Join
              - ''
              - - |
                  [main]
                - stack=
                - !Ref 'AWS::StackId'
                - |+

                - region=
                - !Ref 'AWS::Region'
                - |+

            mode: '000400'
            owner: root
            group: root
          /etc/cfn/hooks.d/cfn-auto-reloader.conf:
            content: !Join
              - ''
              - - |
                  [cfn-auto-reloader-hook]
                - |
                  triggers=post.update
                - >
                  path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init
                - 'action=/opt/aws/bin/cfn-init -s '
                - !Ref 'AWS::StackId'
                - ' -r WebServerHost '
                - ' --region     '
                - !Ref 'AWS::Region'
                - |+

                - |
                  runas=root
        services:
          sysvinit:
            httpd:
              enabled: 'true'
              ensureRunning: 'true'
            sendmail:
```

```
                enabled: 'false'
                ensureRunning: 'false'
    CreationPolicy:
      ResourceSignal:
        Timeout: PT5M
    Properties:
      ImageId: !FindInMap
        - RegionMap
        - !Ref 'AWS::Region'
        - AMI
      KeyName: !Ref KeyName
      InstanceType: t1.micro
      SecurityGroups:
        - !Ref WebServerSecurityGroup
      IamInstanceProfile: !Ref LogRoleInstanceProfile
      UserData: !Base64
        'Fn::Join':
          - ''
          - - |
              #!/bin/bash -xe
            - |
              # Get the latest CloudFormation package
            - |
              yum install -y aws-cfn-bootstrap
            - |
              # Start cfn-init
            - '/opt/aws/bin/cfn-init -s '
            - !Ref 'AWS::StackId'
            - ' -r WebServerHost '
            - ' --region '
            - !Ref 'AWS::Region'
            - |2
               || error_exit 'Failed to run cfn-init'
            - >
              # Start up the cfn-hup daemon to listen for changes to the EC2
              instance metadata
            - |
              /opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'
            - |
              # Get the CloudWatch Logs agent
            - >
              wget
              https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py
```

```
            - |
              # Install the CloudWatch Logs agent
            - 'python awslogs-agent-setup.py -n -r '
            - !Ref 'AWS::Region'
            - |2
               -c /tmp/cwlogs/apacheaccess.conf || error_exit 'Failed to run CloudWatch
 Logs agent setup'
            - |
              # All done so signal success
            - '/opt/aws/bin/cfn-signal -e $? '
            - '         --stack '
            - !Ref 'AWS::StackName'
            - '         --resource WebServerHost '
            - '         --region '
            - !Ref 'AWS::Region'
            - |+

  WebServerLogGroup:
    Type: 'AWS::Logs::LogGroup'
    Properties:
      RetentionInDays: 7
  404MetricFilter:
    Type: 'AWS::Logs::MetricFilter'
    Properties:
      LogGroupName: !Ref WebServerLogGroup
      FilterPattern: >-
        [ip, identity, user_id, timestamp, request, status_code = 404, size,
        ...]
      MetricTransformations:
        - MetricValue: '1'
          MetricNamespace: test/404s
          MetricName: test404Count
  BytesTransferredMetricFilter:
    Type: 'AWS::Logs::MetricFilter'
    Properties:
      LogGroupName: !Ref WebServerLogGroup
      FilterPattern: '[ip, identity, user_id, timestamp, request, status_code,
 size, ...]'
      MetricTransformations:
        - MetricValue: $size
          MetricNamespace: test/BytesTransferred
          MetricName: testBytesTransferred
  404Alarm:
    Type: 'AWS::CloudWatch::Alarm'
```

```
    Properties:
      AlarmDescription: The number of 404s is greater than 2 over 2 minutes
      MetricName: test404Count
      Namespace: test/404s
      Statistic: Sum
      Period: '60'
      EvaluationPeriods: '2'
      Threshold: '2'
      AlarmActions:
        - !Ref AlarmNotificationTopic
      ComparisonOperator: GreaterThanThreshold
  BandwidthAlarm:
    Type: 'AWS::CloudWatch::Alarm'
    Properties:
      AlarmDescription: The average volume of traffic is greater 3500 KB over 10
 minutes
      MetricName: testBytesTransferred
      Namespace: test/BytesTransferred
      Statistic: Average
      Period: '300'
      EvaluationPeriods: '2'
      Threshold: '3500'
      AlarmActions:
        - !Ref AlarmNotificationTopic
      ComparisonOperator: GreaterThanThreshold
  AlarmNotificationTopic:
    Type: 'AWS::SNS::Topic'
    Properties:
      Subscription:
        - Endpoint: !Ref OperatorEmail
          Protocol: email
Outputs:
  InstanceId:
    Description: The instance ID of the web server
    Value: !Ref WebServerHost
  WebsiteURL:
    Value: !Join
      - ''
      - - 'http://'
        - !GetAtt
          - WebServerHost
          - PublicDnsName
    Description: URL for newly created LAMP stack
  PublicIP:
```

```
    Description: Public IP address of the web server
    Value: !GetAtt
      - WebServerHost
      - PublicIp
  CloudWatchLogGroupName:
    Description: The name of the CloudWatch log group
    Value: !Ref WebServerLogGroup
```

Sendet CloudWatch Protokolle von einer Windows-Instanz an Logs

Die folgende Vorlage konfiguriert CloudWatch Logs für eine Windows 2012R2-Instanz.

Der CloudWatch Logs-Agent unter Windows (SSM-Agent unter Windows 2012R2 und Windows 2016 AMIs) sendet Protokolle erst, nachdem er gestartet wurde, sodass alle Protokolle, die vor dem Start generiert wurden, nicht gesendet werden. Um dieses Problem zu umgehen, hilft die Vorlage folgendermaßen dabei, sicherzustellen, dass der Agent gestartet wird, bevor Protokolle geschrieben werden:

- Konfigurieren der Agent-Einrichtung als erstes `config`-Element in cfn-init `configSets`.
- Einfügen einer Pause mit `waitAfterCompletion` nach dem Befehl zum Starten des Agents.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Sample template that sets up and configures CloudWatch logs on
 Windows 2012R2 instance instance.",
    "Parameters": {
        "KeyPair": {
            "Description": "Name of an existing EC2 KeyPair to enable RDP access to the
 instances",
            "Type": "AWS::EC2::KeyPair::KeyName",
            "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
        },
        "RDPLocation": {
            "Description": "The IP address range that can be used to RDP to the EC2
 instances",
            "Type": "String",
            "MinLength": "9",
            "MaxLength": "18",
            "Default": "0.0.0.0/0",
```

```
            "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\
\d{1,2})",
            "ConstraintDescription": "must be a valid IP CIDR range of the form
 x.x.x.x/x."
        },
        "OperatorEmail": {
            "Description": "Email address to notify if there are any scaling
 operations",
            "Type": "String"
        }
    },
    "Mappings": {
        "AWSAMIRegionMap": {
            "ap-northeast-1": {
                "WS2012R2": "ami-09e7006451ad8bf4d"
            },
            "ap-northeast-2": {
                "WS2012R2": "ami-0754980e4d02153f9"
            },
            "ap-south-1": {
                "WS2012R2": "ami-00ad91b37d56c1d08"
            },
            "ap-southeast-1": {
                "WS2012R2": "ami-09e7006451ad8bf4d"
            },
            "ap-southeast-2": {
                "WS2012R2": "ami-000d23d3067008aea"
            },
            "ca-central-1": {
                "WS2012R2": "ami-0d8e70862465b9da0"
            },
            "eu-central-1": {
                "WS2012R2": "ami-0c0f322f5676ba254"
            },
            "eu-west-1": {
                "WS2012R2": "ami-0a46adf18f8875ad6"
            },
            "eu-west-2": {
                "WS2012R2": "ami-0651428174d9438e9"
            },
            "sa-east-1": {
                "WS2012R2": "ami-08ebd138109a6c223"
            },
            "us-east-1": {
```

```
                "WS2012R2": "ami-0ef6fb504535468b2"
            },
            "us-east-2": {
                "WS2012R2": "ami-0f466c6044f510bd3"
            },
            "us-west-1": {
                "WS2012R2": "ami-026f68ef6465e6c09"
            },
            "us-west-2": {
                "WS2012R2": "ami-0274ca53943a86543"
            }
        }
    },
    "Resources": {
        "WebServerSecurityGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "Enable HTTP access via port 80 and RDP access via
 port 3389",
                "SecurityGroupIngress": [
                    {
                        "IpProtocol": "tcp",
                        "FromPort": "80",
                        "ToPort": "80",
                        "CidrIp": "0.0.0.0/0"
                    },
                    {
                        "IpProtocol": "tcp",
                        "FromPort": "3389",
                        "ToPort": "3389",
                        "CidrIp": {
                            "Ref": "RDPLocation"
                        }
                    }
                ]
            }
        },
        "LogRole": {
            "Type": "AWS::IAM::Role",
            "Properties": {
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
```

```
                            "Effect": "Allow",
                            "Principal": {
                                "Service": [
                                    "ec2.amazonaws.com"
                                ]
                            },
                            "Action": [
                                "sts:AssumeRole"
                            ]
                        }
                    ]
                },
                "ManagedPolicyArns": [
                    "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
                ],
                "Path": "/",
                "Policies": [
                    {
                        "PolicyName": "LogRolePolicy",
                        "PolicyDocument": {
                            "Version": "2012-10-17",
                            "Statement": [
                                {
                                    "Effect": "Allow",
                                    "Action": [
                                        "logs:Create*",
                                        "logs:PutLogEvents",
                                        "s3:GetObject"
                                    ],
                                    "Resource": [
                                        "arn:aws:logs:*:*:*",
                                        "arn:aws:s3:::*"
                                    ]
                                }
                            ]
                        }
                    }
                ]
            }
        },
        "LogRoleInstanceProfile": {
            "Type": "AWS::IAM::InstanceProfile",
            "Properties": {
                "Path": "/",
```

```
                "Roles": [
                    {
                        "Ref": "LogRole"
                    }
                ]
            }
        },
        "WebServerHost": {
            "Type": "AWS::EC2::Instance",
            "CreationPolicy": {
                "ResourceSignal": {
                    "Timeout": "PT15M"
                }
            },
            "Metadata": {
                "AWS::CloudFormation::Init": {
                    "configSets": {
                        "config": [
                            "00-ConfigureCWLogs",
                            "01-InstallWebServer",
                            "02-ConfigureApplication",
                            "03-Finalize"
                        ]
                    },
                    "00-ConfigureCWLogs": {
                        "files": {
                            "C:\\Program Files\\Amazon\\SSM\\Plugins\\awsCloudWatch\
\AWS.EC2.Windows.CloudWatch.json": {
                                "content": {
                                    "Fn::Sub": "{\n  \"EngineConfiguration
\": {\n      \"Components\": [\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch
\",\n              \"Id\": \"ApplicationEventLog\",\n              \"Parameters\":
 {\n                  \"Levels\": \"7\",\n                  \"LogName\": \"Application
\"\n              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch
\",\n              \"Id\": \"SystemEventLog\",\n              \"Parameters\":
 {\n                  \"Levels\": \"7\",\n                  \"LogName\": \"System
\"\n              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch
\",\n              \"Id\": \"SecurityEventLog\",\n              \"Parameters\":
 {\n                  \"Levels\": \"7\",\n                  \"LogName\": \"Security
\"\n              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch
```

```
\",\n              \"Id\": \"EC2ConfigLog\",\n              \"Parameters
\": {\n                  \"CultureName\": \"en-US\",\n
 \"Encoding\": \"ASCII\",\n                  \"Filter\": \"EC2ConfigLog.txt
\",\n                  \"LogDirectoryPath\": \"C:\\\\Program Files\\\\Amazon
\\\\Ec2ConfigService\\\\Logs\",\n                  \"TimeZoneKind\": \"UTC
\",\n                  \"TimestampFormat\": \"yyyy-MM-ddTHH:mm:ss.fffZ:\"\n
              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch
\",\n              \"Id\": \"CfnInitLog\",\n              \"Parameters\": {\n
                  \"CultureName\": \"en-US\",\n                  \"Encoding\":
 \"ASCII\",\n                  \"Filter\": \"cfn-init.log\",\n
  \"LogDirectoryPath\": \"C:\\\\cfn\\\\log\",\n                  \"TimeZoneKind
\": \"Local\",\n                  \"TimestampFormat\": \"yyyy-MM-dd HH:mm:ss,fff
\"\n              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch
\",\n              \"Id\": \"IISLogs\",\n              \"Parameters\": {\n
       \"CultureName\": \"en-US\",\n                  \"Encoding\": \"UTF-8\",\n
            \"Filter\": \"\",\n                  \"LineCount\": \"3\",\n
    \"LogDirectoryPath\": \"C:\\\\inetpub\\\\logs\\\\LogFiles\\\\W3SVC1\",\n
         \"TimeZoneKind\": \"UTC\",\n                  \"TimestampFormat\": \"yyyy-MM-
dd HH:mm:ss\"\n              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.
\",\n              \"Id\": \"MemoryPerformanceCounter\",\n              \"Parameters\":
 {\n                  \"CategoryName\": \"Memory\",\n                  \"CounterName
\": \"Available MBytes\",\n                  \"DimensionName\": \"\",\n
           \"DimensionValue\": \"\",\n                  \"InstanceName\": \"\",
\n                  \"MetricName\": \"Memory\",\n                  \"Unit\":
 \"Megabytes\"\n              }\n          },\n          {\n              \"FullName
\": \"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch\",\n
              \"Id\": \"CloudWatchApplicationEventLog\",\n              \"Parameters
\": {\n                  \"AccessKey\": \"\",\n                  \"LogGroup\":
 \"${LogGroup}\",\n                  \"LogStream\": \"{instance_id}/ApplicationEventLog
\",\n                  \"Region\": \"${AWS::Region}\",\n                  \"SecretKey
\": \"\"\n              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch\",
\n              \"Id\": \"CloudWatchSystemEventLog\",\n              \"Parameters
\": {\n                  \"AccessKey\": \"\",\n                  \"LogGroup\":
 \"${LogGroup}\",\n                  \"LogStream\": \"{instance_id}/SystemEventLog
\",\n                  \"Region\": \"${AWS::Region}\",\n                  \"SecretKey
\": \"\"\n              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch\",\n
              \"Id\": \"CloudWatchSecurityEventLog\",\n              \"Parameters
\": {\n                  \"AccessKey\": \"\",\n                  \"LogGroup\":
 \"${LogGroup}\",\n                  \"LogStream\": \"{instance_id}/SecurityEventLog
```

```
\",\n                  \"Region\": \"${AWS::Region}\",\n                  \"SecretKey
\": \"\"\n              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch\",
\n              \"Id\": \"CloudWatchEC2ConfigLog\",\n              \"Parameters
\": {\n                  \"AccessKey\": \"\",\n                  \"LogGroup\":
 \"${LogGroup}\",\n                  \"LogStream\": \"{instance_id}/EC2ConfigLog\",
\n                  \"Region\": \"${AWS::Region}\",\n                  \"SecretKey
\": \"\"\n              }\n          },\n          {\n              \"FullName
\": \"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch
\",\n              \"Id\": \"CloudWatchCfnInitLog\",\n              \"Parameters
\": {\n                  \"AccessKey\": \"\",\n                  \"LogGroup\":
 \"${LogGroup}\",\n                  \"LogStream\": \"{instance_id}/CfnInitLog\",
\n                  \"Region\": \"${AWS::Region}\",\n                  \"SecretKey
\": \"\"\n              }\n          },\n          {\n              \"FullName
\": \"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch
\",\n              \"Id\": \"CloudWatchIISLogs\",\n              \"Parameters
\": {\n                  \"AccessKey\": \"\",\n                  \"LogGroup\":
 \"${LogGroup}\",\n                  \"LogStream\": \"{instance_id}/IISLogs\",\n
                  \"Region\": \"${AWS::Region}\",\n                  \"SecretKey
\": \"\"\n              }\n          },\n          {\n              \"FullName\":
 \"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch
\",\n              \"Id\": \"CloudWatch\",\n              \"Parameters\": {\n
               \"AccessKey\": \"\",\n                  \"NameSpace\": \"Windows/
Default\",\n                  \"Region\": \"${AWS::Region}\",\n
 \"SecretKey\": \"\"\n              }\n          }\n      ],\n      \"Flows\": {\n
       \"Flows\": [\n              \"ApplicationEventLog,CloudWatchApplicationEventLog
\",\n              \"SystemEventLog,CloudWatchSystemEventLog\",\n
             \"SecurityEventLog,CloudWatchSecurityEventLog\",\n
           \"EC2ConfigLog,CloudWatchEC2ConfigLog\",\n
 \"CfnInitLog,CloudWatchCfnInitLog\",\n              \"IISLogs,CloudWatchIISLogs\",\n
              \"MemoryPerformanceCounter,CloudWatch\"\n          ]\n      },\n
 \"PollInterval\": \"00:00:05\"\n  },\n  \"IsEnabled\": true\n}\n"
                                }
                            }
                        },
                        "commands": {
                            "0-enableSSM": {
                                "command": "powershell.exe -Command \"Set-Service -Name
 AmazonSSMAgent -StartupType Automatic\" ",
                                "waitAfterCompletion": "0"
                            },
                            "1-restartSSM": {
                                "command": "powershell.exe -Command \"Restart-Service
 AmazonSSMAgent \"",
```

```
                                "waitAfterCompletion": "30"
                            }
                        }
                    },
                    "01-InstallWebServer": {
                        "commands": {
                            "01_install_webserver": {
                                "command": "powershell.exe -Command \"Install-
WindowsFeature Web-Server  -IncludeAllSubFeature\"",
                                "waitAfterCompletion": "0"
                            }
                        }
                    },
                    "02-ConfigureApplication": {
                        "files": {
                            "c:\\Inetpub\\wwwroot\\index.htm": {
                                "content": "<html> <head> <title>Test Application
 Page</title> </head> <body> <h1>Congratulations !! Your IIS server is configured.</h1>
 </body> </html>"
                            }
                        }
                    },
                    "03-Finalize": {
                        "commands": {
                            "00_signal_success": {
                                "command": {
                                    "Fn::Sub": "cfn-signal.exe -e 0 --resource
 WebServerHost --stack ${AWS::StackName} --region ${AWS::Region}"
                                },
                                "waitAfterCompletion": "0"
                            }
                        }
                    }
                }
            },
            "Properties": {
                "KeyName": {
                    "Ref": "KeyPair"
                },
                "ImageId": {
                    "Fn::FindInMap": [
                        "AWSAMIRegionMap",
                        {
                            "Ref": "AWS::Region"
```

```
                        },
                        "WS2012R2"
                    ]
                },
                "InstanceType": "t2.xlarge",
                "SecurityGroupIds": [
                    {
                        "Ref": "WebServerSecurityGroup"
                    }
                ],
                "IamInstanceProfile": {
                    "Ref": "LogRoleInstanceProfile"
                },
                "UserData": {
                    "Fn::Base64": {
                        "Fn::Sub": "<script>\nwmic product where \"description='Amazon
 SSM Agent' \" uninstall\nwmic product where \"description='aws-cfn-bootstrap'
 \" uninstall \nstart /wait c:\\\\Windows\\\\system32\\\\msiexec /passive /qn /i
 https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-latest.msi
\npowershell.exe -Command \"iwr https://s3.amazonaws.com/ec2-downloads-windows/
SSMAgent/latest/windows_amd64/AmazonSSMAgentSetup.exe  -UseBasicParsing -OutFile C:
\\\\AmazonSSMAgentSetup.exe\"\nstart /wait C:\\\\AmazonSSMAgentSetup.exe /install /
quiet\ncfn-init.exe -v -c config -s ${AWS::StackName} --resource WebServerHost --region
 ${AWS::Region} \n</script>\n"
                    }
                }
            }
        },
        "LogGroup": {
            "Type": "AWS::Logs::LogGroup",
            "Properties": {
                "RetentionInDays": 7
            }
        },
        "404MetricFilter": {
            "Type": "AWS::Logs::MetricFilter",
            "Properties": {
                "LogGroupName": {
                    "Ref": "LogGroup"
                },
                "FilterPattern": "[timestamps, serverip, method, uri, query, port,
 dash, clientip, useragent, status_code = 404, ...]",
                "MetricTransformations": [
                    {
```

```
                        "MetricValue": "1",
                        "MetricNamespace": "test/404s",
                        "MetricName": "test404Count"
                    }
                ]
            }
        },
        "404Alarm": {
            "Type": "AWS::CloudWatch::Alarm",
            "Properties": {
                "AlarmDescription": "The number of 404s is greater than 2 over 2
 minutes",
                "MetricName": "test404Count",
                "Namespace": "test/404s",
                "Statistic": "Sum",
                "Period": "60",
                "EvaluationPeriods": "2",
                "Threshold": "2",
                "AlarmActions": [
                    {
                        "Ref": "AlarmNotificationTopic"
                    }
                ],
                "ComparisonOperator": "GreaterThanThreshold"
            }
        },
        "AlarmNotificationTopic": {
            "Type": "AWS::SNS::Topic",
            "Properties": {
                "Subscription": [
                    {
                        "Endpoint": {
                            "Ref": "OperatorEmail"
                        },
                        "Protocol": "email"
                    }
                ]
            }
        }
    },
    "Outputs": {
        "InstanceId": {
            "Description": "The instance ID of the web server",
            "Value": {
```

```
                "Ref": "WebServerHost"
            }
        },
        "WebsiteURL": {
            "Value": {
                "Fn::Sub": "http://${WebServerHost.PublicDnsName}"
            },
            "Description": "URL for newly created IIS web server"
        },
        "PublicIP": {
            "Description": "Public IP address of the web server",
            "Value": {
                "Fn::GetAtt": [
                    "WebServerHost",
                    "PublicIp"
                ]
            }
        },
        "CloudWatchLogGroupName": {
            "Description": "The name of the CloudWatch log group",
            "Value": {
                "Ref": "LogGroup"
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: >-
  Sample template that sets up and configures CloudWatch logs on Windows 2012R2
  instance instance.
Parameters:
  KeyPair:
    Description: Name of an existing EC2 KeyPair to enable RDP access to the instances
    Type: 'AWS::EC2::KeyPair::KeyName'
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  RDPLocation:
    Description: The IP address range that can be used to RDP to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
```

```
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  OperatorEmail:
    Description: Email address to notify if there are any scaling operations
    Type: String
Mappings:
  AWSAMIRegionMap:
    ap-northeast-1:
      WS2012R2: ami-09e7006451ad8bf4d
    ap-northeast-2:
      WS2012R2: ami-0754980e4d02153f9
    ap-south-1:
      WS2012R2: ami-00ad91b37d56c1d08
    ap-southeast-1:
      WS2012R2: ami-09e7006451ad8bf4d
    ap-southeast-2:
      WS2012R2: ami-000d23d3067008aea
    ca-central-1:
      WS2012R2: ami-0d8e70862465b9da0
    eu-central-1:
      WS2012R2: ami-0c0f322f5676ba254
    eu-west-1:
      WS2012R2: ami-0a46adf18f8875ad6
    eu-west-2:
      WS2012R2: ami-0651428174d9438e9
    sa-east-1:
      WS2012R2: ami-08ebd138109a6c223
    us-east-1:
      WS2012R2: ami-0ef6fb504535468b2
    us-east-2:
      WS2012R2: ami-0f466c6044f510bd3
    us-west-1:
      WS2012R2: ami-026f68ef6465e6c09
    us-west-2:
      WS2012R2: ami-0274ca53943a86543
Resources:
  WebServerSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Enable HTTP access via port 80 and RDP access via port 3389
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '80'
```

```
          ToPort: '80'
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: '3389'
          ToPort: '3389'
          CidrIp: !Ref RDPLocation
  LogRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore'
      Path: /
      Policies:
        - PolicyName: LogRolePolicy
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action:
                  - 'logs:Create*'
                  - 'logs:PutLogEvents'
                  - 's3:GetObject'
                Resource:
                  - 'arn:aws:logs:*:*:*'
                  - 'arn:aws:s3:::*'
  LogRoleInstanceProfile:
    Type: 'AWS::IAM::InstanceProfile'
    Properties:
      Path: /
      Roles:
        - !Ref LogRole
  WebServerHost:
    Type: 'AWS::EC2::Instance'
    CreationPolicy:
      ResourceSignal:
```

```
        Timeout: PT15M
    Metadata:
      'AWS::CloudFormation::Init':
        configSets:
          config:
            - 00-ConfigureCWLogs
            - 01-InstallWebServer
            - 02-ConfigureApplication
            - 03-Finalize
        00-ConfigureCWLogs:
          files:
            'C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch
\AWS.EC2.Windows.CloudWatch.json':
              content: !Sub |
                {
                  "EngineConfiguration": {
                      "Components": [
                          {
                              "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                              "Id": "ApplicationEventLog",
                              "Parameters": {
                                  "Levels": "7",
                                  "LogName": "Application"
                              }
                          },
                          {
                              "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                              "Id": "SystemEventLog",
                              "Parameters": {
                                  "Levels": "7",
                                  "LogName": "System"
                              }
                          },
                          {
                              "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                              "Id": "SecurityEventLog",
                              "Parameters": {
                                  "Levels": "7",
                                  "LogName": "Security"
                              }
                          },
```

```
                            {
                                "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
                                "Id": "EC2ConfigLog",
                                "Parameters": {
                                    "CultureName": "en-US",
                                    "Encoding": "ASCII",
                                    "Filter": "EC2ConfigLog.txt",
                                    "LogDirectoryPath": "C:\\Program Files\\Amazon\
\Ec2ConfigService\\Logs",
                                    "TimeZoneKind": "UTC",
                                    "TimestampFormat": "yyyy-MM-ddTHH:mm:ss.fffZ:"
                                }
                            },
                            {
                                "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
                                "Id": "CfnInitLog",
                                "Parameters": {
                                    "CultureName": "en-US",
                                    "Encoding": "ASCII",
                                    "Filter": "cfn-init.log",
                                    "LogDirectoryPath": "C:\\cfn\\log",
                                    "TimeZoneKind": "Local",
                                    "TimestampFormat": "yyyy-MM-dd HH:mm:ss,fff"
                                }
                            },
                            {
                                "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
                                "Id": "IISLogs",
                                "Parameters": {
                                    "CultureName": "en-US",
                                    "Encoding": "UTF-8",
                                    "Filter": "",
                                    "LineCount": "3",
                                    "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\
\W3SVC1",
                                    "TimeZoneKind": "UTC",
                                    "TimestampFormat": "yyyy-MM-dd HH:mm:ss"
                                }
                            },
                            {
```

```
                                "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E
                                "Id": "MemoryPerformanceCounter",
                                "Parameters": {
                                    "CategoryName": "Memory",
                                    "CounterName": "Available MBytes",
                                    "DimensionName": "",
                                    "DimensionValue": "",
                                    "InstanceName": "",
                                    "MetricName": "Memory",
                                    "Unit": "Megabytes"
                                }
                            },
                            {
                                "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
                                "Id": "CloudWatchApplicationEventLog",
                                "Parameters": {
                                    "AccessKey": "",
                                    "LogGroup": "${LogGroup}",
                                    "LogStream": "{instance_id}/ApplicationEventLog",
                                    "Region": "${AWS::Region}",
                                    "SecretKey": ""
                                }
                            },
                            {
                                "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
                                "Id": "CloudWatchSystemEventLog",
                                "Parameters": {
                                    "AccessKey": "",
                                    "LogGroup": "${LogGroup}",
                                    "LogStream": "{instance_id}/SystemEventLog",
                                    "Region": "${AWS::Region}",
                                    "SecretKey": ""
                                }
                            },
                            {
                                "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
                                "Id": "CloudWatchSecurityEventLog",
                                "Parameters": {
                                    "AccessKey": "",
                                    "LogGroup": "${LogGroup}",
```

```
                                "LogStream": "{instance_id}/SecurityEventLog",
                                "Region": "${AWS::Region}",
                                "SecretKey": ""
                            }
                        },
                        {
                            "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
                            "Id": "CloudWatchEC2ConfigLog",
                            "Parameters": {
                                "AccessKey": "",
                                "LogGroup": "${LogGroup}",
                                "LogStream": "{instance_id}/EC2ConfigLog",
                                "Region": "${AWS::Region}",
                                "SecretKey": ""
                            }
                        },
                        {
                            "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
                            "Id": "CloudWatchCfnInitLog",
                            "Parameters": {
                                "AccessKey": "",
                                "LogGroup": "${LogGroup}",
                                "LogStream": "{instance_id}/CfnInitLog",
                                "Region": "${AWS::Region}",
                                "SecretKey": ""
                            }
                        },
                        {
                            "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
                            "Id": "CloudWatchIISLogs",
                            "Parameters": {
                                "AccessKey": "",
                                "LogGroup": "${LogGroup}",
                                "LogStream": "{instance_id}/IISLogs",
                                "Region": "${AWS::Region}",
                                "SecretKey": ""
                            }
                        },
                        {
                            "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
```

```
                            "Id": "CloudWatch",
                            "Parameters": {
                                "AccessKey": "",
                                "NameSpace": "Windows/Default",
                                "Region": "${AWS::Region}",
                                "SecretKey": ""
                            }
                        }
                    ],
                    "Flows": {
                        "Flows": [
                            "ApplicationEventLog,CloudWatchApplicationEventLog",
                            "SystemEventLog,CloudWatchSystemEventLog",
                            "SecurityEventLog,CloudWatchSecurityEventLog",
                            "EC2ConfigLog,CloudWatchEC2ConfigLog",
                            "CfnInitLog,CloudWatchCfnInitLog",
                            "IISLogs,CloudWatchIISLogs",
                            "MemoryPerformanceCounter,CloudWatch"
                        ]
                    },
                    "PollInterval": "00:00:05"
                },
                "IsEnabled": true
            }
    commands:
      0-enableSSM:
        command: >-
          powershell.exe -Command "Set-Service -Name AmazonSSMAgent
          -StartupType Automatic"
        waitAfterCompletion: '0'
      1-restartSSM:
        command: powershell.exe -Command "Restart-Service AmazonSSMAgent "
        waitAfterCompletion: '30'
  01-InstallWebServer:
    commands:
      01_install_webserver:
        command: >-
          powershell.exe -Command "Install-WindowsFeature Web-Server
          -IncludeAllSubFeature"
        waitAfterCompletion: '0'
  02-ConfigureApplication:
    files:
      'c:\Inetpub\wwwroot\index.htm':
        content: >-
```

```
                <html> <head> <title>Test Application Page</title> </head>
                <body> <h1>Congratulations !! Your IIS server is
                configured.</h1> </body> </html>
        03-Finalize:
          commands:
            00_signal_success:
              command: !Sub >-
                cfn-signal.exe -e 0 --resource WebServerHost --stack
                ${AWS::StackName} --region ${AWS::Region}
              waitAfterCompletion: '0'
    Properties:
      KeyName: !Ref KeyPair
      ImageId: !FindInMap
        - AWSAMIRegionMap
        - !Ref 'AWS::Region'
        - WS2012R2
      InstanceType: t2.xlarge
      SecurityGroupIds:
        - !Ref WebServerSecurityGroup
      IamInstanceProfile: !Ref LogRoleInstanceProfile
      UserData: !Base64
        'Fn::Sub': >
          <script>

          wmic product where "description='Amazon SSM Agent' " uninstall

          wmic product where "description='aws-cfn-bootstrap' " uninstall

          start /wait c:\\Windows\\system32\\msiexec /passive /qn /i
          https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi

          powershell.exe -Command "iwr
          https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe
          -UseBasicParsing -OutFile C:\\AmazonSSMAgentSetup.exe"

          start /wait C:\\AmazonSSMAgentSetup.exe /install /quiet

          cfn-init.exe -v -c config -s ${AWS::StackName} --resource
          WebServerHost --region ${AWS::Region}

          </script>
  LogGroup:
```

```
    Type: 'AWS::Logs::LogGroup'
    Properties:
      RetentionInDays: 7
  404MetricFilter:
    Type: 'AWS::Logs::MetricFilter'
    Properties:
      LogGroupName: !Ref LogGroup
      FilterPattern: >-
        [timestamps, serverip, method, uri, query, port, dash, clientip,
        useragent, status_code = 404, ...]
      MetricTransformations:
        - MetricValue: '1'
          MetricNamespace: test/404s
          MetricName: test404Count
  404Alarm:
    Type: 'AWS::CloudWatch::Alarm'
    Properties:
      AlarmDescription: The number of 404s is greater than 2 over 2 minutes
      MetricName: test404Count
      Namespace: test/404s
      Statistic: Sum
      Period: '60'
      EvaluationPeriods: '2'
      Threshold: '2'
      AlarmActions:
        - !Ref AlarmNotificationTopic
      ComparisonOperator: GreaterThanThreshold
  AlarmNotificationTopic:
    Type: 'AWS::SNS::Topic'
    Properties:
      Subscription:
        - Endpoint: !Ref OperatorEmail
          Protocol: email
Outputs:
  InstanceId:
    Description: The instance ID of the web server
    Value: !Ref WebServerHost
  WebsiteURL:
    Value: !Sub 'http://${WebServerHost.PublicDnsName}'
    Description: URL for newly created IIS web server
  PublicIP:
    Description: Public IP address of the web server
    Value: !GetAtt
      - WebServerHost
```

```
      - PublicIp
  CloudWatchLogGroupName:
    Description: The name of the CloudWatch log group
    Value: !Ref LogGroup
```

Weitere Informationen finden Sie auch unter

Weitere Informationen zu CloudWatch Logs-Ressourcen finden Sie unter oder. AWS::Logs::LogGroupAWS::Logs::MetricFilter

Amazon DynamoDB-Vorlagenausschnitte

Themen

- Auto Scaling von Anwendungen mit einer Amazon DynamoDB-Tabelle
- Weitere Informationen finden Sie auch unter

Auto Scaling von Anwendungen mit einer Amazon DynamoDB-Tabelle

In diesem Beispiel wird Application Auto Scaling für eine `AWS::DynamoDB::Table`-Ressource eingerichtet. Die Vorlage definiert eine `TargetTrackingScaling`-Skalierungsrichtline, die den `WriteCapacityUnits`-Durchsatz für die Tabelle skaliert.

JSON

```
{
    "Resources": {
        "DDBTable": {
            "Type": "AWS::DynamoDB::Table",
            "Properties": {
                "AttributeDefinitions": [
                    {
                        "AttributeName": "ArtistId",
                        "AttributeType": "S"
                    },
                    {
                        "AttributeName": "Concert",
                        "AttributeType": "S"
                    },
                    {
                        "AttributeName": "TicketSales",
                        "AttributeType": "S"
```

```
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "ArtistId",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "Concert",
                        "KeyType": "RANGE"
                    }
                ],
                "GlobalSecondaryIndexes": [
                    {
                        "IndexName": "GSI",
                        "KeySchema": [
                            {
                                "AttributeName": "TicketSales",
                                "KeyType": "HASH"
                            }
                        ],
                        "Projection": {
                            "ProjectionType": "KEYS_ONLY"
                        },
                        "ProvisionedThroughput": {
                            "ReadCapacityUnits": 5,
                            "WriteCapacityUnits": 5
                        }
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 5
                }
            }
        },
        "WriteCapacityScalableTarget": {
            "Type": "AWS::ApplicationAutoScaling::ScalableTarget",
            "Properties": {
                "MaxCapacity": 15,
                "MinCapacity": 5,
                "ResourceId": {
                    "Fn::Join": [
                        "/",
```

```
                        [
                            "table",
                            {
                                "Ref": "DDBTable"
                            }
                        ]
                    ]
                },
                "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable" },
                "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
                "ServiceNamespace": "dynamodb"
            }
        },
        "WriteScalingPolicy": {
            "Type": "AWS::ApplicationAutoScaling::ScalingPolicy",
            "Properties": {
                "PolicyName": "WriteAutoScalingPolicy",
                "PolicyType": "TargetTrackingScaling",
                "ScalingTargetId": {
                    "Ref": "WriteCapacityScalableTarget"
                },
                "TargetTrackingScalingPolicyConfiguration": {
                    "TargetValue": 50,
                    "ScaleInCooldown": 60,
                    "ScaleOutCooldown": 60,
                    "PredefinedMetricSpecification": {
                        "PredefinedMetricType": "DynamoDBWriteCapacityUtilization"
                    }
                }
            }
        }
    }
}
```

YAML

```
Resources:
  DDBTable:
    Type: AWS::DynamoDB::Table
    Properties:
      AttributeDefinitions:
```

```
        - AttributeName: "ArtistId"
          AttributeType: "S"
        - AttributeName: "Concert"
          AttributeType: "S"
        - AttributeName: "TicketSales"
          AttributeType: "S"
      KeySchema:
        - AttributeName: "ArtistId"
          KeyType: "HASH"
        - AttributeName: "Concert"
          KeyType: "RANGE"
      GlobalSecondaryIndexes:
        - IndexName: "GSI"
          KeySchema:
            - AttributeName: "TicketSales"
              KeyType: "HASH"
          Projection:
            ProjectionType: "KEYS_ONLY"
          ProvisionedThroughput:
            ReadCapacityUnits: 5
            WriteCapacityUnits: 5
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 5
  WriteCapacityScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 15
      MinCapacity: 5
      ResourceId: !Join
        - /
        - - table
          - !Ref DDBTable
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable'
      ScalableDimension: dynamodb:table:WriteCapacityUnits
      ServiceNamespace: dynamodb
  WriteScalingPolicy:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: WriteAutoScalingPolicy
      PolicyType: TargetTrackingScaling
```

```
      ScalingTargetId: !Ref WriteCapacityScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 50.0
        ScaleInCooldown: 60
        ScaleOutCooldown: 60
        PredefinedMetricSpecification:
          PredefinedMetricType: DynamoDBWriteCapacityUtilization
```

Weitere Informationen finden Sie auch unter

Weitere Informationen finden Sie im Blogbeitrag How to use AWS CloudFormation to configure auto scaling for DynamoDB tables and indexes im AWS Datenbank-Blog.

Weitere Informationen zu DynamoDB-Ressourcen finden Sie unter. AWS::DynamoDB::Table

EC2 CloudFormation Amazon-Vorlagenschnipsel

Amazon EC2 bietet skalierbare Rechenkapazität in der AWS Cloud. Sie können Amazon verwenden EC2 , um so viele oder so wenige virtuelle Server zu starten, wie Sie benötigen, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten. Diese virtuellen Server, die als Instances bezeichnet werden, können eine Vielzahl von Betriebssystemen und Anwendungen ausführen und sind an Ihre jeweiligen Anforderungen anpassbar. Amazon EC2 ermöglicht es Ihnen, nach oben oder unten zu skalieren, um Änderungen der Anforderungen oder Nutzungsspitzen zu bewältigen.

Mithilfe von CloudFormation Vorlagen können Sie EC2 Amazon-Instances als Teil Ihrer Infrastruktur definieren und bereitstellen. Vorlagen machen es einfach, den Einsatz von EC2 Amazon-Ressourcen wiederholbar und konsistent zu verwalten und zu automatisieren.

Die folgenden Beispielvorlagen beschreiben CloudFormation Ressourcen oder Komponenten für Amazon. EC2 Diese Ausschnitte sind für die Integration in eine Vorlage gedacht, nicht für eine unabhängige Ausführung.

Kategorien von Ausschnitten

- Konfigurieren Sie EC2 Amazon-Instances mit CloudFormation
- Erstellen Sie Startvorlagen mit CloudFormation
- Sicherheitsgruppen verwalten mit CloudFormation
- Elastic IP-Adressen zuweisen und verknüpfen mit CloudFormation
- Konfigurieren Sie Amazon VPC-Ressourcen mit CloudFormation

Konfigurieren Sie EC2 Amazon-Instances mit CloudFormation

Die folgenden Ausschnitte zeigen, wie Sie EC2 Amazon-Instances mithilfe von konfigurieren. CloudFormation

Kategorien von Ausschnitten

- Allgemeine EC2 Amazon-Konfigurationen
- Angeben der Blockgerät-Zuweisungen für eine Instance

Allgemeine EC2 Amazon-Konfigurationen

Die folgenden Ausschnitte veranschaulichen allgemeine Konfigurationen für die Verwendung von EC2 Amazon-Instances. CloudFormation

Beispielausschnitte

- Erstellen Sie eine EC2 Amazon-Instance in einer bestimmten Availability Zone
- Konfiguration einer markierten EC2 Amazon-Instance mit einem EBS-Volume und Benutzerdaten
- Definieren Sie den DynamoDB-Tabellennamen in den Benutzerdaten für den Start der Amazon-Instanz EC2
- Erstellen eines Amazon-EBS-Volumes mit DeletionPolicy

Erstellen Sie eine EC2 Amazon-Instance in einer bestimmten Availability Zone

Das folgende Snippet erstellt mithilfe einer Ressource eine EC2 Amazon-Instance in der angegebenen Availability Zone. AWS::EC2::Instance Der Code für eine Availability Zone ist der Regionscode gefolgt von einem Buchstaben als Bezeichner. Eine Instance können Sie in einer einzigen Availability Zone starten.

JSON

```
"Ec2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
        "AvailabilityZone": "aa-example-1a",
        "ImageId": "ami-1234567890abcdef0"
    }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    AvailabilityZone: aa-example-1a
    ImageId: ami-1234567890abcdef0
```

Konfiguration einer markierten EC2 Amazon-Instance mit einem EBS-Volume und Benutzerdaten

Der folgende Ausschnitt erstellt eine EC2 Amazon-Instance mit einem Tag, einem EBS-Volume und Benutzerdaten. Es verwendet eine Ressource. AWS::EC2::Instance In derselben Vorlage müssen Sie eine AWS::EC2::SecurityGroupRessource, eine AWS::SNS::TopicRessource und eine AWS::EC2::VolumeRessource definieren. Der `KeyName` muss im Abschnitt `Parameters` der Vorlage definiert werden.

Mithilfe von Tags können Sie AWS Ressourcen anhand Ihrer Präferenzen kategorisieren, z. B. nach Zweck, Eigentümer oder Umgebung. Benutzerdaten ermöglichen die Bereitstellung benutzerdefinierter Skripts oder Daten für eine Instance während des Starts. Diese Daten erleichtern die Automatisierung von Aufgaben, die Softwarekonfiguration, die Paketinstallation und weitere Aktionen auf einer Instance während der Initialisierung.

Weitere Informationen zum Taggen Ihrer Ressourcen finden Sie unter Taggen Ihrer EC2 Amazon-Ressourcen im EC2 Amazon-Benutzerhandbuch.

Informationen zu Benutzerdaten finden Sie unter Verwenden von Instance-Metadaten zur Verwaltung Ihrer EC2 Instance im EC2 Amazon-Benutzerhandbuch.

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "KeyName": { "Ref": "KeyName" },
    "SecurityGroups": [ { "Ref": "Ec2SecurityGroup" } ],
    "UserData": {
      "Fn::Base64": {
        "Fn::Join": [ ":", [
            "PORT=80",
            "TOPIC=",
```

```
            { "Ref": "MySNSTopic" }
          ]
        ]
      }
    },
    "InstanceType": "aa.size",
    "AvailabilityZone": "aa-example-1a",
    "ImageId": "ami-1234567890abcdef0",
    "Volumes": [
      {
        "VolumeId": { "Ref": "MyVolumeResource" },
        "Device": "/dev/sdk"
      }
    ],
    "Tags": [ { "Key": "Name", "Value": "MyTag" } ]
  }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    KeyName: !Ref KeyName
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    UserData:
      Fn::Base64:
        Fn::Join:
          - ":"
          - - "PORT=80"
            - "TOPIC="
            - !Ref MySNSTopic
    InstanceType: aa.size
    AvailabilityZone: aa-example-1a
    ImageId: ami-1234567890abcdef0
    Volumes:
      - VolumeId: !Ref MyVolumeResource
        Device: "/dev/sdk"
    Tags:
      - Key: Name
        Value: MyTag
```

Definieren Sie den DynamoDB-Tabellennamen in den Benutzerdaten für den Start der Amazon-Instanz EC2

Der folgende Ausschnitt erstellt eine EC2 Amazon-Instance und definiert einen DynamoDB-Tabellennamen in den Benutzerdaten, der beim Start an die Instance übergeben wird. Es verwendet eine Ressource. AWS::EC2::Instance Sie können Parameter oder dynamische Werte in den Benutzerdaten definieren, um eine EC2 Instanz beim Start zu übergeben.

Weitere Informationen zu Benutzerdaten finden Sie unter Verwenden von Instance-Metadaten zur Verwaltung Ihrer EC2 Instance im EC2 Amazon-Benutzerhandbuch.

JSON

```
"Ec2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
        "UserData": {
            "Fn::Base64": {
                "Fn::Join": [
                    "",
                    [
                        "TableName=",
                        {
                            "Ref": "DynamoDBTableName"
                        }
                    ]
                ]
            }
        },
        "AvailabilityZone": "aa-example-1a",
        "ImageId": "ami-1234567890abcdef0"
    }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    UserData:
      Fn::Base64:
        Fn::Join:
          - ''
```

```
          - - 'TableName='
            - Ref: DynamoDBTableName
    AvailabilityZone: aa-example-1a
    ImageId: ami-1234567890abcdef0
```

Erstellen eines Amazon-EBS-Volumes mit **`DeletionPolicy`**

Die folgenden Codefragmente erstellen mithilfe einer Amazon-Ressource ein Amazon EBS-Volume. EC2 AWS::EC2::Volume Sie können das Volume mit der Eigenschaft `Size` oder `SnapshotID` (aber nicht mit beiden) definieren. Ein `DeletionPolicy`-Attribut ist so eingestellt, dass beim Löschen des Stacks ein Snapshot des Volumes erstellt wird.

Weitere Informationen zum Attribut `DeletionPolicy` finden Sie unter Attribut DeletionPolicy.

Weitere Informationen zum Erstellen von Amazon-EBS-Volumes finden Sie unter Erstellen eines Amazon-EBS-Volumes.

JSON

Mit diesem Ausschnitt wird ein Amazon-EBS-Volume mit einer angegebenen Größe erstellt. Die Größe ist auf 10 eingestellt, Sie können sie jedoch nach Bedarf anpassen. Mit der AWS::EC2::VolumeRessource können Sie entweder die Größe oder eine Snapshot-ID angeben, aber nicht beides.

```
"MyEBSVolume": {
    "Type": "AWS::EC2::Volume",
    "Properties": {
        "Size": "10",
        "AvailabilityZone": {
            "Ref": "AvailabilityZone"
        }
    },
    "DeletionPolicy": "Snapshot"
}
```

Mit diesem Ausschnitt wird ein Amazon-EBS-Volume mit einer angegebenen Snapshot-ID erstellt. Mit der AWS::EC2::VolumeRessource können Sie entweder die Größe oder eine Snapshot-ID angeben, jedoch nicht beides.

```
"MyEBSVolume": {
    "Type": "AWS::EC2::Volume",
    "Properties": {
```

```
        "SnapshotId" : "snap-1234567890abcdef0",
        "AvailabilityZone": {
            "Ref": "AvailabilityZone"
        }
    },
    "DeletionPolicy": "Snapshot"
}
```

YAML

Mit diesem Ausschnitt wird ein Amazon-EBS-Volume mit einer angegebenen Größe erstellt. Die Größe ist auf 10 eingestellt, Sie können sie jedoch nach Bedarf anpassen. Mit der AWS::EC2::VolumeRessource können Sie entweder die Größe oder eine Snapshot-ID angeben, jedoch nicht beides.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 10
    AvailabilityZone:
      Ref: AvailabilityZone
  DeletionPolicy: Snapshot
```

Mit diesem Ausschnitt wird ein Amazon-EBS-Volume mit einer angegebenen Snapshot-ID erstellt. Mit der AWS::EC2::VolumeRessource können Sie entweder die Größe oder eine Snapshot-ID angeben, jedoch nicht beides.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    SnapshotId: snap-1234567890abcdef0
    AvailabilityZone:
      Ref: AvailabilityZone
  DeletionPolicy: Snapshot
```

Angeben der Blockgerät-Zuweisungen für eine Instance

Eine Blockgerät-Zuweisung definiert die Blockgeräte (Instance-Speicher-Volumes und EBS-Volumes), die an eine Instance angefügt werden sollen. Eine Blockgerät-Zuweisung können Sie beim Erstellen eines AMI angeben, sodass die Zuweisung von allen Instances verwendet wird, die über das AMI gestartet werden. Alternativ können Sie eine Blockgerät-Zuweisung beim Starten einer

Instance angeben. Diese Zuweisung gilt dann statt der im AMI angegebenen, über das die Instance gestartet wurde.

Sie können die folgenden Vorlagenausschnitte verwenden, um die Blockgerätezuordnungen für Ihre EBS- oder Instance-Speicher-Volumes mithilfe der `BlockDeviceMappings` Eigenschaft einer Ressource anzugeben. AWS::EC2::Instance

Weitere Informationen zu Blockgerätezuordnungen finden Sie unter Blockgerätezuordnungen für Volumes auf EC2 Amazon-Instances im Amazon-Benutzerhandbuch. EC2

Szenarien

- Angeben der Blockgerät-Zuweisungen für zwei EBS-Volumes
- Anzeigen der Blockgerät-Zuweisung für ein Instance-Speicher-Volume

Angeben der Blockgerät-Zuweisungen für zwei EBS-Volumes

JSON

```
"Ec2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "ImageId": {
        "Fn::FindInMap": [
          "AWSRegionArch2AMI",
          { "Ref": "AWS::Region" },
          {
            "Fn::FindInMap": [
              "AWSInstanceType2Arch",
              { "Ref": "InstanceType" },
              "Arch"
            ]
          }
        ]
      },
      "KeyName": { "Ref": "KeyName" },
      "InstanceType": { "Ref": "InstanceType" },
      "SecurityGroups": [{ "Ref": "Ec2SecurityGroup" }],
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/sda1",
          "Ebs": { "VolumeSize": "50" }
        },
```

```
        {
          "DeviceName": "/dev/sdm",
          "Ebs": { "VolumeSize": "100" }
        }
      ]
    }
  }
}
```

YAML

```
EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !FindInMap [AWSRegionArch2AMI, !Ref 'AWS::Region', !FindInMap
 [AWSInstanceType2Arch, !Ref InstanceType, Arch]]
      KeyName: !Ref KeyName
      InstanceType: !Ref InstanceType
      SecurityGroups:
        - !Ref Ec2SecurityGroup
      BlockDeviceMappings:
        -
          DeviceName: /dev/sda1
          Ebs:
            VolumeSize: 50
        -
          DeviceName: /dev/sdm
          Ebs:
            VolumeSize: 100
```

Anzeigen der Blockgerät-Zuweisung für ein Instance-Speicher-Volume

JSON

```
"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
 "HVM64" ]},
    "KeyName" : { "Ref" : "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups" : [{ "Ref" : "Ec2SecurityGroup" }],
    "BlockDeviceMappings" : [
```

```
      {
        "DeviceName"  : "/dev/sdc",
        "VirtualName" : "ephemeral0"
      }
    ]
  }
}
```

YAML

```
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !FindInMap
      - AWSRegionArch2AMI
      - !Ref AWS::Region
      - HVM64
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      - DeviceName: /dev/sdc
        VirtualName: ephemeral0
```

Erstellen Sie Startvorlagen mit CloudFormation

Dieser Abschnitt enthält ein Beispiel für die Erstellung einer EC2 Amazon-Startvorlage mit CloudFormation. Mithilfe von Startvorlagen können Sie Vorlagen für die Konfiguration und Bereitstellung von EC2 Amazon-Instances innerhalb von AWS Vorlagen erstellen. Mit Startvorlagen können Sie Startparameter speichern, sodass Sie sie nicht bei jedem Start einer Instance neu angeben müssen. Weitere Beispiele finden Sie im Abschnitt Beispiele in der `AWS::EC2::LaunchTemplate` Ressource.

Weitere Informationen zu Startvorlagen finden Sie unter Startparameter für Store-Instances in EC2 Amazon-Startvorlagen im EC2 Amazon-Benutzerhandbuch.

Informationen zum Erstellen von Startvorlagen für die Verwendung mit Auto Scaling Scaling-Gruppen finden Sie unter Auto Scaling Scaling-Startvorlagen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Kategorien von Ausschnitten

- Erstellen Sie eine Startvorlage, die Sicherheitsgruppen, Tags, Benutzerdaten und eine IAM-Rolle angibt

Erstellen Sie eine Startvorlage, die Sicherheitsgruppen, Tags, Benutzerdaten und eine IAM-Rolle angibt

Dieses Snippet zeigt eine AWS::EC2::LaunchTemplateRessource, die die Konfigurationsinformationen zum Starten einer Instance enthält. Sie definieren Werte für die `ImageId`-, `InstanceType`-, `SecurityGroups`-, `UserData`- und `TagSpecifications`-Eigenschaften. Die `SecurityGroups` Eigenschaft gibt eine vorhandene EC2 Sicherheitsgruppe und eine neue Sicherheitsgruppe an. Die `Ref` Funktion ruft die ID der AWS::EC2::SecurityGroupRessource ab`myNewEC2SecurityGroup`, die an anderer Stelle in der Stack-Vorlage deklariert wurde.

Die Einführungsvorlage enthält einen Abschnitt für benutzerdefinierte Benutzerdaten. In diesem Abschnitt können Sie Konfigurations-Aufgaben und Skripts übergeben, die beim Launch einer Instance ausgeführt werden. In diesem Beispiel installieren die Benutzerdaten den AWS Systems Manager Agenten und starten den Agenten.

Die Startvorlage enthält auch eine IAM-Rolle, die es Anwendungen, die auf Instances ausgeführt werden, ermöglicht, Aktionen in Ihrem Namen auszuführen. Dieses Beispiel zeigt eine AWS::IAM::RoleRessource für die Startvorlage, die die `IamInstanceProfile` Eigenschaft verwendet, um die IAM-Rolle zu spezifizieren. Die `Ref` Funktion ruft den Namen der AWS::IAM::InstanceProfileRessource `myInstanceProfile` ab. Um die Berechtigungen der IAM-Rolle zu konfigurieren, geben Sie einen Wert für die `ManagedPolicyArns`-Eigenschaft an.

JSON

```
{
  "Resources":{
    "myLaunchTemplate":{
      "Type":"AWS::EC2::LaunchTemplate",
      "Properties":{
        "LaunchTemplateName":{ "Fn::Sub": "${AWS::StackName}-launch-template" },
        "LaunchTemplateData":{
          "ImageId":"ami-02354e95b3example",
          "InstanceType":"t3.micro",
          "IamInstanceProfile":{
            "Name":{
              "Ref":"myInstanceProfile"
```

```
            }
          },
          "SecurityGroupIds":[
            {
              "Ref":"myNewEC2SecurityGroup"
            },
            "sg-083cd3bfb8example"
          ],
          "UserData":{
            "Fn::Base64":{
              "Fn::Join": [
                "", [
                  "#!/bin/bash\n",
                  "cd /tmp\n",
                  "yum install -y https://s3.amazonaws.com/ec2-downloads-windows/
SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm\n",
                  "systemctl enable amazon-ssm-agent\n",
                  "systemctl start amazon-ssm-agent\n"
                ]
              ]
            }
          },
          "TagSpecifications":[
            {
              "ResourceType":"instance",
              "Tags":[
                {
                  "Key":"environment",
                  "Value":"development"
                }
              ]
            },
            {
              "ResourceType":"volume",
              "Tags":[
                {
                  "Key":"environment",
                  "Value":"development"
                }
              ]
            }
          ]
        }
      }
```

```
    },
    "myInstanceRole":{
      "Type":"AWS::IAM::Role",
      "Properties":{
        "RoleName":"InstanceRole",
        "AssumeRolePolicyDocument":{
          "Version":"2012-10-17",
          "Statement":[
            {
              "Effect":"Allow",
              "Principal":{
                "Service":[
                  "ec2.amazonaws.com"
                ]
              },
              "Action":[
                "sts:AssumeRole"
              ]
            }
          ]
        },
        "ManagedPolicyArns":[
          "arn:aws:iam::aws:policy/myCustomerManagedPolicy"
        ]
      }
    },
    "myInstanceProfile":{
      "Type":"AWS::IAM::InstanceProfile",
      "Properties":{
        "Path":"/",
        "Roles":[
          {
            "Ref":"myInstanceRole"
          }
        ]
      }
    }
  }
}
```

YAML

```
---
```

```
Resources:
  myLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
      LaunchTemplateData:
        ImageId: ami-02354e95b3example
        InstanceType: t3.micro
        IamInstanceProfile:
          Name: !Ref myInstanceProfile
        SecurityGroupIds:
        - !Ref myNewEC2SecurityGroup
        - sg-083cd3bfb8example
        UserData:
          Fn::Base64: !Sub |
            #!/bin/bash
            cd /tmp
            yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
            systemctl enable amazon-ssm-agent
            systemctl start amazon-ssm-agent
        TagSpecifications:
        - ResourceType: instance
          Tags:
          - Key: environment
            Value: development
        - ResourceType: volume
          Tags:
          - Key: environment
            Value: development
  myInstanceRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: InstanceRole
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Effect: 'Allow'
          Principal:
            Service:
            - 'ec2.amazonaws.com'
          Action:
          - 'sts:AssumeRole'
      ManagedPolicyArns:
```

```
        - 'arn:aws:iam::aws:policy/myCustomerManagedPolicy'
  myInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: '/'
      Roles:
      - !Ref myInstanceRole
```

Sicherheitsgruppen verwalten mit CloudFormation

Die folgenden Ausschnitte zeigen, wie Sie CloudFormation Sicherheitsgruppen und EC2 Amazon-Instances verwalten können, um den Zugriff auf Ihre AWS Ressourcen zu kontrollieren.

Kategorien von Ausschnitten

- Ordnen Sie eine EC2 Amazon-Instance einer Sicherheitsgruppe zu
- Erstellen von Sicherheitsgruppen mit Regeln für eingehenden Datenverkehr
- Erstellen eines Elastic Load Balancers mit einer Regel für eingehenden Datenverkehr für eine Sicherheitsgruppe

Ordnen Sie eine EC2 Amazon-Instance einer Sicherheitsgruppe zu

Die folgenden Beispielausschnitte zeigen, wie Sie eine EC2 Amazon-Instance mit einer standardmäßigen Amazon VPC-Sicherheitsgruppe verknüpfen, indem Sie. CloudFormation

Beispielausschnitte

- Ordnen Sie eine EC2 Amazon-Instance einer Standard-VPC-Sicherheitsgruppe zu
- Erstellen Sie eine EC2 Amazon-Instance mit einem angehängten Volume und einer Sicherheitsgruppe

Ordnen Sie eine EC2 Amazon-Instance einer Standard-VPC-Sicherheitsgruppe zu

Das folgende Snippet erstellt eine Amazon-VPC, ein Subnetz innerhalb der VPC und eine Amazon-Instance. EC2 Die VPC wird mithilfe einer AWS::EC2: :VPC-Ressource erstellt. Der IP-Adressbereich für die VPC wird in der größeren Vorlage definiert und durch den Parameter `MyVPCCIDRRange` referenziert.

Ein Subnetz wird innerhalb der VPC mithilfe einer AWS:::EC2: Subnetzressource erstellt. Das Subnetz wird mit der VPC verknüpft, die als `MyVPC` referenziert wird.

Eine EC2 Instance wird innerhalb der VPC und des Subnetzes mithilfe einer AWS::EC2::InstanceRessource gestartet. Diese Ressource gibt das Amazon Machine Image (AMI) an, das zum Starten der Instance verwendet werden soll, das Subnetz, in dem die Instance ausgeführt wird, und die mit der Instance zu verknüpfende Sicherheitsgruppe. Die `Fn::FindInMap` Funktion wird verwendet, um Werte aus dem in der Vorlage definierten `AWSRegionToAMI` Mapping abzurufen und den Wert `ImageId` für die AWS::EC2::InstanceRessource zu ermitteln.

Die ID der Sicherheitsgruppe wird mit der Funktion `Fn::GetAtt` abgerufen, die die Standardsicherheitsgruppe aus der Ressource `MyVPC` abruft.

Die Instance wird in der im Ausschnitt definierten Ressource `MySubnet` platziert.

Wenn Sie eine VPC mit erstellen CloudFormation, erstellt AWS automatisch Standardressourcen innerhalb der VPC, einschließlich einer Standardsicherheitsgruppe. Wenn Sie jedoch eine VPC innerhalb einer CloudFormation Vorlage definieren, haben Sie beim Erstellen IDs der Vorlage möglicherweise keinen Zugriff auf diese Standardressourcen. Um auf die in der Vorlage angegebenen Standardressourcen zuzugreifen und diese zu nutzen, können Sie intrinsische Funktionen wie etwa `Fn::GetAtt` verwenden. Mithilfe dieser Funktion können Sie mit den Standardressourcen arbeiten, die von CloudFormation automatisch erstellt werden.

JSON

```
"MyVPC": {
    "Type": "AWS::EC2::VPC",
    "Properties": {
        "CidrBlock": {
            "Ref": "MyVPCCIDRRange"
        },
        "EnableDnsSupport": false,
        "EnableDnsHostnames": false,
        "InstanceTenancy": "default"
    }
},
"MySubnet": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
        "CidrBlock": {
            "Ref": "MyVPCCIDRRange"
        },
        "VpcId": {
            "Ref": "MyVPC"
```

```
        }
    }
},
"MyInstance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
        "ImageId": {
            "Fn::FindInMap": [
                "AWSRegionToAMI",
                {
                    "Ref": "AWS::Region"
                },
                "64"
            ]
        },
        "SecurityGroupIds": [
            {
                "Fn::GetAtt": [
                    "MyVPC",
                    "DefaultSecurityGroup"
                ]
            }
        ],
        "SubnetId": {
            "Ref": "MySubnet"
        }
    }
}
```

YAML

```
MyVPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    EnableDnsSupport: false
    EnableDnsHostnames: false
    InstanceTenancy: default
MySubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock:
```

```
      Ref: MyVPCCIDRRange
    VpcId:
      Ref: MyVPC
MyInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId:
      Fn::FindInMap:
        - AWSRegionToAMI
        - Ref: AWS::Region
        - "64"
    SecurityGroupIds:
      - Fn::GetAtt:
          - MyVPC
          - DefaultSecurityGroup
    SubnetId:
      Ref: MySubnet
```

Erstellen Sie eine EC2 Amazon-Instance mit einem angehängten Volume und einer Sicherheitsgruppe

Das folgende Snippet erstellt eine EC2 Amazon-Instance mithilfe einer AWS::EC2::InstanceRessource, die von einem bestimmten AMI aus gestartet wird. Die Instance ist einer Sicherheitsgruppe zugeordnet, die eingehenden SSH-Verkehr über Port 22 von einer bestimmten IP-Adresse aus unter Verwendung einer Ressource zulässt. AWS::EC2::SecurityGroup Es erstellt mithilfe einer AWS::EC2::VolumeRessource ein 100-GB-Amazon-EBS-Volume. Das Volume wird in derselben Availability Zone wie die Instance erstellt (wie in der Funktion `GetAtt` angegeben) und in der Instance auf dem Gerät `/dev/sdh` gemountet.

Weitere Informationen zum Erstellen von Amazon-EBS-Volumes finden Sie unter Erstellen eines Amazon-EBS-Volumes.

JSON

```
"Ec2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
        "SecurityGroups": [
            {
                "Ref": "InstanceSecurityGroup"
            }
        ],
```

```
        "ImageId": "ami-1234567890abcdef0"
    }
},
"InstanceSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Enable SSH access via port 22",
        "SecurityGroupIngress": [
            {
                "IpProtocol": "tcp",
                "FromPort": "22",
                "ToPort": "22",
                "CidrIp": "192.0.2.0/24"
            }
        ]
    }
},
"NewVolume": {
    "Type": "AWS::EC2::Volume",
    "Properties": {
        "Size": "100",
        "AvailabilityZone": {
            "Fn::GetAtt": [
                "Ec2Instance",
                "AvailabilityZone"
            ]
        }
    }
},
"MountPoint": {
    "Type": "AWS::EC2::VolumeAttachment",
    "Properties": {
        "InstanceId": {
            "Ref": "Ec2Instance"
        },
        "VolumeId": {
            "Ref": "NewVolume"
        },
        "Device": "/dev/sdh"
    }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    SecurityGroups:
      - !Ref InstanceSecurityGroup
    ImageId: ami-1234567890abcdef0
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 192.0.2.0/24
NewVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 100
    AvailabilityZone: !GetAtt [Ec2Instance, AvailabilityZone]
MountPoint:
  Type: AWS::EC2::VolumeAttachment
  Properties:
    InstanceId: !Ref Ec2Instance
    VolumeId: !Ref NewVolume
    Device: /dev/sdh
```

Erstellen von Sicherheitsgruppen mit Regeln für eingehenden Datenverkehr

Die folgenden Beispielausschnitte zeigen, wie mit CloudFormation Sicherheitsgruppen mit bestimmten Regeln für eingehenden Datenverkehr konfiguriert werden können.

Ausschnitte

- Erstellen einer Sicherheitsgruppe mit Regeln für eingehenden Datenverkehr für SSH- und HTTP-Zugriff
- Erstellen einer Sicherheitsgruppe mit Regeln für eingehenden Datenverkehr für SSH- und HTTP-Zugriff aus angegebenen CIDR-Bereichen
- Erstellen von miteinander kommunizierenden Sicherheitsgruppen mit Regeln für eingehenden Datenverkehr

Erstellen einer Sicherheitsgruppe mit Regeln für eingehenden Datenverkehr für SSH- und HTTP-Zugriff

Der folgende Ausschnitt beschreibt zwei Regeln für den Zugriff auf Sicherheitsgruppen, die eine Ressource verwenden. AWS::EC2::SecurityGroup Die erste Eingangsregel ermöglicht den SSH-Zugriff (Port 22) von einer vorhandenen Sicherheitsgruppe mit dem Namen`MyAdminSecurityGroup`, die dem AWS Konto mit der Kontonummer gehört. `1111-2222-3333` Die zweite Regel für eingehenden Datenverkehr erlaubt den HTTP-Zugriff (Port 80) aus einer anderen Sicherheitsgruppe namens `MySecurityGroupCreatedInCFN`, die in derselben Vorlage erstellt wurde. Mithilfe der Funktion `Ref` wird der logische Name der Sicherheitsgruppe referenziert, die in derselben Vorlage erstellt wurde.

In der ersten Regel für eingehenden Datenverkehr müssen Sie sowohl für die Eigenschaft `SourceSecurityGroupName` als auch für die Eigenschaft `SourceSecurityGroupOwnerId` einen Wert hinzufügen. In der zweiten Regel für eingehenden Datenverkehr referenziert `MySecurityGroupCreatedInCFNTemplate` eine andere Sicherheitsgruppe, die in derselben Vorlage erstellt wurde. Vergewissern Sie sich, dass der logische Name `MySecurityGroupCreatedInCFNTemplate` mit dem tatsächlichen logischen Namen der Sicherheitsgruppe übereinstimmt, die Sie in der größeren Vorlage angeben.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter EC2 Amazon-Sicherheitsgruppen für Ihre EC2 Amazon-Instances im EC2 Amazon-Benutzerhandbuch.

JSON

```
"SecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Allow connections from specified source security group",
        "SecurityGroupIngress": [
            {
                "IpProtocol": "tcp",
                "FromPort": "22",
                "ToPort": "22",
                "SourceSecurityGroupName": "MyAdminSecurityGroup",
                "SourceSecurityGroupOwnerId": "1111-2222-3333"
            },
            {
                "IpProtocol": "tcp",
                "FromPort": "80",
                "ToPort": "80",
```

```
                "SourceSecurityGroupName": {
                    "Ref": "MySecurityGroupCreatedInCFNTemplate"
                }
            }
        ]
    }
}
```

YAML

```
SecurityGroup:
  Type: 'AWS::EC2::SecurityGroup'
  Properties:
    GroupDescription: Allow connections from specified source security group
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
        SourceSecurityGroupName: MyAdminSecurityGroup
        SourceSecurityGroupOwnerId: '1111-2222-3333'
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        SourceSecurityGroupName:
          Ref: MySecurityGroupCreatedInCFNTemplate
```

Erstellen einer Sicherheitsgruppe mit Regeln für eingehenden Datenverkehr für SSH- und HTTP-Zugriff aus angegebenen CIDR-Bereichen

Das folgende Snippet erstellt eine Sicherheitsgruppe für eine EC2 Amazon-Instance mit zwei Regeln für eingehenden Datenverkehr. Diese Regeln erlauben eingehenden TCP-Verkehr an den angegebenen Ports aus den festgelegten CIDR-Bereichen. Eine AWS::EC2::SecurityGroupRessource wird verwendet, um die Regeln zu spezifizieren. Für jede Regel müssen Sie ein Protokoll angeben. Für TCP müssen Sie einen Port oder Portbereich angeben. Wenn Sie weder eine Quellsicherheitsgruppe noch einen CIDR-Bereich angeben, wird der Stack zwar gestartet, die Regel aber nicht auf die Sicherheitsgruppe angewendet.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter EC2 Amazon-Sicherheitsgruppen für Ihre EC2 Amazon-Instances im EC2 Amazon-Benutzerhandbuch.

JSON

```
"ServerSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Allow connections from specified CIDR ranges",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "CidrIp": "192.0.2.0/24"
      },
      {
        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "CidrIp": "192.0.2.0/24"
      }
    ]
  }
}
```

YAML

```
ServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow connections from specified CIDR ranges
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 192.0.2.0/24
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 192.0.2.0/24
```

Erstellen von miteinander kommunizierenden Sicherheitsgruppen mit Regeln für eingehenden Datenverkehr

Das folgende Snippet verwendet die AWS::EC2::SecurityGroupRessource, um zwei EC2 Amazon-Sicherheitsgruppen zu erstellen, und`SGroup1`. `SGroup2` Eingangsregeln, die die Kommunikation zwischen den beiden Sicherheitsgruppen ermöglichen, werden mithilfe der AWS::EC2::SecurityGroup Ingress-Ressource erstellt. `SGroup1Ingress`richtet eine Eingangsregel ein`SGroup1`, die eingehenden TCP-Verkehr von der Quellsicherheitsgruppe auf Port 80 zulässt. `SGroup2` `SGroup2Ingress`richtet eine Eingangsregel ein`SGroup2`, die eingehenden TCP-Verkehr auf Port 80 von der Quellsicherheitsgruppe zulässt,. `SGroup1`

JSON

```
"SGroup1": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "EC2 instance access"
    }
},
"SGroup2": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "EC2 instance access"
    }
},
"SGroup1Ingress": {
    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
        "GroupName": {
            "Ref": "SGroup1"
        },
        "IpProtocol": "tcp",
        "ToPort": "80",
        "FromPort": "80",
        "SourceSecurityGroupName": {
            "Ref": "SGroup2"
        }
    }
},
"SGroup2Ingress": {
    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
```

```
        "GroupName": {
            "Ref": "SGroup2"
        },
        "IpProtocol": "tcp",
        "ToPort": "80",
        "FromPort": "80",
        "SourceSecurityGroupName": {
            "Ref": "SGroup1"
        }
    }
}
```

YAML

```
SGroup1:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: EC2 Instance access
SGroup2:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: EC2 Instance access
SGroup1Ingress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupName: !Ref SGroup1
    IpProtocol: tcp
    ToPort: 80
    FromPort: 80
    SourceSecurityGroupName: !Ref SGroup2
SGroup2Ingress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupName: !Ref SGroup2
    IpProtocol: tcp
    ToPort: 80
    FromPort: 80
    SourceSecurityGroupName: !Ref SGroup1
```

Erstellen eines Elastic Load Balancers mit einer Regel für eingehenden Datenverkehr für eine Sicherheitsgruppe

Die folgende Vorlage erstellt eine AWS::ElasticLoadBalancing::LoadBalancerRessource in der angegebenen Availability Zone. Die AWS::ElasticLoadBalancing::LoadBalancerRessource ist so konfiguriert, dass sie auf Port 80 auf HTTP-Verkehr wartet und Anfragen an Instanzen auch an Port 80 weiterleitet. Der Elastic Load Balancer übernimmt den Lastenausgleich des eingehenden HTTP-Datenverkehrs unter den Instances.

Darüber hinaus generiert diese Vorlage eine AWS::EC2::SecurityGroupRessource, die dem Load Balancer zugeordnet ist. Diese Sicherheitsgruppe wird mit einer einzigen Regel für eingehenden Datenverkehr erstellt, die als `ELB ingress group` bezeichnet wird. Sie erlaubt eingehenden TCP-Verkehr an Port 80. Die Quelle für diese Regel wird mithilfe der Funktion `Fn::GetAtt` definiert, um Attribute aus der Load-Balancer-Ressource abzurufen. `SourceSecurityGroupOwnerId` ruft mit `Fn::GetAtt` den `OwnerAlias` der Quellsicherheitsgruppe des Load Balancers ab. `SourceSecurityGroupName` ruft mit `Fn::Getatt` den `GroupName` der Quellsicherheitsgruppe des ELB ab.

Diese Konfiguration gewährleistet eine sichere Kommunikation zwischen dem ELB und den Instances.

Weitere Informationen zu Elastic Load Balancing finden Sie im Elastic Load Balancing-Entwicklerhandbuch.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "MyELB": {
            "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
            "Properties": {
                "AvailabilityZones": [
                    "aa-example-1a"
                ],
                "Listeners": [
                    {
                        "LoadBalancerPort": "80",
                        "InstancePort": "80",
                        "Protocol": "HTTP"
                    }
```

```
                ]
            }
        },
        "MyELBIngressGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "ELB ingress group",
                "SecurityGroupIngress": [
                    {
                        "IpProtocol": "tcp",
                        "FromPort": 80,
                        "ToPort": 80,
                        "SourceSecurityGroupOwnerId": {
                            "Fn::GetAtt": [
                                "MyELB",
                                "SourceSecurityGroup.OwnerAlias"
                            ]
                        },
                        "SourceSecurityGroupName": {
                            "Fn::GetAtt": [
                                "MyELB",
                                "SourceSecurityGroup.GroupName"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  MyELB:
    Type: 'AWS::ElasticLoadBalancing::LoadBalancer'
    Properties:
      AvailabilityZones:
        - aa-example-1a
      Listeners:
        - LoadBalancerPort: '80'
          InstancePort: '80'
```

```
        Protocol: HTTP
  MyELBIngressGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: ELB ingress group
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '80'
          ToPort: '80'
          SourceSecurityGroupOwnerId:
            Fn::GetAtt:
              - MyELB
              - SourceSecurityGroup.OwnerAlias
          SourceSecurityGroupName:
            Fn::GetAtt:
              - MyELB
              - SourceSecurityGroup.GroupName
```

Elastic IP-Adressen zuweisen und verknüpfen mit CloudFormation

Die folgenden Vorlagenausschnitte sind Beispiele für Elastic IP-Adressen (EIPs) in Amazon. EC2 Diese Beispiele behandeln die Zuweisung, Zuordnung und Verwaltung Ihrer EIPs Instances.

Beispielausschnitte

- Weisen Sie eine Elastic IP-Adresse zu und verknüpfen Sie sie mit einer Amazon-Instance EC2
- Ordnen Sie einer EC2 Amazon-Instance eine Elastic IP-Adresse zu, indem Sie die IP-Adresse angeben
- Ordnen Sie einer EC2 Amazon-Instance eine Elastic IP-Adresse zu, indem Sie die Zuweisungs-ID der IP-Adresse angeben

Weisen Sie eine Elastic IP-Adresse zu und verknüpfen Sie sie mit einer Amazon-Instance EC2

Das folgende Snippet weist eine Amazon EC2 Elastic IP (EIP) -Adresse zu und verknüpft sie mit einer EC2 Amazon-Instance, die eine Ressource verwendet. AWS::EC2::EIP Mithilfe von Bring Your Own IP Addresses (BYOIP) können Sie eine EIP-Adresse aus einem Adresspool zuweisen, der einem öffentlichen Adressbereich gehört, AWS oder aus einem IPv4 Adresspool, der aus einem öffentlichen Adressbereich erstellt wurde, den Sie AWS zur Verwendung mit Ihren AWS Ressourcen nutzen. In diesem Beispiel wird die EIP aus einem Adresspool zugewiesen, der Eigentümer ist. AWS

Weitere Informationen zu Elastic IP-Adressen finden Sie unter Elastic IP address im EC2 Amazon-Benutzerhandbuch.

JSON

```
"ElasticIP": {
    "Type": "AWS::EC2::EIP",
    "Properties": {
        "InstanceId": {
            "Ref": "Ec2Instance"
        }
    }
}
```

YAML

```
ElasticIP:
  Type: AWS::EC2::EIP
  Properties:
    InstanceId: !Ref EC2Instance
```

Ordnen Sie einer EC2 Amazon-Instance eine Elastic IP-Adresse zu, indem Sie die IP-Adresse angeben

Der folgende Ausschnitt verknüpft eine bestehende Amazon EC2 Elastic IP-Adresse mit einer EC2 Instance, die eine AWS::::EC2: EIPAssociation -Ressource verwendet. Dazu müssen Sie zunächst eine Elastic-IP-Adresse für die Verwendung in Ihrem Konto zuweisen. Eine Elastic-IP-Adresse kann mit einer einzelnen Instance verknüpft werden.

JSON

```
"IPAssoc": {
  "Type": "AWS::EC2::EIPAssociation",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    },
    "EIP": "192.0.2.0"
  }
}
```

YAML

```
IPAssoc:
  Type: AWS::EC2::EIPAssociation
  Properties:
    InstanceId: !Ref EC2Instance
    EIP: 192.0.2.0
```

Ordnen Sie einer EC2 Amazon-Instance eine Elastic IP-Adresse zu, indem Sie die Zuweisungs-ID der IP-Adresse angeben

Der folgende Ausschnitt verknüpft eine bestehende Elastic IP-Adresse mit einer EC2 Amazon-Instance, indem die Zuweisungs-ID mithilfe einer AWS:::EC2: EIPAssociation -Ressource angegeben wird. Eine Zuweisungs-ID wird einer Elastic IP-Adresse nach ihrer Zuweisung zugeordnet.

JSON

```
"IPAssoc": {
    "Type": "AWS::EC2::EIPAssociation",
    "Properties": {
        "InstanceId": {
            "Ref": "Ec2Instance"
        },
        "AllocationId": "eipalloc-1234567890abcdef0"
    }
}
```

YAML

```
IPAssoc:
  Type: AWS::EC2::EIPAssociation
  Properties:
    InstanceId: !Ref EC2Instance
    AllocationId: eipalloc-1234567890abcdef0
```

Konfigurieren Sie Amazon VPC-Ressourcen mit CloudFormation

Dieser Abschnitt enthält Beispiele für die Konfiguration von Amazon VPC-Ressourcen mithilfe von CloudFormation. VPCsermöglichen es Ihnen, darin ein virtuelles Netzwerk zu erstellen. Diese Auszüge zeigen AWS, wie Sie Aspekte von konfigurieren können, um Ihre VPCs Netzwerkanforderungen zu erfüllen.

Beispielausschnitte

- Reinen IPv6 ausgehenden Internetzugang in einer VPC aktivieren
- Elastic Network-Schnittstelle (ENI) Vorlagenausschnitte

Reinen IPv6 ausgehenden Internetzugang in einer VPC aktivieren

Ein auf ausgehenden Datenverkehr beschränktes Internet-Gateway erlaubt Instances in einer VPC den Zugriff auf das Internet und verhindert, dass Ressourcen aus dem Internet mit den Instances kommunizieren. Das folgende Snippet ermöglicht den Internetzugang IPv6 nur für ausgehenden Datenverkehr von einer VPC aus. Es erstellt eine VPC mit einem IPv4 Adressbereich, der eine `10.0.0/16` AWS:EC2: :VPC-Ressource verwendet. Eine Routentabelle ist dieser VPC-Ressource mithilfe einer AWS::EC2::RouteTableRessource zugeordnet. Über die Routing-Tabelle werden Routen für Instances in der VPC verwaltet. An AWS::EC2::EgressOnlyInternetGatewaywird verwendet, um ein Internet-Gateway nur für ausgehenden Datenverkehr zu erstellen, um die IPv6 Kommunikation für ausgehenden Datenverkehr von Instances innerhalb der VPC zu ermöglichen und gleichzeitig eingehenden Datenverkehr zu verhindern. Eine AWS::EC2::RouteRessource wird angegeben, um eine IPv6 Route in der Routentabelle zu erstellen, die den gesamten ausgehenden IPv6 Verkehr () `::/0` an das Internet-Gateway weiterleitet, das nur für ausgehenden Verkehr bestimmt ist.

Weitere Informationen zu Internet-Gateways nur für ausgehenden Datenverkehr finden Sie unter Aktivieren von ausgehendem IPv6 Datenverkehr mithilfe eines Internet-Gateways nur für ausgehenden Datenverkehr im Amazon VPC-Benutzerhandbuch.

JSON

```
"DefaultIpv6Route": {
    "Type": "AWS::EC2::Route",
    "Properties": {
        "DestinationIpv6CidrBlock": "::/0",
        "EgressOnlyInternetGatewayId": {
            "Ref": "EgressOnlyInternetGateway"
        },
        "RouteTableId": {
            "Ref": "RouteTable"
        }
    }
},
"EgressOnlyInternetGateway": {
```

```
    "Type": "AWS::EC2::EgressOnlyInternetGateway",
    "Properties": {
        "VpcId": {
            "Ref": "VPC"
        }
    }
},
"RouteTable": {
    "Type": "AWS::EC2::RouteTable",
    "Properties": {
        "VpcId": {
            "Ref": "VPC"
        }
    }
},
"VPC": {
    "Type": "AWS::EC2::VPC",
    "Properties": {
        "CidrBlock": "10.0.0.0/16"
    }
}
```

YAML

```
DefaultIpv6Route:
  Type: "AWS::EC2::Route"
  Properties:
    DestinationIpv6CidrBlock: "::/0"
    EgressOnlyInternetGatewayId:
      Ref: "EgressOnlyInternetGateway"
    RouteTableId:
      Ref: "RouteTable"
EgressOnlyInternetGateway:
  Type: "AWS::EC2::EgressOnlyInternetGateway"
  Properties:
    VpcId:
      Ref: "VPC"
RouteTable:
  Type: "AWS::EC2::RouteTable"
  Properties:
    VpcId:
      Ref: "VPC"
VPC:
```

```
  Type: "AWS::EC2::VPC"
  Properties:
    CidrBlock: "10.0.0.0/16"
```

Elastic Network-Schnittstelle (ENI) Vorlagenausschnitte

Erstellen Sie eine EC2 Amazon-Instance mit angehängten elastischen Netzwerkschnittstellen (ENIs)

Das folgende Beispiel-Snippet erstellt eine EC2 Amazon-Instance mithilfe einer AWS::EC2::InstanceRessource in der angegebenen Amazon-VPC und dem angegebenen Subnetz. Es verbindet zwei Netzwerkschnittstellen (ENIs) mit der Instance, ordnet den Instances über die angehängte ENIs Verbindung Elastic IP-Adressen zu und konfiguriert die Sicherheitsgruppe für SSH- und HTTP-Zugriff. Benutzerdaten werden der Instance im Rahmen der Startkonfiguration bereitgestellt, wenn die Instance erstellt wird. Die Benutzerdaten enthalten ein im `base64`-Format codiertes Skript, damit sie auf jeden Fall an die Instance übergeben werden. Beim Start der Instance wird das Skript automatisch als Teil des Bootstrapping-Prozesses ausgeführt. Es installiert `ec2-net-utils`, konfiguriert die Netzwerkschnittstellen und startet den HTTP-Service.

Um das passende Amazon Machine Image (AMI) für die ausgewählte Region zu ermitteln, wird im Ausschnitt eine `Fn::FindInMap`-Funktion verwendet, die Werte in einer `RegionMap`-Zuordnung nachschlägt. Diese Zuordnung muss in der größeren Vorlage definiert werden. Die beiden Netzwerkschnittstellen werden mithilfe von Ressourcen erstellt. AWS::EC2::NetworkInterface Elastische IP-Adressen werden mithilfe von AWS::EC2::EIPRessourcen spezifiziert, die der `vpc` Domain zugewiesen sind. Diese elastischen IP-Adressen sind den Netzwerkschnittstellen zugeordnet, die AWS::EC2::EIPAssociationRessourcen verwenden.

Im Abschnitt `Outputs` werden Werte oder Ressourcen definiert, auf die Sie nach Erstellung des Stacks zugreifen möchten. In diesem Snippet lautet die definierte Ausgabe`InstancePublicIp`, die die öffentliche IP-Adresse der vom Stack erstellten EC2 Instanz darstellt. Sie können diese Ausgabe auf der Registerkarte Ausgabe auf der AWS CloudFormation Konsole oder mit dem Befehl describe-stacks abrufen.

Weitere Informationen über Elastic-Network-Schnittstellen finden Sie unter Elastic-Network-Schnittstellen.

JSON

```
"Resources": {
    "ControlPortAddress": {
        "Type": "AWS::EC2::EIP",
```

```
        "Properties": {
            "Domain": "vpc"
        }
    },
    "AssociateControlPort": {
        "Type": "AWS::EC2::EIPAssociation",
        "Properties": {
            "AllocationId": {
                "Fn::GetAtt": [
                    "ControlPortAddress",
                    "AllocationId"
                ]
            },
            "NetworkInterfaceId": {
                "Ref": "controlXface"
            }
        }
    },
    "WebPortAddress": {
        "Type": "AWS::EC2::EIP",
        "Properties": {
            "Domain": "vpc"
        }
    },
    "AssociateWebPort": {
        "Type": "AWS::EC2::EIPAssociation",
        "Properties": {
            "AllocationId": {
                "Fn::GetAtt": [
                    "WebPortAddress",
                    "AllocationId"
                ]
            },
            "NetworkInterfaceId": {
                "Ref": "webXface"
            }
        }
    },
    "SSHSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "VpcId": {
                "Ref": "VpcId"
            },
```

```
            "GroupDescription": "Enable SSH access via port 22",
            "SecurityGroupIngress": [
                {
                    "CidrIp": "0.0.0.0/0",
                    "FromPort": 22,
                    "IpProtocol": "tcp",
                    "ToPort": 22
                }
            ]
        }
    },
    "WebSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "VpcId": {
                "Ref": "VpcId"
            },
            "GroupDescription": "Enable HTTP access via user-defined port",
            "SecurityGroupIngress": [
                {
                    "CidrIp": "0.0.0.0/0",
                    "FromPort": 80,
                    "IpProtocol": "tcp",
                    "ToPort": 80
                }
            ]
        }
    },
    "controlXface": {
        "Type": "AWS::EC2::NetworkInterface",
        "Properties": {
            "SubnetId": {
                "Ref": "SubnetId"
            },
            "Description": "Interface for controlling traffic such as SSH",
            "GroupSet": [
                {
                    "Fn::GetAtt": [
                        "SSHSecurityGroup",
                        "GroupId"
                    ]
                }
            ],
            "SourceDestCheck": true,
```

```
            "Tags": [
                {
                    "Key": "Network",
                    "Value": "Control"
                }
            ]
        }
    },
    "webXface": {
        "Type": "AWS::EC2::NetworkInterface",
        "Properties": {
            "SubnetId": {
                "Ref": "SubnetId"
            },
            "Description": "Interface for web traffic",
            "GroupSet": [
                {
                    "Fn::GetAtt": [
                        "WebSecurityGroup",
                        "GroupId"
                    ]
                }
            ],
            "SourceDestCheck": true,
            "Tags": [
                {
                    "Key": "Network",
                    "Value": "Web"
                }
            ]
        }
    },
    "Ec2Instance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
            "ImageId": {
                "Fn::FindInMap": [
                    "RegionMap",
                    {
                        "Ref": "AWS::Region"
                    },
                    "AMI"
                ]
            },
```

```
            "KeyName": {
                "Ref": "KeyName"
            },
            "NetworkInterfaces": [
                {
                    "NetworkInterfaceId": {
                        "Ref": "controlXface"
                    },
                    "DeviceIndex": "0"
                },
                {
                    "NetworkInterfaceId": {
                        "Ref": "webXface"
                    },
                    "DeviceIndex": "1"
                }
            ],
            "Tags": [
                {
                    "Key": "Role",
                    "Value": "Test Instance"
                }
            ],
            "UserData": {
                "Fn::Base64": {
                    "Fn::Sub": "#!/bin/bash -xe\nyum install ec2-net-utils -y\nec2ifup
 eth1\nservice httpd start\n"
                }
            }
        }
    }
},
"Outputs": {
    "InstancePublicIp": {
        "Description": "Public IP Address of the EC2 Instance",
        "Value": {
            "Fn::GetAtt": [
                "Ec2Instance",
                "PublicIp"
            ]
        }
    }
}
```

YAML

```
Resources:
  ControlPortAddress:
    Type: 'AWS::EC2::EIP'
    Properties:
      Domain: vpc
  AssociateControlPort:
    Type: 'AWS::EC2::EIPAssociation'
    Properties:
      AllocationId:
        Fn::GetAtt:
          - ControlPortAddress
          - AllocationId
      NetworkInterfaceId:
        Ref: controlXface
  WebPortAddress:
    Type: 'AWS::EC2::EIP'
    Properties:
      Domain: vpc
  AssociateWebPort:
    Type: 'AWS::EC2::EIPAssociation'
    Properties:
      AllocationId:
        Fn::GetAtt:
          - WebPortAddress
          - AllocationId
      NetworkInterfaceId:
        Ref: webXface
  SSHSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      VpcId:
        Ref: VpcId
      GroupDescription: Enable SSH access via port 22
      SecurityGroupIngress:
        - CidrIp: 0.0.0.0/0
          FromPort: 22
          IpProtocol: tcp
          ToPort: 22
  WebSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      VpcId:
```

```
        Ref: VpcId
      GroupDescription: Enable HTTP access via user-defined port
      SecurityGroupIngress:
        - CidrIp: 0.0.0.0/0
          FromPort: 80
          IpProtocol: tcp
          ToPort: 80
  controlXface:
    Type: 'AWS::EC2::NetworkInterface'
    Properties:
      SubnetId:
        Ref: SubnetId
      Description: Interface for controlling traffic such as SSH
      GroupSet:
        - Fn::GetAtt:
            - SSHSecurityGroup
            - GroupId
      SourceDestCheck: true
      Tags:
        - Key: Network
          Value: Control
  webXface:
    Type: 'AWS::EC2::NetworkInterface'
    Properties:
      SubnetId:
        Ref: SubnetId
      Description: Interface for web traffic
      GroupSet:
        - Fn::GetAtt:
            - WebSecurityGroup
            - GroupId
      SourceDestCheck: true
      Tags:
        - Key: Network
          Value: Web
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId:
        Fn::FindInMap:
          - RegionMap
          - Ref: AWS::Region
          - AMI
      KeyName:
```

```
        Ref: KeyName
      NetworkInterfaces:
        - NetworkInterfaceId:
            Ref: controlXface
          DeviceIndex: "0"
        - NetworkInterfaceId:
            Ref: webXface
          DeviceIndex: "1"
      Tags:
        - Key: Role
          Value: Test Instance
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash -xe
          yum install ec2-net-utils -y
          ec2ifup eth1
          service httpd start
Outputs:
  InstancePublicIp:
    Description: Public IP Address of the EC2 Instance
    Value:
      Fn::GetAtt:
        - Ec2Instance
        - PublicIp
```

Beispielvorlagen für Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) ist ein Container-Management-Service, der das Ausführen, Stoppen und Verwalten von Docker-Containern auf einem Cluster von Amazon Elastic Compute Cloud (Amazon EC2) -Instances vereinfacht.

Erstellen Sie einen Cluster mit dem AL2 023 Amazon ECS-Optimized-AMI

Definieren Sie einen Cluster, der einen Kapazitätsanbieter verwendet, der AL2 023 Instances auf Amazon EC2 startet.

> **Important**
>
> Das neueste AMI IDs finden Sie unter Amazon ECS-Optimized AMI im Amazon Elastic Container Service Developer Guide.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "EC2 ECS cluster that starts out empty, with no EC2 instances yet. An
 ECS capacity provider automatically launches more EC2 instances as required on the fly
 when you request ECS to launch services or standalone tasks.",
  "Parameters": {
      "InstanceType": {
          "Type": "String",
          "Description": "EC2 instance type",
          "Default": "t2.medium",
          "AllowedValues": [
              "t1.micro",
              "t2.2xlarge",
              "t2.large",
              "t2.medium",
              "t2.micro",
              "t2.nano",
              "t2.small",
              "t2.xlarge",
              "t3.2xlarge",
              "t3.large",
              "t3.medium",
              "t3.micro",
              "t3.nano",
              "t3.small",
              "t3.xlarge"
          ]
      },
      "DesiredCapacity": {
          "Type": "Number",
          "Default": "0",
          "Description": "Number of EC2 instances to launch in your ECS cluster."
      },
      "MaxSize": {
          "Type": "Number",
          "Default": "100",
          "Description": "Maximum number of EC2 instances that can be launched in your
 ECS cluster."
      },
      "ECSAMI": {
          "Description": "The Amazon Machine Image ID used for the cluster",
          "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
```

```
          "Default": "/aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/
image_id"
      },
      "VpcId": {
          "Type": "AWS::EC2::VPC::Id",
          "Description": "VPC ID where the ECS cluster is launched",
          "Default": "vpc-1234567890abcdef0"
      },
      "SubnetIds": {
          "Type": "List<AWS::EC2::Subnet::Id>",
          "Description": "List of subnet IDs where the EC2 instances will be launched",
          "Default": "subnet-021345abcdef67890"
      }
  },
  "Resources": {
      "ECSCluster": {
          "Type": "AWS::ECS::Cluster",
          "Properties": {
              "ClusterSettings": [
                  {
                      "Name": "containerInsights",
                      "Value": "enabled"
                  }
              ]
          }
      },
      "ECSAutoScalingGroup": {
          "Type": "AWS::AutoScaling::AutoScalingGroup",
          "DependsOn": [
              "ECSCluster",
              "EC2Role"
          ],
          "Properties": {
              "VPCZoneIdentifier": {
                  "Ref": "SubnetIds"
              },
              "LaunchTemplate": {
                  "LaunchTemplateId": {
                      "Ref": "ContainerInstances"
                  },
                  "Version": {
                      "Fn::GetAtt": [
                          "ContainerInstances",
                          "LatestVersionNumber"
```

```
                    ]
                }
            },
            "MinSize": 0,
            "MaxSize": {
                "Ref": "MaxSize"
            },
            "DesiredCapacity": {
                "Ref": "DesiredCapacity"
            },
            "NewInstancesProtectedFromScaleIn": true
        },
        "UpdatePolicy": {
            "AutoScalingReplacingUpdate": {
                "WillReplace": "true"
            }
        }
    },
    "ContainerInstances": {
        "Type": "AWS::EC2::LaunchTemplate",
        "Properties": {
            "LaunchTemplateName": "asg-launch-template",
            "LaunchTemplateData": {
                "ImageId": {
                    "Ref": "ECSAMI"
                },
                "InstanceType": {
                    "Ref": "InstanceType"
                },
                "IamInstanceProfile": {
                    "Name": {
                        "Ref": "EC2InstanceProfile"
                    }
                },
                "SecurityGroupIds": [
                    {
                        "Ref": "ContainerHostSecurityGroup"
                    }
                ],
                "UserData": {
                    "Fn::Base64": {
                        "Fn::Sub": "#!/bin/bash -xe\n echo ECS_CLUSTER=${ECSCluster}
>> /etc/ecs/ecs.config\n yum install -y aws-cfn-bootstrap\n /opt/aws/bin/cfn-init -
```

```
v --stack ${AWS::StackId} --resource ContainerInstances --configsets full_install --
region ${AWS::Region} &\n"
                        }
                    },
                    "MetadataOptions": {
                        "HttpEndpoint": "enabled",
                        "HttpTokens": "required"
                    }
                }
            }
        },
        "EC2InstanceProfile": {
            "Type": "AWS::IAM::InstanceProfile",
            "Properties": {
                "Path": "/",
                "Roles": [
                    {
                        "Ref": "EC2Role"
                    }
                ]
            }
        },
        "CapacityProvider": {
            "Type": "AWS::ECS::CapacityProvider",
            "Properties": {
                "AutoScalingGroupProvider": {
                    "AutoScalingGroupArn": {
                        "Ref": "ECSAutoScalingGroup"
                    },
                    "ManagedScaling": {
                        "InstanceWarmupPeriod": 60,
                        "MinimumScalingStepSize": 1,
                        "MaximumScalingStepSize": 100,
                        "Status": "ENABLED",
                        "TargetCapacity": 100
                    },
                    "ManagedTerminationProtection": "ENABLED"
                }
            }
        },
        "CapacityProviderAssociation": {
            "Type": "AWS::ECS::ClusterCapacityProviderAssociations",
            "Properties": {
                "CapacityProviders": [
```

```
                {
                    "Ref": "CapacityProvider"
                }
            ],
            "Cluster": {
                "Ref": "ECSCluster"
            },
            "DefaultCapacityProviderStrategy": [
                {
                    "Base": 0,
                    "CapacityProvider": {
                        "Ref": "CapacityProvider"
                    },
                    "Weight": 1
                }
            ]
        }
    },
    "ContainerHostSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Access to the EC2 hosts that run containers",
            "VpcId": {
                "Ref": "VpcId"
            }
        }
    },
    "EC2Role": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": [
                                "ec2.amazonaws.com"
                            ]
                        },
                        "Action": [
                            "sts:AssumeRole"
                        ]
                    }
                ]
```

```
              },
              "Path": "/",
              "ManagedPolicyArns": [
                  "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
                  "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
              ]
          }
      },
      "ECSTaskExecutionRole": {
          "Type": "AWS::IAM::Role",
          "Properties": {
              "AssumeRolePolicyDocument": {
                  "Statement": [
                      {
                          "Effect": "Allow",
                          "Principal": {
                              "Service": [
                                  "ecs-tasks.amazonaws.com"
                              ]
                          },
                          "Action": [
                              "sts:AssumeRole"
                          ],
                          "Condition": {
                              "ArnLike": {
                                  "aws:SourceArn": {
                                      "Fn::Sub": "arn:${AWS::Partition}:ecs:
${AWS::Region}:${AWS::AccountId}:*"
                                  }
                              },
                              "StringEquals": {
                                  "aws:SourceAccount": {
                                        "Fn::Sub": "${AWS::AccountId}"
                                    }
                              }
                          }
                      }
                  ]
              },
              "Path": "/",
              "ManagedPolicyArns": [
                  "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy"
```

```
                ]
            }
        }
    },
    "Outputs": {
        "ClusterName": {
            "Description": "The ECS cluster into which to launch resources",
            "Value": "ECSCluster"
        },
        "ECSTaskExecutionRole": {
            "Description": "The role used to start up a task",
            "Value": "ECSTaskExecutionRole"
        },
        "CapacityProvider": {
            "Description": "The cluster capacity provider that the service should use to
 request capacity when it wants to start up a task",
            "Value": "CapacityProvider"
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: EC2 ECS cluster that starts out empty, with no EC2 instances yet.
  An ECS capacity provider automatically launches more EC2 instances as required
  on the fly when you request ECS to launch services or standalone tasks.
Parameters:
  InstanceType:
    Type: String
    Description: EC2 instance type
    Default: "t2.medium"
    AllowedValues:
      - t1.micro
      - t2.2xlarge
      - t2.large
      - t2.medium
      - t2.micro
      - t2.nano
      - t2.small
      - t2.xlarge
      - t3.2xlarge
```

```
      - t3.large
      - t3.medium
      - t3.micro
      - t3.nano
      - t3.small
      - t3.xlarge
  DesiredCapacity:
    Type: Number
    Default: "0"
    Description: Number of EC2 instances to launch in your ECS cluster.
  MaxSize:
    Type: Number
    Default: "100"
    Description: Maximum number of EC2 instances that can be launched in your ECS
 cluster.
  ECSAMI:
    Description: The Amazon Machine Image ID used for the cluster
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/image_id
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VPC ID where the ECS cluster is launched
    Default: vpc-1234567890abcdef0
  SubnetIds:
    Type: List<AWS::EC2::Subnet::Id>
    Description: List of subnet IDs where the EC2 instances will be launched
    Default: "subnet-021345abcdef67890"
Resources:
# This is authorizes ECS to manage resources on your
  # account on your behalf. This role is likely already created on your account
  # ECSRole:
  #  Type: AWS::IAM::ServiceLinkedRole
  #  Properties:
  #    AWSServiceName: 'ecs.amazonaws.com'

   # ECS Resources
  ECSCluster:
    Type: AWS::ECS::Cluster
    Properties:
      ClusterSettings:
        - Name: containerInsights
          Value: enabled

  # Autoscaling group. This launches the actual EC2 instances that will register
```

```
  # themselves as members of the cluster, and run the docker containers.
  ECSAutoScalingGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    DependsOn:
      # This is to ensure that the ASG gets deleted first before these
    # resources, when it comes to stack teardown.
      - ECSCluster
      - EC2Role
    Properties:
      VPCZoneIdentifier:
        Ref: SubnetIds
      LaunchTemplate:
        LaunchTemplateId: !Ref ContainerInstances
        Version: !GetAtt ContainerInstances.LatestVersionNumber
      MinSize: 0
      MaxSize:
        Ref: MaxSize
      DesiredCapacity:
        Ref: DesiredCapacity
      NewInstancesProtectedFromScaleIn: true
    UpdatePolicy:
      AutoScalingReplacingUpdate:
        WillReplace: "true"
  # The config for each instance that is added to the cluster
  ContainerInstances:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: "asg-launch-template"
      LaunchTemplateData:
        ImageId:
          Ref: ECSAMI
        InstanceType:
          Ref: InstanceType
        IamInstanceProfile:
          Name: !Ref EC2InstanceProfile
        SecurityGroupIds:
          - !Ref ContainerHostSecurityGroup
        # This injected configuration file is how the EC2 instance
      # knows which ECS cluster on your AWS account it should be joining
        UserData:
          Fn::Base64: !Sub |
           #!/bin/bash -xe
            echo ECS_CLUSTER=${ECSCluster} >> /etc/ecs/ecs.config
            yum install -y aws-cfn-bootstrap
```

```
            /opt/aws/bin/cfn-init -v --stack ${AWS::StackId} --resource
 ContainerInstances --configsets full_install --region ${AWS::Region} &
         # Disable IMDSv1, and require IMDSv2
        MetadataOptions:
          HttpEndpoint: enabled
          HttpTokens: required
  EC2InstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: /
      Roles:
      - !Ref EC2Role
  # Create an ECS capacity provider to attach the ASG to the ECS cluster
  # so that it autoscales as we launch more containers
  CapacityProvider:
    Type: AWS::ECS::CapacityProvider
    Properties:
      AutoScalingGroupProvider:
        AutoScalingGroupArn: !Ref ECSAutoScalingGroup
        ManagedScaling:
          InstanceWarmupPeriod: 60
          MinimumScalingStepSize: 1
          MaximumScalingStepSize: 100
          Status: ENABLED
          # Percentage of cluster reservation to try to maintain
          TargetCapacity: 100
        ManagedTerminationProtection: ENABLED
   # Create a cluster capacity provider assocation so that the cluster
  # will use the capacity provider
  CapacityProviderAssociation:
    Type: AWS::ECS::ClusterCapacityProviderAssociations
    Properties:
      CapacityProviders:
        - !Ref CapacityProvider
      Cluster: !Ref ECSCluster
      DefaultCapacityProviderStrategy:
        - Base: 0
          CapacityProvider: !Ref CapacityProvider
          Weight: 1
  # A security group for the EC2 hosts that will run the containers.
  # This can be used to limit incoming traffic to or outgoing traffic
  # from the container's host EC2 instance.
  ContainerHostSecurityGroup:
    Type: AWS::EC2::SecurityGroup
```

```
    Properties:
      GroupDescription: Access to the EC2 hosts that run containers
      VpcId:
        Ref: VpcId
  # Role for the EC2 hosts. This allows the ECS agent on the EC2 hosts
  # to communciate with the ECS control plane, as well as download the docker
  # images from ECR to run on your host.
  EC2Role:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: /
      ManagedPolicyArns:
      # See reference: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/
security-iam-awsmanpol.html#security-iam-awsmanpol-AmazonEC2ContainerServiceforEC2Role
        - arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
      # This managed policy allows us to connect to the instance using SSM
        - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
  # This is a role which is used within Fargate to allow the Fargate agent
  # to download images, and upload logs.
  ECSTaskExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ecs-tasks.amazonaws.com
            Action:
              - sts:AssumeRole
            Condition:
              ArnLike:
                aws:SourceArn: !Sub arn:${AWS::Partition}:ecs:${AWS::Region}:
${AWS::AccountId}:*
              StringEquals:
                aws:SourceAccount: !Sub ${AWS::AccountId}
```

```
      Path: /
      # This role enables all features of ECS. See reference:
    # https://docs.aws.amazon.com/AmazonECS/latest/developerguide/security-iam-
awsmanpol.html#security-iam-awsmanpol-AmazonECSTaskExecutionRolePolicy
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
Outputs:
  ClusterName:
    Description: The ECS cluster into which to launch resources
    Value: ECSCluster
  ECSTaskExecutionRole:
    Description: The role used to start up a task
    Value: ECSTaskExecutionRole
  CapacityProvider:
    Description: The cluster capacity provider that the service should use to
      request capacity when it wants to start up a task
    Value: CapacityProvider
```

Stellen Sie einen Service bereit

Die folgende Vorlage definiert einen Dienst, der den Kapazitätsanbieter verwendet, um AL2 023-Kapazität zur Ausführung anzufordern. Container werden auf den AL2 023-Instances gestartet, sobald sie online gehen:

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "An example service that deploys in AWS VPC networking mode on EC2
 capacity. Service uses a capacity provider to request EC2 instances to run on. Service
 runs with networking in private subnets, but still accessible to the internet via a
 load balancer hosted in public subnets.",
  "Parameters": {
      "VpcId": {
          "Type": "String",
          "Description": "The VPC that the service is running inside of"
      },
      "PublicSubnetIds": {
          "Type": "List<AWS::EC2::Subnet::Id>",
          "Description": "List of public subnet ID's to put the load balancer in"
      },
      "PrivateSubnetIds": {
          "Type": "List<AWS::EC2::Subnet::Id>",
```

```
            "Description": "List of private subnet ID's that the AWS VPC tasks are in"
        },
        "ClusterName": {
            "Type": "String",
            "Description": "The name of the ECS cluster into which to launch capacity."
        },
        "ECSTaskExecutionRole": {
            "Type": "String",
            "Description": "The role used to start up an ECS task"
        },
        "CapacityProvider": {
            "Type": "String",
            "Description": "The cluster capacity provider that the service should use to
 request capacity when it wants to start up a task"
        },
        "ServiceName": {
            "Type": "String",
            "Default": "web",
            "Description": "A name for the service"
        },
        "ImageUrl": {
            "Type": "String",
            "Default": "public.ecr.aws/docker/library/nginx:latest",
            "Description": "The url of a docker image that contains the application
 process that will handle the traffic for this service"
        },
        "ContainerCpu": {
            "Type": "Number",
            "Default": 256,
            "Description": "How much CPU to give the container. 1024 is 1 CPU"
        },
        "ContainerMemory": {
            "Type": "Number",
            "Default": 512,
            "Description": "How much memory in megabytes to give the container"
        },
        "ContainerPort": {
            "Type": "Number",
            "Default": 80,
            "Description": "What port that the application expects traffic on"
        },
        "DesiredCount": {
            "Type": "Number",
            "Default": 2,
```

```
            "Description": "How many copies of the service task to run"
        }
    },
    "Resources": {
        "TaskDefinition": {
            "Type": "AWS::ECS::TaskDefinition",
            "Properties": {
                "Family": {
                    "Ref": "ServiceName"
                },
                "Cpu": {
                    "Ref": "ContainerCpu"
                },
                "Memory": {
                    "Ref": "ContainerMemory"
                },
                "NetworkMode": "awsvpc",
                "RequiresCompatibilities": [
                    "EC2"
                ],
                "ExecutionRoleArn": {
                    "Ref": "ECSTaskExecutionRole"
                },
                "ContainerDefinitions": [
                    {
                        "Name": {
                            "Ref": "ServiceName"
                        },
                        "Cpu": {
                            "Ref": "ContainerCpu"
                        },
                        "Memory": {
                            "Ref": "ContainerMemory"
                        },
                        "Image": {
                            "Ref": "ImageUrl"
                        },
                        "PortMappings": [
                            {
                                "ContainerPort": {
                                    "Ref": "ContainerPort"
                                },
                                "HostPort": {
                                    "Ref": "ContainerPort"
```

```
                            }
                        }
                    ],
                    "LogConfiguration": {
                        "LogDriver": "awslogs",
                        "Options": {
                            "mode": "non-blocking",
                            "max-buffer-size": "25m",
                            "awslogs-group": {
                                "Ref": "LogGroup"
                            },
                            "awslogs-region": {
                                "Ref": "AWS::Region"
                            },
                            "awslogs-stream-prefix": {
                                "Ref": "ServiceName"
                            }
                        }
                    }
                }
            ]
        }
    },
    "Service": {
        "Type": "AWS::ECS::Service",
        "DependsOn": "PublicLoadBalancerListener",
        "Properties": {
            "ServiceName": {
                "Ref": "ServiceName"
            },
            "Cluster": {
                "Ref": "ClusterName"
            },
            "PlacementStrategies": [
                {
                    "Field": "attribute:ecs.availability-zone",
                    "Type": "spread"
                },
                {
                    "Field": "cpu",
                    "Type": "binpack"
                }
            ],
            "CapacityProviderStrategy": [
```

```
                {
                    "Base": 0,
                    "CapacityProvider": {
                        "Ref": "CapacityProvider"
                    },
                    "Weight": 1
                }
            ],
            "NetworkConfiguration": {
                "AwsvpcConfiguration": {
                    "SecurityGroups": [
                        {
                            "Ref": "ServiceSecurityGroup"
                        }
                    ],
                    "Subnets": {
                        "Ref": "PrivateSubnetIds"
                    }
                }
            },
            "DeploymentConfiguration": {
                "MaximumPercent": 200,
                "MinimumHealthyPercent": 75
            },
            "DesiredCount": {
                "Ref": "DesiredCount"
            },
            "TaskDefinition": {
                "Ref": "TaskDefinition"
            },
            "LoadBalancers": [
                {
                    "ContainerName": {
                        "Ref": "ServiceName"
                    },
                    "ContainerPort": {
                        "Ref": "ContainerPort"
                    },
                    "TargetGroupArn": {
                        "Ref": "ServiceTargetGroup"
                    }
                }
            ]
        }
```

```
    },
    "ServiceSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Security group for service",
            "VpcId": {
                "Ref": "VpcId"
            }
        }
    },
    "ServiceTargetGroup": {
        "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
        "Properties": {
            "HealthCheckIntervalSeconds": 6,
            "HealthCheckPath": "/",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 2,
            "TargetType": "ip",
            "Port": {
                "Ref": "ContainerPort"
            },
            "Protocol": "HTTP",
            "UnhealthyThresholdCount": 10,
            "VpcId": {
                "Ref": "VpcId"
            },
            "TargetGroupAttributes": [
                {
                    "Key": "deregistration_delay.timeout_seconds",
                    "Value": 0
                }
            ]
        }
    },
    "PublicLoadBalancerSG": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Access to the public facing load balancer",
            "VpcId": {
                "Ref": "VpcId"
            },
            "SecurityGroupIngress": [
                {
```

```
                    "CidrIp": "0.0.0.0/0",
                    "IpProtocol": -1
                }
            ]
        }
    },
    "PublicLoadBalancer": {
        "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
        "Properties": {
            "Scheme": "internet-facing",
            "LoadBalancerAttributes": [
                {
                    "Key": "idle_timeout.timeout_seconds",
                    "Value": "30"
                }
            ],
            "Subnets": {
                "Ref": "PublicSubnetIds"
            },
            "SecurityGroups": [
                {
                    "Ref": "PublicLoadBalancerSG"
                }
            ]
        }
    },
    "PublicLoadBalancerListener": {
        "Type": "AWS::ElasticLoadBalancingV2::Listener",
        "Properties": {
            "DefaultActions": [
                {
                    "Type": "forward",
                    "ForwardConfig": {
                        "TargetGroups": [
                            {
                                "TargetGroupArn": {
                                    "Ref": "ServiceTargetGroup"
                                },
                                "Weight": 100
                            }
                        ]
                    }
                }
            ],
```

```
                "LoadBalancerArn": {
                    "Ref": "PublicLoadBalancer"
                },
                "Port": 80,
                "Protocol": "HTTP"
            }
        },
        "ServiceIngressfromLoadBalancer": {
            "Type": "AWS::EC2::SecurityGroupIngress",
            "Properties": {
                "Description": "Ingress from the public ALB",
                "GroupId": {
                    "Ref": "ServiceSecurityGroup"
                },
                "IpProtocol": -1,
                "SourceSecurityGroupId": {
                    "Ref": "PublicLoadBalancerSG"
                }
            }
        },
        "LogGroup": {
            "Type": "AWS::Logs::LogGroup"
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Description: >-
  An example service that deploys in AWS VPC networking mode on EC2 capacity.
  Service uses a capacity provider to request EC2 instances to run on. Service
  runs with networking in private subnets, but still accessible to the internet
  via a load balancer hosted in public subnets.
Parameters:
  VpcId:
    Type: String
    Description: The VPC that the service is running inside of
  PublicSubnetIds:
    Type: 'List<AWS::EC2::Subnet::Id>'
    Description: List of public subnet ID's to put the load balancer in
  PrivateSubnetIds:
    Type: 'List<AWS::EC2::Subnet::Id>'
```

```
    Description: List of private subnet ID's that the AWS VPC tasks are in
  ClusterName:
    Type: String
    Description: The name of the ECS cluster into which to launch capacity.
  ECSTaskExecutionRole:
    Type: String
    Description: The role used to start up an ECS task
  CapacityProvider:
    Type: String
    Description: >-
      The cluster capacity provider that the service should use to request
      capacity when it wants to start up a task
  ServiceName:
    Type: String
    Default: web
    Description: A name for the service
  ImageUrl:
    Type: String
    Default: 'public.ecr.aws/docker/library/nginx:latest'
    Description: >-
      The url of a docker image that contains the application process that will
      handle the traffic for this service
  ContainerCpu:
    Type: Number
    Default: 256
    Description: How much CPU to give the container. 1024 is 1 CPU
  ContainerMemory:
    Type: Number
    Default: 512
    Description: How much memory in megabytes to give the container
  ContainerPort:
    Type: Number
    Default: 80
    Description: What port that the application expects traffic on
  DesiredCount:
    Type: Number
    Default: 2
    Description: How many copies of the service task to run
Resources:
  TaskDefinition:
    Type: 'AWS::ECS::TaskDefinition'
    Properties:
      Family: !Ref ServiceName
      Cpu: !Ref ContainerCpu
```

```
      Memory: !Ref ContainerMemory
      NetworkMode: awsvpc
      RequiresCompatibilities:
        - EC2
      ExecutionRoleArn: !Ref ECSTaskExecutionRole
      ContainerDefinitions:
        - Name: !Ref ServiceName
          Cpu: !Ref ContainerCpu
          Memory: !Ref ContainerMemory
          Image: !Ref ImageUrl
          PortMappings:
            - ContainerPort: !Ref ContainerPort
              HostPort: !Ref ContainerPort
          LogConfiguration:
            LogDriver: awslogs
            Options:
              mode: non-blocking
              max-buffer-size: 25m
              awslogs-group: !Ref LogGroup
              awslogs-region: !Ref AWS::Region
              awslogs-stream-prefix: !Ref ServiceName
  Service:
    Type: AWS::ECS::Service
    DependsOn: PublicLoadBalancerListener
    Properties:
      ServiceName: !Ref ServiceName
      Cluster: !Ref ClusterName
      PlacementStrategies:
        - Field: 'attribute:ecs.availability-zone'
          Type: spread
        - Field: cpu
          Type: binpack
      CapacityProviderStrategy:
        - Base: 0
          CapacityProvider: !Ref CapacityProvider
          Weight: 1
      NetworkConfiguration:
        AwsvpcConfiguration:
          SecurityGroups:
            - !Ref ServiceSecurityGroup
          Subnets: !Ref PrivateSubnetIds
      DeploymentConfiguration:
        MaximumPercent: 200
        MinimumHealthyPercent: 75
```

```
      DesiredCount: !Ref DesiredCount
      TaskDefinition: !Ref TaskDefinition
      LoadBalancers:
        - ContainerName: !Ref ServiceName
          ContainerPort: !Ref ContainerPort
          TargetGroupArn: !Ref ServiceTargetGroup
  ServiceSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Security group for service
      VpcId: !Ref VpcId
  ServiceTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      HealthCheckIntervalSeconds: 6
      HealthCheckPath: /
      HealthCheckProtocol: HTTP
      HealthCheckTimeoutSeconds: 5
      HealthyThresholdCount: 2
      TargetType: ip
      Port: !Ref ContainerPort
      Protocol: HTTP
      UnhealthyThresholdCount: 10
      VpcId: !Ref VpcId
      TargetGroupAttributes:
        - Key: deregistration_delay.timeout_seconds
          Value: 0
  PublicLoadBalancerSG:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Access to the public facing load balancer
      VpcId: !Ref VpcId
      SecurityGroupIngress:
        - CidrIp: 0.0.0.0/0
          IpProtocol: -1
  PublicLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Scheme: internet-facing
      LoadBalancerAttributes:
        - Key: idle_timeout.timeout_seconds
          Value: '30'
      Subnets: !Ref PublicSubnetIds
      SecurityGroups:
```

```
      - !Ref PublicLoadBalancerSG
  PublicLoadBalancerListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      DefaultActions:
        - Type: forward
          ForwardConfig:
            TargetGroups:
              - TargetGroupArn: !Ref ServiceTargetGroup
                Weight: 100
      LoadBalancerArn: !Ref PublicLoadBalancer
      Port: 80
      Protocol: HTTP
  ServiceIngressfromLoadBalancer:
    Type: 'AWS::EC2::SecurityGroupIngress'
    Properties:
      Description: Ingress from the public ALB
      GroupId: !Ref ServiceSecurityGroup
      IpProtocol: -1
      SourceSecurityGroupId: !Ref PublicLoadBalancerSG
  LogGroup:
    Type: 'AWS::Logs::LogGroup'
```

Amazon Elastic File System-Beispielvorlage

Amazon Elastic File System (Amazon EFS) ist ein Dateispeicherservice für Amazon Elastic Compute Cloud (Amazon EC2) -Instances. Mit Amazon EFS haben ihre Anwendungen Speicher, wenn sie ihn benötigen, da die Speicherkapazität automatisch wächst oder sinkt, wenn Sie Dateien hinzufügen oder entfernen.

Die folgende Beispielvorlage stellt EC2 Instances (in einer Auto Scaling Scaling-Gruppe) bereit, die einem Amazon EFS-Dateisystem zugeordnet sind. Um die Instances mit dem Dateisystem zu verknüpfen, führen die Instances das cfn-init Hilfsskript aus, das den `nfs-utils` runterlädt und installiert, yum package erstellt ein neues Verzeichnis und verwendet den DNS-Namen des Dateisystems um das Dateisystem am Verzeichnis zu mounten. Der DNS-Name des Dateisystems wird in die IP-Adresse eines Mount-Ziels in der Availability Zone der EC2 Amazon-Instance aufgelöst. Weitere Informationen über den DNS-Namenstruktur finden sie unter Mounting von Dateisystemen im Amazon Elastic File System-Benutzerhandbuch.

Um die Aktivität des Netzwerk-Dateisystems zu messen, enthält die Vorlage benutzerdefinierte CloudWatch Amazon-Metriken. Die Vorlage erstellt außerdem eine VPC, Subnetz und Sicherheitsgruppen. Damit die Instances mit dem Dateisystem kommunizieren können, muss DNS für die VPC aktiviert sein und das Mount-Ziel und die EC2 Instances müssen sich in derselben Availability Zone (AZ) befinden, die durch das Subnetz spezifiziert wird.

Die Sicherheitsgruppe des Mount-Ziels ermöglicht eine Netzwerkverbindung zum TCP-Port 2049, die erforderlich ist, damit ein NFSv4 Client ein Dateisystem mounten kann. Weitere Informationen zu Sicherheitsgruppen für EC2 Instances und Mount-Ziele finden Sie unter Sicherheit im Amazon Elastic File System-Benutzerhandbuch.

> **Note**
>
> Wenn Sie eine Aktualisierung des Bereitstellungsziels durchführen, dass zu einer Ersetzung führt, werden Instances oder Anwendungen die von den verknüpften Dateisystemen verwenden werden, möglicherweise unterbrochen. Dies kann dazu führen, dass nicht festgeschriebene Schreibvorgänge verloren gehen. Um Unterbrechungen zu vermeiden, beenden Sie Ihre Instances, wenn Sie das Bereitstellungsziel aktualisieren, indem Sie die gewünschte Kapazität auf null setzen. Auf diese Weise können die Instances das Mounting des Dateisystems aufheben, bevor das Bereitstellungsziel gelöscht wird. Nachdem die Mount-Aktualisierung abgeschlossen ist, starten Sie Ihre Instances in einer nachfolgenden Aktualisierung, indem Sie die gewünschte Kapazitäten setzen.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "This template creates an Amazon EFS file system and mount target and
 associates it with Amazon EC2 instances in an Auto Scaling group. **WARNING** This
 template creates Amazon EC2 instances and related resources. You will be billed for
 the AWS resources used if you create a stack from this template.",
  "Parameters": {
    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "t2.small",
      "AllowedValues" : [
        "t1.micro",
        "t2.nano",
```

```
        "t2.micro",
        "t2.small",
        "t2.medium",
        "t2.large",
        "m1.small",
        "m1.medium",
        "m1.large",
        "m1.xlarge",
        "m2.xlarge",
        "m2.2xlarge",
        "m2.4xlarge",
        "m3.medium",
        "m3.large",
        "m3.xlarge",
        "m3.2xlarge",
        "m4.large",
        "m4.xlarge",
        "m4.2xlarge",
        "m4.4xlarge",
        "m4.10xlarge",
        "c1.medium",
        "c1.xlarge",
        "c3.large",
        "c3.xlarge",
        "c3.2xlarge",
        "c3.4xlarge",
        "c3.8xlarge",
        "c4.large",
        "c4.xlarge",
        "c4.2xlarge",
        "c4.4xlarge",
        "c4.8xlarge",
        "g2.2xlarge",
        "g2.8xlarge",
        "r3.large",
        "r3.xlarge",
        "r3.2xlarge",
        "r3.4xlarge",
        "r3.8xlarge",
        "i2.xlarge",
        "i2.2xlarge",
        "i2.4xlarge",
        "i2.8xlarge",
        "d2.xlarge",
```

```
        "d2.2xlarge",
        "d2.4xlarge",
        "d2.8xlarge",
        "hi1.4xlarge",
        "hs1.8xlarge",
        "cr1.8xlarge",
        "cc2.8xlarge",
        "cg1.4xlarge"
      ],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    },
    "KeyName": {
      "Type": "AWS::EC2::KeyPair::KeyName",
      "Description": "Name of an existing EC2 key pair to enable SSH access to the EC2
 instances"
    },
    "AsgMaxSize": {
      "Type": "Number",
      "Description": "Maximum size and initial desired capacity of Auto Scaling Group",
      "Default": "2"
    },
    "SSHLocation" : {
      "Description" : "The IP address range that can be used to connect to the EC2
 instances by using SSH",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    },
    "VolumeName" : {
      "Description" : "The name to be used for the EFS volume",
      "Type": "String",
      "MinLength": "1",
      "Default": "myEFSvolume"
    },
    "MountPoint" : {
      "Description" : "The Linux mount point for the EFS volume",
      "Type": "String",
      "MinLength": "1",
      "Default": "myEFSvolume"
    }
  },
```

```
  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "HVM64"  },
      "t2.nano"     : { "Arch" : "HVM64"  },
      "t2.micro"    : { "Arch" : "HVM64"  },
      "t2.small"    : { "Arch" : "HVM64"  },
      "t2.medium"   : { "Arch" : "HVM64"  },
      "t2.large"    : { "Arch" : "HVM64"  },
      "m1.small"    : { "Arch" : "HVM64"  },
      "m1.medium"   : { "Arch" : "HVM64"  },
      "m1.large"    : { "Arch" : "HVM64"  },
      "m1.xlarge"   : { "Arch" : "HVM64"  },
      "m2.xlarge"   : { "Arch" : "HVM64"  },
      "m2.2xlarge"  : { "Arch" : "HVM64"  },
      "m2.4xlarge"  : { "Arch" : "HVM64"  },
      "m3.medium"   : { "Arch" : "HVM64"  },
      "m3.large"    : { "Arch" : "HVM64"  },
      "m3.xlarge"   : { "Arch" : "HVM64"  },
      "m3.2xlarge"  : { "Arch" : "HVM64"  },
      "m4.large"    : { "Arch" : "HVM64"  },
      "m4.xlarge"   : { "Arch" : "HVM64"  },
      "m4.2xlarge"  : { "Arch" : "HVM64"  },
      "m4.4xlarge"  : { "Arch" : "HVM64"  },
      "m4.10xlarge" : { "Arch" : "HVM64"  },
      "c1.medium"   : { "Arch" : "HVM64"  },
      "c1.xlarge"   : { "Arch" : "HVM64"  },
      "c3.large"    : { "Arch" : "HVM64"  },
      "c3.xlarge"   : { "Arch" : "HVM64"  },
      "c3.2xlarge"  : { "Arch" : "HVM64"  },
      "c3.4xlarge"  : { "Arch" : "HVM64"  },
      "c3.8xlarge"  : { "Arch" : "HVM64"  },
      "c4.large"    : { "Arch" : "HVM64"  },
      "c4.xlarge"   : { "Arch" : "HVM64"  },
      "c4.2xlarge"  : { "Arch" : "HVM64"  },
      "c4.4xlarge"  : { "Arch" : "HVM64"  },
      "c4.8xlarge"  : { "Arch" : "HVM64"  },
      "g2.2xlarge"  : { "Arch" : "HVMG2"  },
      "g2.8xlarge"  : { "Arch" : "HVMG2"  },
      "r3.large"    : { "Arch" : "HVM64"  },
      "r3.xlarge"   : { "Arch" : "HVM64"  },
      "r3.2xlarge"  : { "Arch" : "HVM64"  },
      "r3.4xlarge"  : { "Arch" : "HVM64"  },
      "r3.8xlarge"  : { "Arch" : "HVM64"  },
      "i2.xlarge"   : { "Arch" : "HVM64"  },
```

```
      "i2.2xlarge"  : { "Arch" : "HVM64"  },
      "i2.4xlarge"  : { "Arch" : "HVM64"  },
      "i2.8xlarge"  : { "Arch" : "HVM64"  },
      "d2.xlarge"   : { "Arch" : "HVM64"  },
      "d2.2xlarge"  : { "Arch" : "HVM64"  },
      "d2.4xlarge"  : { "Arch" : "HVM64"  },
      "d2.8xlarge"  : { "Arch" : "HVM64"  },
      "hi1.4xlarge" : { "Arch" : "HVM64"  },
      "hs1.8xlarge" : { "Arch" : "HVM64"  },
      "cr1.8xlarge" : { "Arch" : "HVM64"  },
      "cc2.8xlarge" : { "Arch" : "HVM64"  }
    },
    "AWSRegionArch2AMI" : {
      "us-east-1"        : {"HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
 "ami-0a584ac55a7631c0c"},
      "us-west-2"        : {"HVM64" : "ami-a0cfeed8", "HVMG2" :
 "ami-0e09505bc235aa82d"},
      "us-west-1"        : {"HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
 "ami-066ee5fd4a9ef77f1"},
      "eu-west-1"        : {"HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
 "ami-0a7c483d527806435"},
      "eu-west-2"        : {"HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},
      "eu-west-3"        : {"HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
 "NOT_SUPPORTED"},
      "eu-central-1"     : {"HVM64" : "ami-0233214e13e500f77", "HVMG2" :
 "ami-06223d46a6d0661c7"},
      "ap-northeast-1"   : {"HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
 "ami-053cdd503598e4a9d"},
      "ap-northeast-2"   : {"HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
 "NOT_SUPPORTED"},
      "ap-northeast-3"   : {"HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
 "NOT_SUPPORTED"},
      "ap-southeast-1"   : {"HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
 "ami-0be9df32ae9f92309"},
      "ap-southeast-2"   : {"HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
 "ami-0a9ce9fecc3d1daf8"},
      "ap-south-1"       : {"HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
 "ami-097b15e89dbdcfcf4"},
      "us-east-2"        : {"HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
 "NOT_SUPPORTED"},
      "ca-central-1"     : {"HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
      "sa-east-1"        : {"HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
 "NOT_SUPPORTED"},
```

```
      "cn-north-1"       : {"HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
 "NOT_SUPPORTED"},
      "cn-northwest-1"   : {"HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
    }
  },
  "Resources": {
    "CloudWatchPutMetricsRole" : {
      "Type"  : "AWS::IAM::Role",
      "Properties" : {
          "AssumeRolePolicyDocument" : {
              "Statement" : [ {
                  "Effect" : "Allow",
                  "Principal" : {
                      "Service" : [ "ec2.amazonaws.com" ]
                  },
                  "Action" : [ "sts:AssumeRole" ]
              } ]
          },
          "Path" : "/"
      }
    },
    "CloudWatchPutMetricsRolePolicy" : {
        "Type" : "AWS::IAM::Policy",
        "Properties" : {
            "PolicyName" : "CloudWatch_PutMetricData",
            "PolicyDocument" : {
              "Version": "2012-10-17",
              "Statement": [
                {
                  "Sid": "CloudWatchPutMetricData",
                  "Effect": "Allow",
                  "Action": ["cloudwatch:PutMetricData"],
                  "Resource": ["*"]
                }
              ]
            },
            "Roles" : [ { "Ref" : "CloudWatchPutMetricsRole" } ]
        }
    },
    "CloudWatchPutMetricsInstanceProfile" : {
      "Type" : "AWS::IAM::InstanceProfile",
      "Properties" : {
        "Path" : "/",
        "Roles" : [ { "Ref" : "CloudWatchPutMetricsRole" } ]
```

```
      }
    },
    "VPC": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "EnableDnsSupport" : "true",
        "EnableDnsHostnames" : "true",
        "CidrBlock": "10.0.0.0/16",
        "Tags": [ {"Key": "Application", "Value": { "Ref": "AWS::StackId"} } ]
      }
    },
    "InternetGateway" : {
      "Type" : "AWS::EC2::InternetGateway",
      "Properties" : {
        "Tags" : [
          { "Key" : "Application", "Value" : { "Ref" : "AWS::StackName" } },
          { "Key" : "Network", "Value" : "Public" }
        ]
      }
    },
    "GatewayToInternet" : {
      "Type" : "AWS::EC2::VPCGatewayAttachment",
      "Properties" : {
        "VpcId" : { "Ref" : "VPC" },
        "InternetGatewayId" : { "Ref" : "InternetGateway" }
      }
    },
    "RouteTable":{
      "Type":"AWS::EC2::RouteTable",
      "Properties":{
        "VpcId": {"Ref":"VPC"}
      }
    },
    "SubnetRouteTableAssoc": {
      "Type" : "AWS::EC2::SubnetRouteTableAssociation",
      "Properties" : {
        "RouteTableId" : {"Ref":"RouteTable"},
        "SubnetId" : {"Ref":"Subnet"}
      }
    },
    "InternetGatewayRoute": {
        "Type":"AWS::EC2::Route",
        "Properties":{
            "DestinationCidrBlock":"0.0.0.0/0",
```

```
            "RouteTableId":{"Ref":"RouteTable"},
            "GatewayId":{"Ref":"InternetGateway"}
        }
    },
    "Subnet": {
      "Type": "AWS::EC2::Subnet",
      "Properties": {
        "VpcId": { "Ref": "VPC" },
        "CidrBlock": "10.0.0.0/24",
        "Tags": [ { "Key": "Application", "Value": { "Ref": "AWS::StackId" } } ]
      }
    },
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "VpcId": { "Ref": "VPC" },
        "GroupDescription": "Enable SSH access via port 22",
        "SecurityGroupIngress": [
          { "IpProtocol": "tcp", "FromPort": 22, "ToPort": 22, "CidrIp": { "Ref":
 "SSHLocation" } },
          { "IpProtocol": "tcp", "FromPort": 80, "ToPort": 80, "CidrIp": "0.0.0.0/0" }
         ]
      }
    },
    "MountTargetSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "VpcId": { "Ref": "VPC" },
        "GroupDescription": "Security group for mount target",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 2049,
            "ToPort": 2049,
            "CidrIp": "0.0.0.0/0"
          }
        ]
      }
    },
    "FileSystem": {
      "Type": "AWS::EFS::FileSystem",
      "Properties": {
        "PerformanceMode": "generalPurpose",
        "FileSystemTags": [
```

```
          {
            "Key": "Name",
            "Value": { "Ref" : "VolumeName" }
          }
        ]
      }
    },
    "MountTarget": {
      "Type": "AWS::EFS::MountTarget",
      "Properties": {
        "FileSystemId": { "Ref": "FileSystem" },
        "SubnetId": { "Ref": "Subnet" },
        "SecurityGroups": [ { "Ref": "MountTargetSecurityGroup" } ]
      }
    },
    "LaunchConfiguration": {
      "Type": "AWS::AutoScaling::LaunchConfiguration",
      "Metadata" : {
        "AWS::CloudFormation::Init" : {
          "configSets" : {
            "MountConfig" : [ "setup", "mount" ]
          },
          "setup" : {
            "packages" : {
              "yum" : {
                "nfs-utils" : []
              }
            },
            "files" : {
              "/home/ec2-user/post_nfsstat" : {
                "content" : { "Fn::Join" : [ "", [
                      "#!/bin/bash\n",
                      "\n",
                      "INPUT=\"$(cat)\"\n",
                      "CW_JSON_OPEN='{ \"Namespace\": \"EFS\", \"MetricData\": [ '\n",
                      "CW_JSON_CLOSE=' ] }'\n",
                      "CW_JSON_METRIC=''\n",
                      "METRIC_COUNTER=0\n",
                      "\n",
                      "for COL in 1 2 3 4 5 6; do\n",
                      "\n",
                      " COUNTER=0\n",
                      " METRIC_FIELD=$COL\n",
                      " DATA_FIELD=$(($COL+($COL-1)))\n",
```

```
                        "\n",
                        " while read line; do\n",
                        "   if [[ COUNTER -gt 0 ]]; then\n",
                        "\n",
                        "     LINE=`echo $line | tr -s ' ' `\n",
                        "     AWS_COMMAND=\"aws cloudwatch put-metric-data --region ",
 { "Ref": "AWS::Region" }, "\"\n",
                        "     MOD=$(( $COUNTER % 2))\n",
                        "\n",
                        "     if [ $MOD -eq 1 ]; then\n",
                        "       METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`
\n",
                        "     else\n",
                        "       METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`\n",
                        "     fi\n",
                        "\n",
                        "     if [[ -n \"$METRIC_NAME\" && -n \"$METRIC_VALUE\" ]]; then
\n",
                        "       INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-
data/instance-id)\n",
                        "       CW_JSON_METRIC=\"$CW_JSON_METRIC { \\\"MetricName\\\": \\
\"$METRIC_NAME\\\", \\\"Dimensions\\\": [{\\\"Name\\\": \\\"InstanceId\\\", \\\"Value\\
\": \\\"$INSTANCE_ID\\\"} ], \\\"Value\\\": $METRIC_VALUE },\"\n",
                        "       unset METRIC_NAME\n",
                        "       unset METRIC_VALUE\n",
                        "\n",
                        "       METRIC_COUNTER=$((METRIC_COUNTER+1))\n",
                        "       if [ $METRIC_COUNTER -eq 20 ]; then\n",
                        "         # 20 is max metric collection size, so we have to
 submit here\n",
                        "         aws cloudwatch put-metric-data --region ", { "Ref":
 "AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}
 $CW_JSON_CLOSE`\"\n",
                        "\n",
                        "         # reset\n",
                        "         METRIC_COUNTER=0\n",
                        "         CW_JSON_METRIC=''\n",
                        "       fi\n",
                        "     fi  \n",
                        "\n",
                        "\n",
                        "\n",
                        "     COUNTER=$((COUNTER+1))\n",
                        "   fi\n",
```

```
                      "\n",
                      "   if [[ \"$line\" == \"Client nfs v4:\" ]]; then\n",
                      "     # the next line is the good stuff \n",
                      "     COUNTER=$((COUNTER+1))\n",
                      "   fi\n",
                      " done <<< \"$INPUT\"\n",
                      "done\n",
                      "\n",
                      "# submit whatever is left\n",
                      "aws cloudwatch put-metric-data --region ", { "Ref":
 "AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}
 $CW_JSON_CLOSE`\""
                    ] ] },
                "mode": "000755",
                "owner": "ec2-user",
                "group": "ec2-user"
              },
              "/home/ec2-user/crontab" : {
                "content" : { "Fn::Join" : [ "", [
                  "* * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
                ] ] },
                "owner": "ec2-user",
                "group": "ec2-user"
              }
            },
            "commands" : {
              "01_createdir" : {
                "command" : {"Fn::Join" : [ "", [ "mkdir /", { "Ref" :
 "MountPoint" }]]}
              }
            }
          },
          "mount" : {
            "commands" : {
              "01_mount" : {
                "command" : { "Fn::Sub": "sudo mount -t nfs4 -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 ${FileSystem}.efs.
${AWS::Region}.amazonaws.com:/ /${MountPoint}"}
              },
              "02_permissions" : {
                "command" : {"Fn::Join" : [ "", [ "chown ec2-user:ec2-user /",
 { "Ref" : "MountPoint" }]]}
              }
            }
```

```
          }
        }
      },
      "Properties": {
        "AssociatePublicIpAddress" : true,
        "ImageId": {
          "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" }, {
            "Fn::FindInMap": [ "AWSInstanceType2Arch", { "Ref": "InstanceType" },
 "Arch" ]
          } ]
        },
        "InstanceType": { "Ref": "InstanceType" },
        "KeyName": { "Ref": "KeyName" },
        "SecurityGroups": [ { "Ref": "InstanceSecurityGroup" } ],
        "IamInstanceProfile" : { "Ref" : "CloudWatchPutMetricsInstanceProfile" },
        "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
             "#!/bin/bash -xe\n",
             "yum install -y aws-cfn-bootstrap\n",

             "/opt/aws/bin/cfn-init -v ",
             "         --stack ", { "Ref" : "AWS::StackName" },
             "         --resource LaunchConfiguration ",
             "         --configsets MountConfig ",
             "         --region ", { "Ref" : "AWS::Region" }, "\n",

             "crontab /home/ec2-user/crontab\n",

             "/opt/aws/bin/cfn-signal -e $? ",
             "         --stack ", { "Ref" : "AWS::StackName" },
             "         --resource AutoScalingGroup ",
             "         --region ", { "Ref" : "AWS::Region" }, "\n"
        ]]}}
      }
    },
    "AutoScalingGroup": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "DependsOn": ["MountTarget", "GatewayToInternet"],
      "CreationPolicy" : {
        "ResourceSignal" : {
          "Timeout" : "PT15M",
          "Count"   : { "Ref": "AsgMaxSize" }
        }
      },
      "Properties": {
```

```
        "VPCZoneIdentifier": [ { "Ref": "Subnet" } ],
        "LaunchConfigurationName": { "Ref": "LaunchConfiguration" },
        "MinSize": "1",
        "MaxSize": { "Ref": "AsgMaxSize" },
        "DesiredCapacity": { "Ref": "AsgMaxSize" },
        "Tags": [ {
          "Key": "Name",
          "Value": "EFS FileSystem Mounted Instance",
          "PropagateAtLaunch": "true"
        } ]
      }
    }
  },
  "Outputs" : {
    "MountTargetID" : {
      "Description" : "Mount target ID",
      "Value" :  { "Ref" : "MountTarget" }
    },
    "FileSystemID" : {
      "Description" : "File system ID",
      "Value" :  { "Ref" : "FileSystem" }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Description: This template creates an Amazon EFS file system and mount target and
  associates it with Amazon EC2 instances in an Auto Scaling group. **WARNING** This
  template creates Amazon EC2 instances and related resources. You will be billed
  for the AWS resources used if you create a stack from this template.
Parameters:
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.small
    AllowedValues:
      - t1.micro
      - t2.nano
      - t2.micro
      - t2.small
      - t2.medium
```

```
      - t2.large
      - m1.small
      - m1.medium
      - m1.large
      - m1.xlarge
      - m2.xlarge
      - m2.2xlarge
      - m2.4xlarge
      - m3.medium
      - m3.large
      - m3.xlarge
      - m3.2xlarge
      - m4.large
      - m4.xlarge
      - m4.2xlarge
      - m4.4xlarge
      - m4.10xlarge
      - c1.medium
      - c1.xlarge
      - c3.large
      - c3.xlarge
      - c3.2xlarge
      - c3.4xlarge
      - c3.8xlarge
      - c4.large
      - c4.xlarge
      - c4.2xlarge
      - c4.4xlarge
      - c4.8xlarge
      - g2.2xlarge
      - g2.8xlarge
      - r3.large
      - r3.xlarge
      - r3.2xlarge
      - r3.4xlarge
      - r3.8xlarge
      - i2.xlarge
      - i2.2xlarge
      - i2.4xlarge
      - i2.8xlarge
      - d2.xlarge
      - d2.2xlarge
      - d2.4xlarge
      - d2.8xlarge
```

```
      - hi1.4xlarge
      - hs1.8xlarge
      - cr1.8xlarge
      - cc2.8xlarge
      - cg1.4xlarge
    ConstraintDescription: must be a valid EC2 instance type.
  KeyName:
    Type: AWS::EC2::KeyPair::KeyName
    Description: Name of an existing EC2 key pair to enable SSH access to the ECS
      instances
  AsgMaxSize:
    Type: Number
    Description: Maximum size and initial desired capacity of Auto Scaling Group
    Default: '2'
  SSHLocation:
    Description: The IP address range that can be used to connect to the EC2 instances
      by using SSH
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})"
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  VolumeName:
    Description: The name to be used for the EFS volume
    Type: String
    MinLength: '1'
    Default: myEFSvolume
  MountPoint:
    Description: The Linux mount point for the EFS volume
    Type: String
    MinLength: '1'
    Default: myEFSvolume
Mappings:
  AWSInstanceType2Arch:
    t1.micro:
      Arch: HVM64
    t2.nano:
      Arch: HVM64
    t2.micro:
      Arch: HVM64
    t2.small:
      Arch: HVM64
    t2.medium:
```

```
      Arch: HVM64
    t2.large:
      Arch: HVM64
    m1.small:
      Arch: HVM64
    m1.medium:
      Arch: HVM64
    m1.large:
      Arch: HVM64
    m1.xlarge:
      Arch: HVM64
    m2.xlarge:
      Arch: HVM64
    m2.2xlarge:
      Arch: HVM64
    m2.4xlarge:
      Arch: HVM64
    m3.medium:
      Arch: HVM64
    m3.large:
      Arch: HVM64
    m3.xlarge:
      Arch: HVM64
    m3.2xlarge:
      Arch: HVM64
    m4.large:
      Arch: HVM64
    m4.xlarge:
      Arch: HVM64
    m4.2xlarge:
      Arch: HVM64
    m4.4xlarge:
      Arch: HVM64
    m4.10xlarge:
      Arch: HVM64
    c1.medium:
      Arch: HVM64
    c1.xlarge:
      Arch: HVM64
    c3.large:
      Arch: HVM64
    c3.xlarge:
      Arch: HVM64
    c3.2xlarge:
```

```
      Arch: HVM64
    c3.4xlarge:
      Arch: HVM64
    c3.8xlarge:
      Arch: HVM64
    c4.large:
      Arch: HVM64
    c4.xlarge:
      Arch: HVM64
    c4.2xlarge:
      Arch: HVM64
    c4.4xlarge:
      Arch: HVM64
    c4.8xlarge:
      Arch: HVM64
    g2.2xlarge:
      Arch: HVMG2
    g2.8xlarge:
      Arch: HVMG2
    r3.large:
      Arch: HVM64
    r3.xlarge:
      Arch: HVM64
    r3.2xlarge:
      Arch: HVM64
    r3.4xlarge:
      Arch: HVM64
    r3.8xlarge:
      Arch: HVM64
    i2.xlarge:
      Arch: HVM64
    i2.2xlarge:
      Arch: HVM64
    i2.4xlarge:
      Arch: HVM64
    i2.8xlarge:
      Arch: HVM64
    d2.xlarge:
      Arch: HVM64
    d2.2xlarge:
      Arch: HVM64
    d2.4xlarge:
      Arch: HVM64
    d2.8xlarge:
```

```
      Arch: HVM64
    hi1.4xlarge:
      Arch: HVM64
    hs1.8xlarge:
      Arch: HVM64
    cr1.8xlarge:
      Arch: HVM64
    cc2.8xlarge:
      Arch: HVM64
  AWSRegionArch2AMI:
    us-east-1:
      HVM64: ami-0ff8a91507f77f867
      HVMG2: ami-0a584ac55a7631c0c
    us-west-2:
      HVM64: ami-a0cfeed8
      HVMG2: ami-0e09505bc235aa82d
    us-west-1:
      HVM64: ami-0bdb828fd58c52235
      HVMG2: ami-066ee5fd4a9ef77f1
    eu-west-1:
      HVM64: ami-047bb4163c506cd98
      HVMG2: ami-0a7c483d527806435
    eu-west-2:
      HVM64: ami-f976839e
      HVMG2: NOT_SUPPORTED
    eu-west-3:
      HVM64: ami-0ebc281c20e89ba4b
      HVMG2: NOT_SUPPORTED
    eu-central-1:
      HVM64: ami-0233214e13e500f77
      HVMG2: ami-06223d46a6d0661c7
    ap-northeast-1:
      HVM64: ami-06cd52961ce9f0d85
      HVMG2: ami-053cdd503598e4a9d
    ap-northeast-2:
      HVM64: ami-0a10b2721688ce9d2
      HVMG2: NOT_SUPPORTED
    ap-northeast-3:
      HVM64: ami-0d98120a9fb693f07
      HVMG2: NOT_SUPPORTED
    ap-southeast-1:
      HVM64: ami-08569b978cc4dfa10
      HVMG2: ami-0be9df32ae9f92309
    ap-southeast-2:
```

```
      HVM64: ami-09b42976632b27e9b
      HVMG2: ami-0a9ce9fecc3d1daf8
    ap-south-1:
      HVM64: ami-0912f71e06545ad88
      HVMG2: ami-097b15e89dbdcfcf4
    us-east-2:
      HVM64: ami-0b59bfac6be064b78
      HVMG2: NOT_SUPPORTED
    ca-central-1:
      HVM64: ami-0b18956f
      HVMG2: NOT_SUPPORTED
    sa-east-1:
      HVM64: ami-07b14488da8ea02a0
      HVMG2: NOT_SUPPORTED
    cn-north-1:
      HVM64: ami-0a4eaf6c4454eda75
      HVMG2: NOT_SUPPORTED
    cn-northwest-1:
      HVM64: ami-6b6a7d09
      HVMG2: NOT_SUPPORTED
Resources:
  CloudWatchPutMetricsRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
        - Effect: Allow
          Principal:
            Service:
            - ec2.amazonaws.com
          Action:
          - sts:AssumeRole
      Path: "/"
  CloudWatchPutMetricsRolePolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: CloudWatch_PutMetricData
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Sid: CloudWatchPutMetricData
          Effect: Allow
          Action:
          - cloudwatch:PutMetricData
```

```
          Resource:
          - "*"
      Roles:
      - Ref: CloudWatchPutMetricsRole
  CloudWatchPutMetricsInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: "/"
      Roles:
      - Ref: CloudWatchPutMetricsRole
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
      CidrBlock: 10.0.0.0/16
      Tags:
      - Key: Application
        Value:
          Ref: AWS::StackId
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
      - Key: Application
        Value:
          Ref: AWS::StackName
      - Key: Network
        Value: Public
  GatewayToInternet:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId:
        Ref: VPC
      InternetGatewayId:
        Ref: InternetGateway
  RouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId:
        Ref: VPC
  SubnetRouteTableAssoc:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
```

```
      RouteTableId:
        Ref: RouteTable
      SubnetId:
        Ref: Subnet
  InternetGatewayRoute:
    Type: AWS::EC2::Route
    Properties:
      DestinationCidrBlock: 0.0.0.0/0
      RouteTableId:
        Ref: RouteTable
      GatewayId:
        Ref: InternetGateway
  Subnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      Tags:
      - Key: Application
        Value:
          Ref: AWS::StackId
  InstanceSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      VpcId:
        Ref: VPC
      GroupDescription: Enable SSH access via port 22
      SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp:
          Ref: SSHLocation
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
  MountTargetSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      VpcId:
        Ref: VPC
      GroupDescription: Security group for mount target
```

```
      SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 2049
        ToPort: 2049
        CidrIp: 0.0.0.0/0
  FileSystem:
    Type: AWS::EFS::FileSystem
    Properties:
      PerformanceMode: generalPurpose
      FileSystemTags:
      - Key: Name
        Value:
          Ref: VolumeName
  MountTarget:
    Type: AWS::EFS::MountTarget
    Properties:
      FileSystemId:
        Ref: FileSystem
      SubnetId:
        Ref: Subnet
      SecurityGroups:
      - Ref: MountTargetSecurityGroup
  LaunchConfiguration:
    Type: AWS::AutoScaling::LaunchConfiguration
    Metadata:
      AWS::CloudFormation::Init:
        configSets:
          MountConfig:
          - setup
          - mount
        setup:
          packages:
            yum:
              nfs-utils: []
          files:
            "/home/ec2-user/post_nfsstat":
              content: !Sub |
                #!/bin/bash

                INPUT="$(cat)"
                CW_JSON_OPEN='{ "Namespace": "EFS", "MetricData": [ '
                CW_JSON_CLOSE=' ] }'
                CW_JSON_METRIC=''
                METRIC_COUNTER=0
```

```
                for COL in 1 2 3 4 5 6; do

                 COUNTER=0
                 METRIC_FIELD=$COL
                 DATA_FIELD=$(($COL+($COL-1)))

                 while read line; do
                   if [[ COUNTER -gt 0 ]]; then

                     LINE=`echo $line | tr -s ' ' `
                     AWS_COMMAND="aws cloudwatch put-metric-data --region
 ${AWS::Region}"
                     MOD=$(( $COUNTER % 2))

                     if [ $MOD -eq 1 ]; then
                       METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`
                     else
                       METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`
                     fi

                     if [[ -n "$METRIC_NAME" && -n "$METRIC_VALUE" ]]; then
                       INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/
instance-id)
                       CW_JSON_METRIC="$CW_JSON_METRIC { \"MetricName\": \"$METRIC_NAME
\", \"Dimensions\": [{\"Name\": \"InstanceId\", \"Value\": \"$INSTANCE_ID\"} ], \"Value
\": $METRIC_VALUE },"
                       unset METRIC_NAME
                       unset METRIC_VALUE

                       METRIC_COUNTER=$((METRIC_COUNTER+1))
                       if [ $METRIC_COUNTER -eq 20 ]; then
                         # 20 is max metric collection size, so we have to submit here
                         aws cloudwatch put-metric-data --region ${AWS::Region} --cli-
input-json "`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"

                         # reset
                         METRIC_COUNTER=0
                         CW_JSON_METRIC=''
                       fi
                     fi
```

```
                      COUNTER=$((COUNTER+1))
                    fi

                    if [[ "$line" == "Client nfs v4:" ]]; then
                      # the next line is the good stuff
                      COUNTER=$((COUNTER+1))
                    fi
                  done <<< "$INPUT"
                done

                # submit whatever is left
                aws cloudwatch put-metric-data --region ${AWS::Region} --cli-input-json
 "`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"
              mode: '000755'
              owner: ec2-user
              group: ec2-user
            "/home/ec2-user/crontab":
              content: "* * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
              owner: ec2-user
              group: ec2-user
          commands:
            01_createdir:
              command: !Sub "mkdir /${MountPoint}"
        mount:
          commands:
            01_mount:
              command: !Sub >
                mount -t nfs4 -o nfsvers=4.1 ${FileSystem}.efs.
${AWS::Region}.amazonaws.com:/ /${MountPoint}
            02_permissions:
              command: !Sub "chown ec2-user:ec2-user /${MountPoint}"
    Properties:
      AssociatePublicIpAddress: true
      ImageId:
        Fn::FindInMap:
        - AWSRegionArch2AMI
        - Ref: AWS::Region
        - Fn::FindInMap:
          - AWSInstanceType2Arch
          - Ref: InstanceType
          - Arch
      InstanceType:
        Ref: InstanceType
      KeyName:
```

```
        Ref: KeyName
      SecurityGroups:
      - Ref: InstanceSecurityGroup
      IamInstanceProfile:
        Ref: CloudWatchPutMetricsInstanceProfile
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash -xe
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
 LaunchConfiguration --configsets MountConfig --region ${AWS::Region}
          crontab /home/ec2-user/crontab
          /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
 AutoScalingGroup --region ${AWS::Region}
  AutoScalingGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    DependsOn:
    - MountTarget
    - GatewayToInternet
    CreationPolicy:
      ResourceSignal:
        Timeout: PT15M
        Count:
          Ref: AsgMaxSize
    Properties:
      VPCZoneIdentifier:
      - Ref: Subnet
      LaunchConfigurationName:
        Ref: LaunchConfiguration
      MinSize: '1'
      MaxSize:
        Ref: AsgMaxSize
      DesiredCapacity:
        Ref: AsgMaxSize
      Tags:
      - Key: Name
        Value: EFS FileSystem Mounted Instance
        PropagateAtLaunch: 'true'
Outputs:
  MountTargetID:
    Description: Mount target ID
    Value:
      Ref: MountTarget
  FileSystemID:
```

```
    Description: File system ID
    Value:
      Ref: FileSystem
```

Elastic Beanstalk-Vorlagenausschnitte

Mit Elastic Beanstalk können Sie Anwendungen schnell bereitstellen und verwalten, AWS ohne sich Gedanken über die Infrastruktur machen zu müssen, auf der diese Anwendungen ausgeführt werden. Die folgende Beispielvorlage kann Ihnen helfen, Elastic Beanstalk Beanstalk-Ressourcen in Ihrer AWS CloudFormation Vorlage zu beschreiben.

Elastic Beanstalk-Beispiel-PHP

Die folgende Beispielvorlage stellt eine PHP-Beispielwebanwendung bereit, die in einem Amazon S3-Bucket gespeichert wird. Die Umgebung ist auch eine Umgebung mit auto-scaling und Lastenausgleich mit mindestens zwei EC2 Amazon-Instances und maximal sechs. Es zeigt eine Elastic Beanstalk Beanstalk-Umgebung, die eine ältere Startkonfiguration verwendet. Informationen darüber, wie Sie stattdessen eine Startvorlage verwenden können, finden Sie unter Launch Templates im AWS Elastic Beanstalk Developer Guide.

Ersetzen Sie *solution-stack* durch einen Lösungs-Stack-Namen (Plattformversion). Verwenden Sie den AWS CLI Befehlaws elasticbeanstalk list-available-solution-stacks, um eine Liste der verfügbaren Lösungsstapel zu erhalten.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "sampleApplication": {
            "Type": "AWS::ElasticBeanstalk::Application",
            "Properties": {
                "Description": "AWS Elastic Beanstalk Sample Application"
            }
        },
        "sampleApplicationVersion": {
            "Type": "AWS::ElasticBeanstalk::ApplicationVersion",
            "Properties": {
                "ApplicationName": {
                    "Ref": "sampleApplication"
                },
                "Description": "AWS ElasticBeanstalk Sample Application Version",
```

```
                "SourceBundle": {
                    "S3Bucket": {
                        "Fn::Sub": "elasticbeanstalk-samples-${AWS::Region}"
                    },
                    "S3Key": "php-newsample-app.zip"
                }
            }
        },
        "sampleConfigurationTemplate": {
            "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
            "Properties": {
                "ApplicationName": {
                    "Ref": "sampleApplication"
                },
                "Description": "AWS ElasticBeanstalk Sample Configuration Template",
                "OptionSettings": [
                    {
                        "Namespace": "aws:autoscaling:asg",
                        "OptionName": "MinSize",
                        "Value": "2"
                    },
                    {
                        "Namespace": "aws:autoscaling:asg",
                        "OptionName": "MaxSize",
                        "Value": "6"
                    },
                    {
                        "Namespace": "aws:elasticbeanstalk:environment",
                        "OptionName": "EnvironmentType",
                        "Value": "LoadBalanced"
                    },
                    {
                        "Namespace": "aws:autoscaling:launchconfiguration",
                        "OptionName": "IamInstanceProfile",
                        "Value": {
                            "Ref": "MyInstanceProfile"
                        }
                    }
                ],
                "SolutionStackName": "solution-stack"
            }
        },
        "sampleEnvironment": {
            "Type": "AWS::ElasticBeanstalk::Environment",
```

```
            "Properties": {
                "ApplicationName": {
                    "Ref": "sampleApplication"
                },
                "Description": "AWS ElasticBeanstalk Sample Environment",
                "TemplateName": {
                    "Ref": "sampleConfigurationTemplate"
                },
                "VersionLabel": {
                    "Ref": "sampleApplicationVersion"
                }
            }
        },
        "MyInstanceRole": {
            "Type": "AWS::IAM::Role",
            "Properties": {
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": [
                                    "ec2.amazonaws.com"
                                ]
                            },
                            "Action": [
                                "sts:AssumeRole"
                            ]
                        }
                    ]
                },
                "Description": "Beanstalk EC2 role",
                "ManagedPolicyArns": [
                    "arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier",
                    "arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker",
                    "arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier"
                ]
            }
        },
        "MyInstanceProfile": {
            "Type": "AWS::IAM::InstanceProfile",
            "Properties": {
                "Roles": [
```

```
                    {
                        "Ref": "MyInstanceRole"
                    }
                ]
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  sampleApplication:
    Type: AWS::ElasticBeanstalk::Application
    Properties:
      Description: AWS Elastic Beanstalk Sample Application
  sampleApplicationVersion:
    Type: AWS::ElasticBeanstalk::ApplicationVersion
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Application Version
      SourceBundle:
        S3Bucket: !Sub "elasticbeanstalk-samples-${AWS::Region}"
        S3Key: php-newsample-app.zip
  sampleConfigurationTemplate:
    Type: AWS::ElasticBeanstalk::ConfigurationTemplate
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Configuration Template
      OptionSettings:
      - Namespace: aws:autoscaling:asg
        OptionName: MinSize
        Value: '2'
      - Namespace: aws:autoscaling:asg
        OptionName: MaxSize
        Value: '6'
      - Namespace: aws:elasticbeanstalk:environment
        OptionName: EnvironmentType
        Value: LoadBalanced
      - Namespace: aws:autoscaling:launchconfiguration
```

```
        OptionName: IamInstanceProfile
        Value: !Ref MyInstanceProfile
      SolutionStackName: solution-stack
  sampleEnvironment:
    Type: AWS::ElasticBeanstalk::Environment
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Environment
      TemplateName:
        Ref: sampleConfigurationTemplate
      VersionLabel:
        Ref: sampleApplicationVersion
  MyInstanceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - sts:AssumeRole
      Description: Beanstalk EC2 role
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier
        - arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker
        - arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier
  MyInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Roles:
        - !Ref MyInstanceRole
```

Elastic Load Balancing-Vorlagenausschnitte

Um einen Application Load Balancer, einen Network Load Balancer oder einen Gateway Load Balancer zu erstellen, verwenden Sie die V2-Ressourcentypen, die mit beginnen. `AWS::ElasticLoadBalancingV2` Um einen Classic Load Balancer zu erstellen, verwenden Sie die Ressourcentypen, die mit `AWS::ElasticLoadBalancing` beginnen.

Inhalt

- ELBv2 Ressourcen
- Ressourcen für den Classic Load Balancer

ELBv2 Ressourcen

Dieses Beispiel definiert einen Application Load Balancer mit einem HTTP-Listener und einer Standardaktion, die den Traffic an die Zielgruppe weiterleitet. Der Load Balancer verwendet die Standardeinstellungen für die Integritätsprüfung. Die Zielgruppe hat zwei registrierte EC2 Instanzen.

YAML

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup

  myHTTPlistener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTP
      Port: 80
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup

  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: "my-target-group"
      Protocol: HTTP
      Port: 80
```

```
      TargetType: instance
      VpcId: !Ref myVPC
      Targets:
        - Id: !GetAtt Instance1.InstanceId
          Port: 80
        - Id: !GetAtt Instance2.InstanceId
          Port: 80
```

JSON

```
{
    "Resources": {
        "myLoadBalancer": {
            "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
            "Properties": {
                "Name": "my-alb",
                "Type": "application",
                "Scheme": "internal",
                "Subnets": [
                    {
                        "Ref": "subnet-AZ1"
                    },
                    {
                        "Ref": "subnet-AZ2"
                    }
                ],
                "SecurityGroups": [
                    {
                        "Ref": "mySecurityGroup"
                    }
                ]
            }
        },
        "myHTTPlistener": {
            "Type": "AWS::ElasticLoadBalancingV2::Listener",
            "Properties": {
                "LoadBalancerArn": {
                    "Ref": "myLoadBalancer"
                },
                "Protocol": "HTTP",
                "Port": 80,
                "DefaultActions": [
                    {
```

```
                        "Type": "forward",
                        "TargetGroupArn": {
                            "Ref": "myTargetGroup"
                        }
                    }
                ]
            }
        },
        "myTargetGroup": {
            "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
            "Properties": {
                "Name": "my-target-group",
                "Protocol": "HTTP",
                "Port": 80,
                "TargetType": "instance",
                "VpcId": {
                    "Ref": "myVPC"
                },
                "Targets": [
                    {
                        "Id": {
                            "Fn::GetAtt": [
                                "Instance1",
                                "InstanceId"
                            ]
                        },
                        "Port": 80
                    },
                    {
                        "Id": {
                            "Fn::GetAtt": [
                                "Instance2",
                                "InstanceId"
                            ]
                        },
                        "Port": 80
                    }
                ]
            }
        }
    }
}
```

Ressourcen für den Classic Load Balancer

Dieses Beispiel definiert einen Classic Load Balancer mit einem HTTP-Listener und ohne registrierte EC2 Instances. Der Load Balancer verwendet die Standardeinstellungen für die Integritätsprüfung.

YAML

```
myLoadBalancer:
  Type: 'AWS::ElasticLoadBalancing::LoadBalancer'
  Properties:
    AvailabilityZones:
    - "us-east-1a"
    Listeners:
    - LoadBalancerPort: '80'
      InstancePort: '80'
      Protocol: HTTP
```

JSON

```
"myLoadBalancer" : {
    "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
    "Properties" : {
        "AvailabilityZones" : [ "us-east-1a" ],
        "Listeners" : [ {
            "LoadBalancerPort" : "80",
            "InstancePort" : "80",
            "Protocol" : "HTTP"
        } ]
    }
}
```

In diesem Beispiel wird ein Classic Load Balancer mit einem HTTP-Listener, zwei registrierten EC2 Instances und benutzerdefinierten Einstellungen für die Zustandsprüfung definiert.

YAML

```
myClassicLoadBalancer:
  Type: 'AWS::ElasticLoadBalancing::LoadBalancer'
  Properties:
    AvailabilityZones:
    - "us-east-1a"
```

```
    Instances:
    - Ref: Instance1
    - Ref: Instance2
    Listeners:
    - LoadBalancerPort: '80'
      InstancePort: '80'
      Protocol: HTTP
    HealthCheck:
      Target: HTTP:80/
      HealthyThreshold: '3'
      UnhealthyThreshold: '5'
      Interval: '30'
      Timeout: '5'
```

JSON

```
"myClassicLoadBalancer" : {
    "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
    "Properties" : {
        "AvailabilityZones" : [ "us-east-1a" ],
        "Instances" : [
            { "Ref" : "Instance1" },
            { "Ref" : "Instance2" }
        ],
        "Listeners" : [ {
            "LoadBalancerPort" : "80",
            "InstancePort" : "80",
            "Protocol" : "HTTP"
        } ],

        "HealthCheck" : {
            "Target" : "HTTP:80/",
            "HealthyThreshold" : "3",
            "UnhealthyThreshold" : "5",
            "Interval" : "30",
            "Timeout" : "5"
        }
    }
}
```

AWS Identity and Access Management Schnipsel aus Vorlagen

Dieser Abschnitt enthält AWS Identity and Access Management Vorlagenausschnitte.

Themen

- Deklarieren einer IAM-Benutzerressource
- Deklarieren einer IAM-Zugriffsschlüssel-Ressource
- Deklarieren einer IAM-Gruppenressource
- Hinzufügen von Benutzern zu einer Gruppe
- Deklarieren einer IAM-Richtlinie
- Deklarieren einer Amazon S3-Bucket-Richtlinie
- Deklarieren einer Amazon SNS-Themenrichtlinie
- Deklarieren einer Amazon SQS-Richtlinie
- Vorlagenbeispiele für IAM-Rollen

> **Important**
>
> Beim Erstellen oder Aktualisieren eines Stacks anhand einer Vorlage, die IAM-Ressourcen enthält, müssen Sie die Verwendung von IAM-Funktionen bestätigen. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.

Deklarieren einer IAM-Benutzerressource

Dieses Codefragment zeigt, wie eine AWS::IAM::User-Ressource deklariert wird, um einen IAM-Benutzer zu erstellen. Der Benutzer wird mit dem Pfad (`"/"`) und einem Anmeldungsprofil mit dem Passwort (`myP@ssW0rd`) deklariert.

Im Richtliniendokument namens `giveaccesstoqueueonly` erhält der Benutzer die Berechtigung zum Ausführen aller Amazon SQS-Aktionen auf der Amazon SQS-Warteschlangenressource `myqueue`. Der Zugriff auf alle anderen Amazon SQS-Warteschlangenressourcen wird verweigert. Die `Fn::GetAtt`-Funktion ruft das Arn-Attribut der AWS::SQS::Queue-Ressource `myqueue` ab.

Das Richtliniendokument namens `giveaccesstotopiconly` wird dem Benutzer hinzugefügt, um ihm die Berechtigung zum Ausführen aller Amazon SNS-Aktionen auf der Amazon SNS-Themenressource `mytopic` zu erteilen und den Zugriff auf alle anderen Amazon SNS-Ressourcen

zu verweigern. Die `Ref`-Funktion ruft das ARN-Attribut der AWS::SNS::Topic-Ressource `mytopic` ab.

JSON

```
"myuser" : {
   "Type" : "AWS::IAM::User",
   "Properties" : {
      "Path" : "/",
      "LoginProfile" : {
         "Password" : "myP@ssW0rd"
      },
      "Policies" : [ {
         "PolicyName" : "giveaccesstoqueueonly",
         "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement" : [ {
               "Effect" : "Allow",
               "Action" : [ "sqs:*" ],
               "Resource" : [ {
                  "Fn::GetAtt" : [ "myqueue", "Arn" ]
               } ]
            }, {
               "Effect" : "Deny",
               "Action" : [ "sqs:*" ],
               "NotResource" : [ {
                  "Fn::GetAtt" : [ "myqueue", "Arn" ]
               } ]
            }
         ] }
      }, {
         "PolicyName" : "giveaccesstotopiconly",
         "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement" : [ {
               "Effect" : "Allow",
               "Action" : [ "sns:*" ],
               "Resource" : [ { "Ref" : "mytopic" } ]
            }, {
               "Effect" : "Deny",
               "Action" : [ "sns:*" ],
               "NotResource" : [ { "Ref" : "mytopic" } ]
            } ]
         }
```

```
      } ]
   }
}
```

YAML

```
myuser:
  Type: AWS::IAM::User
  Properties:
    Path: "/"
    LoginProfile:
      Password: myP@ssW0rd
    Policies:
    - PolicyName: giveaccesstoqueueonly
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Effect: Allow
          Action:
          - sqs:*
          Resource:
          - !GetAtt myqueue.Arn
        - Effect: Deny
          Action:
          - sqs:*
          NotResource:
          - !GetAtt myqueue.Arn
    - PolicyName: giveaccesstotopiconly
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Effect: Allow
          Action:
          - sns:*
          Resource:
          - !Ref mytopic
        - Effect: Deny
          Action:
          - sns:*
          NotResource:
          - !Ref mytopic
```

Deklarieren einer IAM-Zugriffsschlüssel-Ressource

Dieses Codefragment zeigt eine AWS::IAM::AccessKey-Ressource. Die `myaccesskey`-Ressource erstellt einen Zugriffsschlüssel und weist ihn einem IAM-Benutzer zu, der in der Vorlage als eine AWS::IAM::User-Ressource deklariert ist.

JSON

```
"myaccesskey" : {
   "Type" : "AWS::IAM::AccessKey",
   "Properties" : {
      "UserName" : { "Ref" : "myuser" }
   }
}
```

YAML

```
myaccesskey:
  Type: AWS::IAM::AccessKey
  Properties:
    UserName:
      !Ref myuser
```

Sie können den geheimen Schlüssel für eine `AWS::IAM::AccessKey`-Ressource mithilfe der `Fn::GetAtt`-Funktion abrufen. Eine Möglichkeit zum Abrufen des geheimen Schlüssels besteht darin, ihn in einen `Output`-Wert zu setzen. Sie können den Zugriffsschlüssel mit der `Ref`-Funktion abrufen. Die folgenden `Output`-Wertdeklarationen rufen den Zugriffsschlüssel und den geheimen Schlüssel für `myaccesskey` ab.

JSON

```
"AccessKeyformyaccesskey" : {
   "Value" : { "Ref" : "myaccesskey" }
},
"SecretKeyformyaccesskey" : {
   "Value" : {
      "Fn::GetAtt" : [ "myaccesskey", "SecretAccessKey" ]
   }
}
```

YAML

```
AccessKeyformyaccesskey:
  Value:
    !Ref myaccesskey
SecretKeyformyaccesskey:
  Value: !GetAtt myaccesskey.SecretAccessKey
```

Sie können den AWS Zugriffsschlüssel und den geheimen Schlüssel auch an eine EC2 Amazon-Instance oder Auto Scaling Scaling-Gruppe übergeben, die in der Vorlage definiert sind. Die folgende AWS::EC2::Instance-Deklaration verwendet die `UserData`-Eigenschaft zum Übergeben des Zugriffsschlüssels und geheimen Schlüssels für die `myaccesskey` Ressource.

JSON

```
"myinstance" : {
   "Type" : "AWS::EC2::Instance",
   "Properties" : {
      "AvailabilityZone" : "us-east-1a",
      "ImageId" : "ami-0ff8a91507f77f867",
      "UserData" : {
         "Fn::Base64" : {
            "Fn::Join" : [
               "", [
                  "ACCESS_KEY=", {
                     "Ref" : "myaccesskey"
                  },
                  "&",
                  "SECRET_KEY=",
                  {
                     "Fn::GetAtt" : [
                        "myaccesskey",
                        "SecretAccessKey"
                     ]
                  }
               ]
            ]
         }
      }
   }
}
```

YAML

```
myinstance:
  Type: AWS::EC2::Instance
  Properties:
    AvailabilityZone: "us-east-1a"
    ImageId: ami-0ff8a91507f77f867
    UserData:
      Fn::Base64: !Sub "ACCESS_KEY=${myaccesskey}&SECRET_KEY=
${myaccesskey.SecretAccessKey}"
```

Deklarieren einer IAM-Gruppenressource

Dieses Codefragment zeigt eine AWS::IAM::Group-Ressource. Die Gruppe hat einen Pfad (`"/myapplication/"`). Das Richtliniendokument namens `myapppolicy` wird der Gruppe hinzugefügt, um den Benutzern der Gruppe die Berechtigung zum Ausführen aller Amazon SQS-Aktionen auf der Amazon SQS-Warteschlangenressource "myqueue" zu erteilen und den Zugriff auf alle anderen Amazon SQS-Ressourcen außer `myqueue` zu verweigern.

Zum Zuweisen einer Richtlinie zu einer Ressource benötigt IAM den Amazon-Ressourcennamen (ARN) für die Ressource. In dem Codebeispiel ruft die `Fn::GetAtt`-Funktion den ARN der AWS::SQS::Queue-Ressourcenwarteschlange ab.

JSON

```
"mygroup" : {
   "Type" : "AWS::IAM::Group",
   "Properties" : {
      "Path" : "/myapplication/",
      "Policies" : [ {
         "PolicyName" : "myapppolicy",
         "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement" : [ {
               "Effect" : "Allow",
               "Action" : [ "sqs:*" ],
               "Resource" : [ {
                  "Fn::GetAtt" : [ "myqueue", "Arn" ]
               } ]
            },
            {
```

```
                "Effect" : "Deny",
                "Action" : [ "sqs:*" ],
                "NotResource" : [ { "Fn::GetAtt" : [ "myqueue", "Arn" ] } ]
            }
         ] }
      } ]
   }
}
```

YAML

```
mygroup:
  Type: AWS::IAM::Group
  Properties:
    Path: "/myapplication/"
    Policies:
    - PolicyName: myapppolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Effect: Allow
          Action:
          - sqs:*
          Resource: !GetAtt myqueue.Arn
        - Effect: Deny
          Action:
          - sqs:*
          NotResource: !GetAtt myqueue.Arn
```

Hinzufügen von Benutzern zu einer Gruppe

Die AWS::IAM::UserToGroupAddition-Ressource fügt Benutzer zu einer Gruppe hinzu. Im folgenden Codeausschnitt fügt die `addUserToGroup`-Ressource die folgenden Benutzer zu einer bestehenden Gruppe mit dem Namen `myexistinggroup2` hinzu: den vorhandenen Benutzer `existinguser1` und den Benutzer `myuser`, der in der Vorlage als AWS::IAM::User-Ressource deklariert ist.

JSON

```
"addUserToGroup" : {
   "Type" : "AWS::IAM::UserToGroupAddition",
   "Properties" : {
      "GroupName" : "myexistinggroup2",
```

```
      "Users" : [ "existinguser1", { "Ref" : "myuser" } ]
   }
}
```

YAML

```
addUserToGroup:
  Type: AWS::IAM::UserToGroupAddition
  Properties:
    GroupName: myexistinggroup2
    Users:
    - existinguser1
    - !Ref myuser
```

Deklarieren einer IAM-Richtlinie

Im folgenden Codeausschnitt wird gezeigt, wie Sie eine Richtlinie erstellen und unter Verwendung einer AWS::IAM::Policy-Ressource mit dem Namen `mypolicy` auf mehreren Gruppen anwenden. Die `mypolicy`-Ressource enthält eine `PolicyDocument`-Eigenschaft, die `GetObject`-, `PutObject`- und `PutObjectAcl`-Aktionen für die Objekte in dem durch den ARN `arn:aws:s3:::myAWSBucket` repräsentierten S3-Bucket zulässt. Die `mypolicy`-Ressource wendet die Richtlinie auf eine vorhandene Gruppe mit dem Namen `myexistinggroup1` und eine Gruppe `mygroup` an, die in der Vorlage als AWS::IAM::Group-Ressource deklariert ist. Dieses Beispiel zeigt, wie eine Richtlinie mithilfe der `Groups`-Eigenschaft auf eine Gruppe angewendet wird. Sie können jedoch auch die `Users`-Eigenschaft verwenden, um einer Liste von Benutzern ein Richtliniendokument hinzuzufügen.

JSON

```
"mypolicy" : {
   "Type" : "AWS::IAM::Policy",
   "Properties" : {
      "PolicyName" : "mygrouppolicy",
      "PolicyDocument" : {
         "Version": "2012-10-17",
         "Statement" : [ {
            "Effect" : "Allow",
            "Action" : [
               "s3:GetObject" , "s3:PutObject" , "s3:PutObjectAcl" ],
            "Resource" : "arn:aws:s3:::myAWSBucket/*"
         } ]
```

```
      },
      "Groups" : [ "myexistinggroup1", { "Ref" : "mygroup" } ]
   }
}
```

YAML

```
mypolicy:
  Type: AWS::IAM::Policy
  Properties:
    PolicyName: mygrouppolicy
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
      - Effect: Allow
        Action:
        - s3:GetObject
        - s3:PutObject
        - s3:PutObjectAcl
        Resource: arn:aws:s3:::myAWSBucket/*
    Groups:
    - myexistinggroup1
    - !Ref mygroup
```

Deklarieren einer Amazon S3-Bucket-Richtlinie

Im folgenden Codeausschnitt wird gezeigt, wie Sie eine Richtlinie erstellen und unter Verwendung der AWS::S3::BucketPolicy-Ressource auf einen Amazon S3-Bucket anwenden. Die `mybucketpolicy`-Ressource deklariert ein Richtliniendokument, das dem `user1`-Benutzer IAM erlaubt, die `GetObject`-Aktion auf alle Objekte in dem S3-Bucket auszuführen, auf den die Richtlinie angewendet wird. In dem Codebeispiel ruft die `Fn::GetAtt`-Funktion den ARN der `user1`-Ressource ab. Die `mybucketpolicy`-Ressource wendet die Richtlinie auf die `AWS::S3::BucketPolicy`-Ressource mybucket an. Die `Ref`-Funktion ruft den Bucket-Namen der `mybucket`-Ressource ab.

JSON

```
"mybucketpolicy" : {
   "Type" : "AWS::S3::BucketPolicy",
   "Properties" : {
      "PolicyDocument" : {
```

```
         "Id" : "MyPolicy",
         "Version": "2012-10-17",
         "Statement" : [ {
            "Sid" : "ReadAccess",
            "Action" : [ "s3:GetObject" ],
            "Effect" : "Allow",
            "Resource" : { "Fn::Join" : [
                  "", [ "arn:aws:s3:::", { "Ref" : "mybucket" } , "/*" ]
               ] },
            "Principal" : {
               "AWS" : { "Fn::GetAtt" : [ "user1", "Arn" ] }
            }
         } ]
      },
      "Bucket" : { "Ref" : "mybucket" }
   }
}
```

YAML

```
mybucketpolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    PolicyDocument:
      Id: MyPolicy
      Version: '2012-10-17'
      Statement:
      - Sid: ReadAccess
        Action:
        - s3:GetObject
        Effect: Allow
        Resource: !Sub "arn:aws:s3:::${mybucket}/*"
        Principal:
          AWS: !GetAtt user1.Arn
    Bucket: !Ref mybucket
```

Deklarieren einer Amazon SNS-Themenrichtlinie

Im folgenden Codeausschnitt wird gezeigt, wie Sie eine Richtlinie erstellen und unter Verwendung der AWS::SNS::TopicPolicy-Ressource auf ein Amazon SNS-Thema anwenden. Die `mysnspolicy`-Ressource enthält eine `PolicyDocument`-Eigenschaft, die der AWS::IAM::User-Ressource `myuser` die Ausführung der `Publish`-Aktion auf eine AWS::SNS::Topic-Ressource `mytopic` erlaubt. In dem

Codebeispiel ruft die `Fn::GetAtt`-Funktion den ARN für die `myuser`-Ressource ab und die `Ref`-Funktion ruft den ARN für die `mytopic`-Ressource ab.

JSON

```
"mysnspolicy" : {
   "Type" : "AWS::SNS::TopicPolicy",
   "Properties" : {
      "PolicyDocument" :  {
         "Id" : "MyTopicPolicy",
         "Version" : "2012-10-17",
         "Statement" : [ {
            "Sid" : "My-statement-id",
            "Effect" : "Allow",
            "Principal" : {
               "AWS" : { "Fn::GetAtt" : [ "myuser", "Arn" ] }
            },
            "Action" : "sns:Publish",
            "Resource" : "*"
         } ]
      },
      "Topics" : [ { "Ref" : "mytopic" } ]
   }
}
```

YAML

```
mysnspolicy:
  Type: AWS::SNS::TopicPolicy
  Properties:
    PolicyDocument:
      Id: MyTopicPolicy
      Version: '2012-10-17'
      Statement:
      - Sid: My-statement-id
        Effect: Allow
        Principal:
          AWS: !GetAtt myuser.Arn
        Action: sns:Publish
        Resource: "*"
    Topics:
    - !Ref mytopic
```

Deklarieren einer Amazon SQS-Richtlinie

Im folgenden Codeausschnitt wird gezeigt, wie Sie eine Richtlinie erstellen und unter Verwendung der AWS::SQS::QueuePolicy-Ressource auf eine Amazon SQS-Warteschlange anwenden. Die `PolicyDocument`-Eigenschaft ermöglicht dem vorhandenen Benutzer `myapp` (angegeben durch seinen ARN) die Ausführung der `SendMessage`-Aktion für eine vorhandene Warteschlange, die durch die URL angegeben ist, und eine AWS::SQS::Queue-Ressource myqueue. Die Ref-Funktion ruft die URL für die `myqueue`-Ressource ab.

JSON

```
"mysqspolicy" : {
   "Type" : "AWS::SQS::QueuePolicy",
   "Properties" : {
      "PolicyDocument" : {
         "Id" : "MyQueuePolicy",
         "Version" : "2012-10-17",
         "Statement" : [ {
            "Sid" : "Allow-User-SendMessage",
            "Effect" : "Allow",
            "Principal" : {
               "AWS" : "arn:aws:iam::123456789012:user/myapp"
            },
            "Action" : [ "sqs:SendMessage" ],
            "Resource" : "*"
         } ]
      },
      "Queues" : [
         "https://sqs.us-east-2aws-region.amazonaws.com/123456789012/myexistingqueue",
         { "Ref" : "myqueue" }
      ]
   }
}
```

YAML

```
mysqspolicy:
  Type: AWS::SQS::QueuePolicy
  Properties:
    PolicyDocument:
      Id: MyQueuePolicy
      Version: '2012-10-17'
```

```
      Statement:
      - Sid: Allow-User-SendMessage
        Effect: Allow
        Principal:
          AWS: arn:aws:iam::123456789012:user/myapp
        Action:
        - sqs:SendMessage
        Resource: "*"
    Queues:
    - https://sqs.aws-region.amazonaws.com/123456789012/myexistingqueue
    - !Ref myqueue
```

Vorlagenbeispiele für IAM-Rollen

Dieser Abschnitt enthält CloudFormation Vorlagenbeispiele für IAM-Rollen für EC2 Instances.

Weitere Informationen finden Sie unter IAM-Rollen für Amazon EC2 im EC2 Amazon-Benutzerhandbuch.

IAM-Rolle bei EC2

In diesem Beispiel wird das Instanzprofil durch die `IamInstanceProfile` Eigenschaft der EC2 Instanz referenziert. Sowohl die Instance-Richtlinie als auch die Rollenrichtlinie referenzieren AWS::IAM::Role.

JSON

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
      "myEC2Instance": {
         "Type": "AWS::EC2::Instance",
         "Version": "2009-05-15",
         "Properties": {
            "ImageId": "ami-0ff8a91507f77f867",
            "InstanceType": "m1.small",
            "Monitoring": "true",
            "DisableApiTermination": "false",
            "IamInstanceProfile": {
               "Ref": "RootInstanceProfile"
            }
         }
      },
```

```
      "RootRole": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/"
         }
      },
      "RolePolicies": {
         "Type": "AWS::IAM::Policy",
         "Properties": {
            "PolicyName": "root",
            "PolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Action": "*",
                  "Resource": "*"
               } ]
            },
            "Roles": [ { "Ref": "RootRole" } ]
         }
      },
      "RootInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ { "Ref": "RootRole" } ]
         }
      }
   }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myEC2Instance:
    Type: AWS::EC2::Instance
    Version: '2009-05-15'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: m1.small
      Monitoring: 'true'
      DisableApiTermination: 'false'
      IamInstanceProfile:
        !Ref RootInstanceProfile
  RootRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Effect: Allow
          Principal:
            Service:
            - ec2.amazonaws.com
          Action:
          - sts:AssumeRole
      Path: "/"
  RolePolicies:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: root
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Effect: Allow
          Action: "*"
          Resource: "*"
      Roles:
      - !Ref RootRole
  RootInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: "/"
      Roles:
```

```
      - !Ref RootRole
```

IAM-Rolle mit Auto Scaling Scaling-Gruppe

In diesem Beispiel wird das Instance-Profil durch die `IamInstanceProfile` Eigenschaft einer Amazon EC2 Auto Scaling Scaling-Startkonfiguration referenziert.

JSON

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
      "myLCOne": {
         "Type": "AWS::AutoScaling::LaunchConfiguration",
         "Version": "2009-05-15",
         "Properties": {
            "ImageId": "ami-0ff8a91507f77f867",
            "InstanceType": "m1.small",
            "InstanceMonitoring": "true",
            "IamInstanceProfile": { "Ref": "RootInstanceProfile" }
         }
      },
      "myASGrpOne": {
         "Type": "AWS::AutoScaling::AutoScalingGroup",
         "Version": "2009-05-15",
         "Properties": {
            "AvailabilityZones": [ "us-east-1a" ],
            "LaunchConfigurationName": { "Ref": "myLCOne" },
            "MinSize": "0",
            "MaxSize": "0",
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": "120"
         }
      },
      "RootRole": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
```

```
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/"
         }
      },
      "RolePolicies": {
         "Type": "AWS::IAM::Policy",
         "Properties": {
            "PolicyName": "root",
            "PolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Action": "*",
                  "Resource": "*"
               } ]
            },
            "Roles": [ { "Ref": "RootRole" } ]
         }
      },
      "RootInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ { "Ref": "RootRole" } ]
         }
      }
   }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myLCOne:
    Type: AWS::AutoScaling::LaunchConfiguration
    Version: '2009-05-15'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: m1.small
      InstanceMonitoring: 'true'
```

```
      IamInstanceProfile:
        !Ref RootInstanceProfile
  myASGrpOne:
    Type: AWS::AutoScaling::AutoScalingGroup
    Version: '2009-05-15'
    Properties:
      AvailabilityZones:
      - "us-east-1a"
      LaunchConfigurationName:
        !Ref myLCOne
      MinSize: '0'
      MaxSize: '0'
      HealthCheckType: EC2
      HealthCheckGracePeriod: '120'
  RootRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Effect: Allow
          Principal:
            Service:
            - ec2.amazonaws.com
          Action:
          - sts:AssumeRole
      Path: "/"
  RolePolicies:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: root
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Effect: Allow
          Action: "*"
          Resource: "*"
      Roles:
      - !Ref RootRole
  RootInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: "/"
      Roles:
```

```
      - !Ref RootRole
```

AWS Lambda Vorlage

Die folgende Vorlage verwendet eine AWS Lambda (Lambda-) Funktion und eine benutzerdefinierte Ressource, um eine neue Sicherheitsgruppe an eine Liste vorhandener Sicherheitsgruppen anzuhängen. Diese Funktion ist hilfreich, wenn Sie dynamisch eine Liste von Sicherheitsgruppen erstellen möchten, sodass die Liste sowohl neue als auch vorhandene Sicherheitsgruppen enthält. Sie können beispielsweise eine Liste vorhandener Sicherheitsgruppen als Parameterwert übergeben, den neuen Wert an die Liste anhängen und dann alle Ihre Werte einer Instanz zuordnen. EC2 Weitere Hinweise zum Ressourcentyp der Lambda-Funktion finden Sie unter AWS::Lambda::Function.

In diesem Beispiel CloudFormation wird beim Erstellen der `AllSecurityGroups` benutzerdefinierten Ressource die `AppendItemToListFunction` Lambda-Funktion CloudFormation aufgerufen. CloudFormation übergibt die Liste der vorhandenen Sicherheitsgruppen und eine neue Sicherheitsgruppe (`NewSecurityGroup`) an die Funktion, die die neue Sicherheitsgruppe an die Liste anhängt und dann die geänderte Liste zurückgibt. CloudFormation verwendet die geänderte Liste, um der `MyEC2Instance` Ressource alle Sicherheitsgruppen zuzuordnen.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Parameters": {
        "ExistingSecurityGroups": {
            "Type": "List<AWS::EC2::SecurityGroup::Id>"
        },
        "ExistingVPC": {
            "Type": "AWS::EC2::VPC::Id",
            "Description": "The VPC ID that includes the security groups in the
 ExistingSecurityGroups parameter."
        },
        "InstanceType": {
            "Type": "String",
            "Default": "t2.micro",
            "AllowedValues": [
                "t2.micro",
                "m1.small"
            ]
```

```
        }
    },
    "Mappings": {
        "AWSInstanceType2Arch": {
            "t2.micro": {
                "Arch": "HVM64"
            },
            "m1.small": {
                "Arch": "HVM64"
            }
        },
        "AWSRegionArch2AMI": {
            "us-east-1": {
                "HVM64": "ami-0ff8a91507f77f867",
                "HVMG2": "ami-0a584ac55a7631c0c"
            },
            "us-west-2": {
                "HVM64": "ami-a0cfeed8",
                "HVMG2": "ami-0e09505bc235aa82d"
            },
            "us-west-1": {
                "HVM64": "ami-0bdb828fd58c52235",
                "HVMG2": "ami-066ee5fd4a9ef77f1"
            },
            "eu-west-1": {
                "HVM64": "ami-047bb4163c506cd98",
                "HVMG2": "ami-0a7c483d527806435"
            },
            "eu-central-1": {
                "HVM64": "ami-0233214e13e500f77",
                "HVMG2": "ami-06223d46a6d0661c7"
            },
            "ap-northeast-1": {
                "HVM64": "ami-06cd52961ce9f0d85",
                "HVMG2": "ami-053cdd503598e4a9d"
            },
            "ap-southeast-1": {
                "HVM64": "ami-08569b978cc4dfa10",
                "HVMG2": "ami-0be9df32ae9f92309"
            },
            "ap-southeast-2": {
                "HVM64": "ami-09b42976632b27e9b",
                "HVMG2": "ami-0a9ce9fecc3d1daf8"
            },
```

```
            "sa-east-1": {
                "HVM64": "ami-07b14488da8ea02a0",
                "HVMG2": "NOT_SUPPORTED"
            },
            "cn-north-1": {
                "HVM64": "ami-0a4eaf6c4454eda75",
                "HVMG2": "NOT_SUPPORTED"
            }
        }
    },
    "Resources": {
        "SecurityGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "Allow HTTP traffic to the host",
                "VpcId": {
                    "Ref": "ExistingVPC"
                },
                "SecurityGroupIngress": [
                    {
                        "IpProtocol": "tcp",
                        "FromPort": 80,
                        "ToPort": 80,
                        "CidrIp": "0.0.0.0/0"
                    }
                ],
                "SecurityGroupEgress": [
                    {
                        "IpProtocol": "tcp",
                        "FromPort": 80,
                        "ToPort": 80,
                        "CidrIp": "0.0.0.0/0"
                    }
                ]
            }
        },
        "AllSecurityGroups": {
            "Type": "Custom::Split",
            "Properties": {
                "ServiceToken": {
                    "Fn::GetAtt": [
                        "AppendItemToListFunction",
                        "Arn"
                    ]
```

```
                },
                "List": {
                    "Ref": "ExistingSecurityGroups"
                },
                "AppendedItem": {
                    "Ref": "SecurityGroup"
                }
            }
        },
        "AppendItemToListFunction": {
            "Type": "AWS::Lambda::Function",
            "Properties": {
                "Handler": "index.handler",
                "Role": {
                    "Fn::GetAtt": [
                        "LambdaExecutionRole",
                        "Arn"
                    ]
                },
                "Code": {
                    "ZipFile": {
                        "Fn::Join": [
                            "",
                            [
                                "var response = require('cfn-response');",
                                "exports.handler = function(event, context) {",
                                "   var responseData = {Value:
event.ResourceProperties.List};",
                                "
responseData.Value.push(event.ResourceProperties.AppendedItem);",
                                "   response.send(event, context, response.SUCCESS,
responseData);",
                                "};"
                            ]
                        ]
                    }
                },
                "Runtime": "nodejs20.x"
            }
        },
        "MyEC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId": {
```

```
                    "Fn::FindInMap": [
                        "AWSRegionArch2AMI",
                        {
                            "Ref": "AWS::Region"
                        },
                        {
                            "Fn::FindInMap": [
                                "AWSInstanceType2Arch",
                                {
                                    "Ref": "InstanceType"
                                },
                                "Arch"
                            ]
                        }
                    ]
                },
                "SecurityGroupIds": {
                    "Fn::GetAtt": [
                        "AllSecurityGroups",
                        "Value"
                    ]
                },
                "InstanceType": {
                    "Ref": "InstanceType"
                }
            }
        },
        "LambdaExecutionRole": {
            "Type": "AWS::IAM::Role",
            "Properties": {
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": [
                                    "lambda.amazonaws.com"
                                ]
                            },
                            "Action": [
                                "sts:AssumeRole"
                            ]
                        }
```

```
                    ]
                },
                "Path": "/",
                "Policies": [
                    {
                        "PolicyName": "root",
                        "PolicyDocument": {
                            "Version": "2012-10-17",
                            "Statement": [
                                {
                                    "Effect": "Allow",
                                    "Action": [
                                        "logs:*"
                                    ],
                                    "Resource": "arn:aws:logs:*:*:*"
                                }
                            ]
                        }
                    }
                ]
            }
        }
    },
    "Outputs": {
        "AllSecurityGroups": {
            "Description": "Security Groups that are associated with the EC2 instance",
            "Value": {
                "Fn::Join": [
                    ", ",
                    {
                        "Fn::GetAtt": [
                            "AllSecurityGroups",
                            "Value"
                        ]
                    }
                ]
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  ExistingSecurityGroups:
    Type: List<AWS::EC2::SecurityGroup::Id>
  ExistingVPC:
    Type: AWS::EC2::VPC::Id
    Description: The VPC ID that includes the security groups in the
 ExistingSecurityGroups parameter.
  InstanceType:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - m1.small
Mappings:
  AWSInstanceType2Arch:
    t2.micro:
      Arch: HVM64
    m1.small:
      Arch: HVM64
  AWSRegionArch2AMI:
    us-east-1:
      HVM64: ami-0ff8a91507f77f867
      HVMG2: ami-0a584ac55a7631c0c
    us-west-2:
      HVM64: ami-a0cfeed8
      HVMG2: ami-0e09505bc235aa82d
    us-west-1:
      HVM64: ami-0bdb828fd58c52235
      HVMG2: ami-066ee5fd4a9ef77f1
    eu-west-1:
      HVM64: ami-047bb4163c506cd98
      HVMG2: ami-0a7c483d527806435
    eu-central-1:
      HVM64: ami-0233214e13e500f77
      HVMG2: ami-06223d46a6d0661c7
    ap-northeast-1:
      HVM64: ami-06cd52961ce9f0d85
      HVMG2: ami-053cdd503598e4a9d
    ap-southeast-1:
      HVM64: ami-08569b978cc4dfa10
```

```
      HVMG2: ami-0be9df32ae9f92309
    ap-southeast-2:
      HVM64: ami-09b42976632b27e9b
      HVMG2: ami-0a9ce9fecc3d1daf8
    sa-east-1:
      HVM64: ami-07b14488da8ea02a0
      HVMG2: NOT_SUPPORTED
    cn-north-1:
      HVM64: ami-0a4eaf6c4454eda75
      HVMG2: NOT_SUPPORTED
Resources:
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow HTTP traffic to the host
      VpcId: !Ref ExistingVPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
      SecurityGroupEgress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
  AllSecurityGroups:
    Type: Custom::Split
    Properties:
      ServiceToken: !GetAtt AppendItemToListFunction.Arn
      List: !Ref ExistingSecurityGroups
      AppendedItem: !Ref SecurityGroup
  AppendItemToListFunction:
    Type: AWS::Lambda::Function
    Properties:
      Handler: index.handler
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile: !Join
          - ''
          - - var response = require('cfn-response');
            - exports.handler = function(event, context) {
            - '   var responseData = {Value: event.ResourceProperties.List};'
            - '   responseData.Value.push(event.ResourceProperties.AppendedItem);'
```

```
            - '   response.send(event, context, response.SUCCESS, responseData);'
            - '};'
      Runtime: nodejs20.x
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !FindInMap
        - AWSRegionArch2AMI
        - !Ref AWS::Region
        - !FindInMap
          - AWSInstanceType2Arch
          - !Ref InstanceType
          - Arch
      SecurityGroupIds: !GetAtt AllSecurityGroups.Value
      InstanceType: !Ref InstanceType
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - lambda.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: /
      Policies:
        - PolicyName: root
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
                  - logs:*
                Resource: arn:aws:logs:*:*:*
Outputs:
  AllSecurityGroups:
    Description: Security Groups that are associated with the EC2 instance
    Value: !Join
      - ', '
      - !GetAtt AllSecurityGroups.Value
```

Amazon Redshift-Vorlagenausschnitte

Amazon Redshift ist ein vollständig verwalteter Data-Warehouse-Service in Petabytegröße in der Cloud. Sie können mit Amazon-Redshift-Cluster von AWS CloudFormation bereitstellen und verwalten.

Amazon-Redshift-Cluster

Die folgende Beispielvorlage erstellt einen Amazon Redshift-Cluster entsprechend den Parameterwerten, die bei der Stack-Erstellung angegeben werden. Die Cluster-Parametergruppe, die dem Amazon Redshift-Cluster zugeordnet ist, ermöglicht die Protokollierung von Benutzeraktivitäten. Die Vorlage startet zudem die Amazon Redshift-Cluster in einer Amazon VPC, die in der Vorlage definiert ist. Die VPC enthält ein Internet-Gateway, sodass Sie aus dem Internet auf die Amazon Redshift-Cluster zugreifen können. Die Kommunikation zwischen dem Cluster und dem Internet-Gateway muss jedoch auch aktiviert werden, was durch den Eintrag in die Routing-Tabelle erfolgt.

> **Note**
>
> Die Vorlage enthält die Bedingung `IsMultiNodeCluster`, sodass der Parameter `NumberOfNodes` nur deklariert wird, wenn dem Parameter `ClusterType` der Wert `multi-node` zugewiesen wird.

Das Beispiel definiert den Parameter `MysqlRootPassword` mit auf `true` gesetzter Eigenschaft `NoEcho`. Wenn Sie das `NoEcho`-Attribut auf `true` festlegen, gibt CloudFormation für alle Aufrufe, die den Stack oder die Stapelereignisse beschreiben, den als Sternchen (*****) maskierten Parameterwert zurück, mit Ausnahme der Informationen, die an den folgenden Speicherorten gespeichert sind.

> **Important**
>
> Durch die Verwendung des `NoEcho`-Attributs werden keine Informationen maskiert, die im Folgenden gespeichert sind:
>
> - Der `Metadata` Vorlagenbereich. CloudFormation transformiert, modifiziert oder redigiert keine Informationen, die Sie in den `Metadata` Abschnitt aufnehmen. Weitere Informationen finden Sie unter Metadaten.
> - Der `Outputs`-Vorlagenabschnitt. Weitere Informationen finden Sie unter Ausgaben:

- Das `Metadata`-Attribut einer Ressourcendefinition. Weitere Informationen finden Sie unter `Metadata`-Attribut.

Es wird dringend empfohlen, diese Mechanismen nicht zu verwenden, um vertrauliche Informationen wie Passwörter oder Geheimnisse einzugeben.

Important

Anstatt vertrauliche Informationen direkt in Ihre CloudFormation Vorlagen einzubetten, empfehlen wir Ihnen, dynamische Parameter in der Stack-Vorlage zu verwenden, um auf vertrauliche Informationen zu verweisen, die außerhalb von gespeichert und verwaltet werden CloudFormation, z. B. im AWS Systems Manager Parameterspeicher oder. AWS Secrets Manager
Weitere Informationen finden Sie in den bewährten Methoden zu Keine Anmeldeinformationen in Ihre Vorlagen einbetten.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "DatabaseName" : {
      "Description" : "The name of the first database to be created when the cluster is
 created",
      "Type" : "String",
      "Default" : "dev",
      "AllowedPattern" : "([a-z]|[0-9])+"
    },
    "ClusterType" : {
      "Description" : "The type of cluster",
      "Type" : "String",
      "Default" : "single-node",
      "AllowedValues" : [ "single-node", "multi-node" ]
    },
    "NumberOfNodes" : {
      "Description" : "The number of compute nodes in the cluster. For multi-node
 clusters, the NumberOfNodes parameter must be greater than 1",
      "Type" : "Number",
```

```
      "Default" : "1"
    },
    "NodeType" : {
      "Description" : "The type of node to be provisioned",
      "Type" : "String",
      "Default" : "ds2.xlarge",
      "AllowedValues" : [ "ds2.xlarge", "ds2.8xlarge", "dc1.large", "dc1.8xlarge" ]
    },
    "MasterUsername" : {
      "Description" : "The user name that is associated with the master user account
 for the cluster that is being created",
      "Type" : "String",
      "Default" : "defaultuser",
      "AllowedPattern" : "([a-z])([a-z]|[0-9])*"
    },
    "MasterUserPassword" :  {
      "Description" : "The password that is associated with the master user account for
 the cluster that is being created.",
      "Type" : "String",
      "NoEcho" : "true"
    },
    "InboundTraffic" : {
      "Description" : "Allow inbound traffic to the cluster from this CIDR range.",
      "Type" : "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default" : "0.0.0.0/0",
      "AllowedPattern" : "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\
\d{1,2})",
      "ConstraintDescription" : "must be a valid CIDR range of the form x.x.x.x/x."
    },
    "PortNumber" : {
      "Description" : "The port number on which the cluster accepts incoming
 connections.",
      "Type" : "Number",
      "Default" : "5439"
    }
  },
  "Conditions" : {
    "IsMultiNodeCluster" : {
      "Fn::Equals" : [{ "Ref" : "ClusterType" }, "multi-node" ]
    }
  },
  "Resources" : {
```

```
    "RedshiftCluster" : {
      "Type" : "AWS::Redshift::Cluster",
      "DependsOn" : "AttachGateway",
      "Properties" : {
        "ClusterType" : { "Ref" : "ClusterType" },
        "NumberOfNodes" : { "Fn::If" : [ "IsMultiNodeCluster",  { "Ref" :
 "NumberOfNodes" }, { "Ref" : "AWS::NoValue" }]},
        "NodeType" : { "Ref" : "NodeType" },
        "DBName" : { "Ref" : "DatabaseName" },
        "MasterUsername" : { "Ref" : "MasterUsername" },
        "MasterUserPassword" : { "Ref" : "MasterUserPassword" },
        "ClusterParameterGroupName" : { "Ref" : "RedshiftClusterParameterGroup" },
        "VpcSecurityGroupIds" : [ { "Ref" : "SecurityGroup" } ],
        "ClusterSubnetGroupName" : { "Ref" : "RedshiftClusterSubnetGroup" },
        "PubliclyAccessible" : "true",
        "Port" : { "Ref" : "PortNumber" }
      }
    },
    "RedshiftClusterParameterGroup" : {
      "Type" : "AWS::Redshift::ClusterParameterGroup",
      "Properties" : {
        "Description" : "Cluster parameter group",
        "ParameterGroupFamily" : "redshift-1.0",
        "Parameters" : [{
          "ParameterName" : "enable_user_activity_logging",
          "ParameterValue" : "true"
        }]
      }
    },
    "RedshiftClusterSubnetGroup" : {
      "Type" : "AWS::Redshift::ClusterSubnetGroup",
      "Properties" : {
        "Description" : "Cluster subnet group",
        "SubnetIds" : [ { "Ref" : "PublicSubnet" } ]
      }
    },
    "VPC" : {
      "Type" : "AWS::EC2::VPC",
      "Properties" : {
        "CidrBlock" : "10.0.0.0/16"
      }
    },
    "PublicSubnet" : {
      "Type" : "AWS::EC2::Subnet",
```

```
      "Properties" : {
        "CidrBlock" : "10.0.0.0/24",
        "VpcId" : { "Ref" : "VPC" }
      }
    },
    "SecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Security group",
        "SecurityGroupIngress" : [ {
          "CidrIp" : { "Ref": "InboundTraffic" },
          "FromPort" : { "Ref" : "PortNumber" },
          "ToPort" : { "Ref" : "PortNumber" },
          "IpProtocol" : "tcp"
        } ],
        "VpcId" : { "Ref" : "VPC" }
      }
    },
    "myInternetGateway" : {
      "Type" : "AWS::EC2::InternetGateway"
    },
    "AttachGateway" : {
      "Type" : "AWS::EC2::VPCGatewayAttachment",
      "Properties" : {
        "VpcId" : { "Ref" : "VPC" },
        "InternetGatewayId" : { "Ref" : "myInternetGateway" }
      }
    },
    "PublicRouteTable" : {
      "Type" : "AWS::EC2::RouteTable",
      "Properties" : {
        "VpcId" : {
          "Ref" : "VPC"
        }
      }
    },
    "PublicRoute" : {
      "Type" : "AWS::EC2::Route",
      "DependsOn" : "AttachGateway",
      "Properties"  : {
        "RouteTableId" : {
          "Ref" : "PublicRouteTable"
        },
        "DestinationCidrBlock" : "0.0.0.0/0",
```

```
        "GatewayId" : {
          "Ref" : "myInternetGateway"
        }
      }
    },
    "PublicSubnetRouteTableAssociation" : {
      "Type" : "AWS::EC2::SubnetRouteTableAssociation",
      "Properties" : {
        "SubnetId" : {
          "Ref" : "PublicSubnet"
        },
        "RouteTableId" : {
          "Ref" : "PublicRouteTable"
        }
      }
    }
  },
  "Outputs" : {
    "ClusterEndpoint" : {
      "Description" : "Cluster endpoint",
      "Value" : { "Fn::Join" : [ ":", [ { "Fn::GetAtt" : [ "RedshiftCluster",
 "Endpoint.Address" ] }, { "Fn::GetAtt" : [ "RedshiftCluster",
 "Endpoint.Port" ] } ] ] }
    },
    "ClusterName" : {
      "Description" : "Name of cluster",
      "Value" : { "Ref" : "RedshiftCluster" }
    },
    "ParameterGroupName" : {
      "Description" : "Name of parameter group",
      "Value" : { "Ref" : "RedshiftClusterParameterGroup" }
    },
    "RedshiftClusterSubnetGroupName" : {
      "Description" : "Name of cluster subnet group",
      "Value" : { "Ref" : "RedshiftClusterSubnetGroup" }
    },
    "RedshiftClusterSecurityGroupName" : {
      "Description" : "Name of cluster security group",
      "Value" : { "Ref" : "SecurityGroup" }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  DatabaseName:
    Description: The name of the first database to be created when the cluster is
      created
    Type: String
    Default: dev
    AllowedPattern: "([a-z]|[0-9])+"
  ClusterType:
    Description: The type of cluster
    Type: String
    Default: single-node
    AllowedValues:
    - single-node
    - multi-node
  NumberOfNodes:
    Description: The number of compute nodes in the cluster. For multi-node clusters,
      the NumberOfNodes parameter must be greater than 1
    Type: Number
    Default: '1'
  NodeType:
    Description: The type of node to be provisioned
    Type: String
    Default: ds2.xlarge
    AllowedValues:
    - ds2.xlarge
    - ds2.8xlarge
    - dc1.large
    - dc1.8xlarge
  MasterUsername:
    Description: The user name that is associated with the master user account for
      the cluster that is being created
    Type: String
    Default: defaultuser
    AllowedPattern: "([a-z])([a-z]|[0-9])*"
  MasterUserPassword:
    Description: The password that is associated with the master user account for
      the cluster that is being created.
    Type: String
    NoEcho: 'true'
  InboundTraffic:
    Description: Allow inbound traffic to the cluster from this CIDR range.
```

```
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})"
    ConstraintDescription: must be a valid CIDR range of the form x.x.x.x/x.
  PortNumber:
    Description: The port number on which the cluster accepts incoming connections.
    Type: Number
    Default: '5439'
Conditions:
  IsMultiNodeCluster:
    Fn::Equals:
    - Ref: ClusterType
    - multi-node
Resources:
  RedshiftCluster:
    Type: AWS::Redshift::Cluster
    DependsOn: AttachGateway
    Properties:
      ClusterType:
        Ref: ClusterType
      NumberOfNodes:
        Fn::If:
        - IsMultiNodeCluster
        - Ref: NumberOfNodes
        - Ref: AWS::NoValue
      NodeType:
        Ref: NodeType
      DBName:
        Ref: DatabaseName
      MasterUsername:
        Ref: MasterUsername
      MasterUserPassword:
        Ref: MasterUserPassword
      ClusterParameterGroupName:
        Ref: RedshiftClusterParameterGroup
      VpcSecurityGroupIds:
      - Ref: SecurityGroup
      ClusterSubnetGroupName:
        Ref: RedshiftClusterSubnetGroup
      PubliclyAccessible: 'true'
      Port:
        Ref: PortNumber
```

```
  RedshiftClusterParameterGroup:
    Type: AWS::Redshift::ClusterParameterGroup
    Properties:
      Description: Cluster parameter group
      ParameterGroupFamily: redshift-1.0
      Parameters:
      - ParameterName: enable_user_activity_logging
        ParameterValue: 'true'
  RedshiftClusterSubnetGroup:
    Type: AWS::Redshift::ClusterSubnetGroup
    Properties:
      Description: Cluster subnet group
      SubnetIds:
      - Ref: PublicSubnet
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
  PublicSubnet:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.0.0/24
      VpcId:
        Ref: VPC
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Security group
      SecurityGroupIngress:
      - CidrIp:
          Ref: InboundTraffic
        FromPort:
          Ref: PortNumber
        ToPort:
          Ref: PortNumber
        IpProtocol: tcp
      VpcId:
        Ref: VPC
  myInternetGateway:
    Type: AWS::EC2::InternetGateway
  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId:
```

```
        Ref: VPC
      InternetGatewayId:
        Ref: myInternetGateway
  PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId:
        Ref: VPC
  PublicRoute:
    Type: AWS::EC2::Route
    DependsOn: AttachGateway
    Properties:
      RouteTableId:
        Ref: PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId:
        Ref: myInternetGateway
  PublicSubnetRouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      SubnetId:
        Ref: PublicSubnet
      RouteTableId:
        Ref: PublicRouteTable
Outputs:
  ClusterEndpoint:
    Description: Cluster endpoint
    Value: !Sub "${RedshiftCluster.Endpoint.Address}:${RedshiftCluster.Endpoint.Port}"
  ClusterName:
    Description: Name of cluster
    Value:
      Ref: RedshiftCluster
  ParameterGroupName:
    Description: Name of parameter group
    Value:
      Ref: RedshiftClusterParameterGroup
  RedshiftClusterSubnetGroupName:
    Description: Name of cluster subnet group
    Value:
      Ref: RedshiftClusterSubnetGroup
  RedshiftClusterSecurityGroupName:
    Description: Name of cluster security group
    Value:
```

```
      Ref: SecurityGroup
```

Weitere Informationen finden Sie auch unter

AWS::Redshift::Cluster

Amazon RDS-Vorlagenausschnitte

Themen

- Amazon RDS-DB-Instance-Ressource
- Amazon RDS-Oracle-Datenbank-DB-Instance-Ressource
- Amazon RDS DBSecurity Group-Ressource für den CIDR-Bereich
- Amazon DBSecurity RDS-Gruppe mit einer EC2 Amazon-Sicherheitsgruppe
- Mehrere VPC-Sicherheitsgruppen
- Amazon RDS-Datenbank-Instance in einer VPC-Sicherheitsgruppe

Amazon RDS-DB-Instance-Ressource

Dieses Beispiel zeigt eine Amazon-RDS-DB-Instance-Ressource mit einem verwalteten Masterbenutzerpasswort. Weitere Informationen finden Sie unter Passwortverwaltung mit AWS Secrets Manager im Amazon-RDS-Benutzerhandbuch und Passwortverwaltung mit AWS Secrets Manager im Aurora-Benutzerhandbuch. Da die optionale `EngineVersion` Eigenschaft nicht angegeben ist, wird die Standard-Engine-Version für diese DB-Instance verwendet. Einzelheiten zur Standard-Engine-Version und anderen Standardeinstellungen finden Sie unter Erstellen DBInstance. Die `DBSecurityGroups` Eigenschaft autorisiert den Netzwerkzugriff auf die `AWS::RDS::DBSecurityGroup` Ressourcen mit dem Namen `MyDbSecurityByEC2SecurityGroup` und. MyDbSecurityBy CIDRIPGroup Details hierzu finden Sie unter AWS::RDS::DBInstance. Die DB-Instance-Ressource hat auch ein `DeletionPolicy` Attribut, das auf gesetzt ist. `Snapshot` Mit diesem `Snapshot` `DeletionPolicy` Set AWS CloudFormation wird ein Snapshot dieser DB-Instance erstellt, bevor sie beim Löschen des Stacks gelöscht wird.

JSON

```
"MyDB" : {
 "Type" : "AWS::RDS::DBInstance",
 "Properties" : {
```

```
    "DBSecurityGroups" : [
       {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" :
 "MyDbSecurityByCIDRIPGroup"} ],
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "MySQL",
    "MasterUsername" : "MyName",
    "ManageMasterUserPassword" : true,
    "MasterUserSecret" : {
       "KmsKeyId" : {"Ref" : "KMSKey"}
    }
 },
 "DeletionPolicy" : "Snapshot"
}
```

YAML

```
MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    DBSecurityGroups:
    - Ref: MyDbSecurityByEC2SecurityGroup
    - Ref: MyDbSecurityByCIDRIPGroup
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: MySQL
    MasterUsername: MyName
    ManageMasterUserPassword: true
    MasterUserSecret:
      KmsKeyId: !Ref KMSKey
  DeletionPolicy: Snapshot
```

Amazon RDS-Oracle-Datenbank-DB-Instance-Ressource

In diesem Beispiel wird eine Oracle-Database-DB-Instance-Ressource mit einem verwalteten Masterbenutzerpasswort erstellt. Weitere Informationen finden Sie unter Passwortverwaltung mit AWS Secrets Manager im Amazon-RDS-Benutzerhandbuch. Das Beispiel spezifiziert das `Engine` wie `oracle-ee` bei einem Lizenzmodell von bring-your-own-license. Einzelheiten zu den Einstellungen für Oracle-Datenbank-DB-Instances finden Sie unter Create DBInstance. Die Eigenschaft DBSecurity Groups autorisiert den Netzwerkzugriff auf die `AWS::RDS::DBSecurityGroup` Ressourcen mit dem Namen MyDbSecurityBy EC2 SecurityGroup und. MyDbSecurityBy CIDRIPGroup Details hierzu finden Sie unter AWS::RDS::DBInstance. Die

DB-Instance-Ressource hat auch ein `DeletionPolicy` Attribut, das auf gesetzt ist. `Snapshot` Mit diesem `Snapshot` `DeletionPolicy` Set AWS CloudFormation wird ein Snapshot dieser DB-Instance erstellt, bevor sie beim Löschen des Stacks gelöscht wird.

JSON

```
"MyDB" : {
 "Type" : "AWS::RDS::DBInstance",
 "Properties" : {
     "DBSecurityGroups" : [
        {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" :
 "MyDbSecurityByCIDRIPGroup"} ],
     "AllocatedStorage" : "5",
     "DBInstanceClass" : "db.t2.small",
     "Engine" : "oracle-ee",
     "LicenseModel" : "bring-your-own-license",
     "MasterUsername" : "master",
     "ManageMasterUserPassword" : true,
     "MasterUserSecret" : {
        "KmsKeyId" : {"Ref" : "KMSKey"}
     }
 },
 "DeletionPolicy" : "Snapshot"
}
```

YAML

```
MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    DBSecurityGroups:
    - Ref: MyDbSecurityByEC2SecurityGroup
    - Ref: MyDbSecurityByCIDRIPGroup
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: oracle-ee
    LicenseModel: bring-your-own-license
    MasterUsername: master
    ManageMasterUserPassword: true
    MasterUserSecret:
      KmsKeyId: !Ref KMSKey
  DeletionPolicy: Snapshot
```

Amazon RDS DBSecurity Group-Ressource für den CIDR-Bereich

Dieses Beispiel zeigt eine Amazon `DBSecurityGroup` RDS-Ressource mit Eingangsautorisierung für den angegebenen CIDR-Bereich im Format. `ddd.ddd.ddd.ddd/dd` Einzelheiten finden Sie unter AWS: :RDS:: DBSecurity Group and Ingress.

JSON

```
"MyDbSecurityByCIDRIPGroup" : {
 "Type" : "AWS::RDS::DBSecurityGroup",
 "Properties" : {
     "GroupDescription" : "Ingress for CIDRIP",
     "DBSecurityGroupIngress" : {
         "CIDRIP" : "192.168.0.0/32"
     }
 }
}
```

YAML

```
MyDbSecurityByCIDRIPGroup:
  Type: AWS::RDS::DBSecurityGroup
  Properties:
    GroupDescription: Ingress for CIDRIP
    DBSecurityGroupIngress:
      CIDRIP: "192.168.0.0/32"
```

Amazon DBSecurity RDS-Gruppe mit einer EC2 Amazon-Sicherheitsgruppe

Dieses Beispiel zeigt eine AWS: :RDS:: DBSecurity -Gruppenressource mit Eingangsautorisierung von einer EC2 Amazon-Sicherheitsgruppe, auf die von verwiesen wird. `MyEc2SecurityGroup`

Dazu definieren Sie eine EC2 Sicherheitsgruppe und verwenden dann die systeminterne `Ref` Funktion, um auf die EC2 Sicherheitsgruppe innerhalb Ihrer Gruppe zu verweisen. `DBSecurityGroup`

JSON

```

"DBInstance" : {
   "Type": "AWS::RDS::DBInstance",
   "Properties": {
```

```
      "DBName"            : { "Ref" : "DBName" },
      "Engine"            : "MySQL",
      "MasterUsername"    : { "Ref" : "DBUsername" },
      "DBInstanceClass"   : { "Ref" : "DBClass" },
      "DBSecurityGroups"  : [ { "Ref" : "DBSecurityGroup" } ],
      "AllocatedStorage"  : { "Ref" : "DBAllocatedStorage" },
      "MasterUserPassword": { "Ref" : "DBPassword" }
   }
},

"DBSecurityGroup": {
   "Type": "AWS::RDS::DBSecurityGroup",
   "Properties": {
      "DBSecurityGroupIngress": {
         "EC2SecurityGroupName": {
            "Fn::GetAtt": ["WebServerSecurityGroup", "GroupName"]
         }
      },
      "GroupDescription" : "Frontend Access"
   }
},

"WebServerSecurityGroup" : {
   "Type" : "AWS::EC2::SecurityGroup",
   "Properties" : {
      "GroupDescription" : "Enable HTTP access via port 80 and SSH access",
      "SecurityGroupIngress" : [
         {"IpProtocol" : "tcp", "FromPort" : 80, "ToPort" : 80, "CidrIp" :
 "0.0.0.0/0"},
         {"IpProtocol" : "tcp", "FromPort" : 22, "ToPort" : 22, "CidrIp" : "0.0.0.0/0"}
      ]
   }
}
```

YAML

Dieses Beispiel wurde aus dem folgenden vollständigen Beispiel extrahiert: Drupal_Single_Instance_With_RDS.template

```
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName:
```

```
      Ref: DBName
    Engine: MySQL
    MasterUsername:
      Ref: DBUsername
    DBInstanceClass:
      Ref: DBClass
    DBSecurityGroups:
    - Ref: DBSecurityGroup
    AllocatedStorage:
      Ref: DBAllocatedStorage
    MasterUserPassword:
      Ref: DBPassword
DBSecurityGroup:
  Type: AWS::RDS::DBSecurityGroup
  Properties:
    DBSecurityGroupIngress:
      EC2SecurityGroupName:
        Ref: WebServerSecurityGroup
    GroupDescription: Frontend Access
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access
    SecurityGroupIngress:
    - IpProtocol: tcp
      FromPort: 80
      ToPort: 80
      CidrIp: 0.0.0.0/0
    - IpProtocol: tcp
      FromPort: 22
      ToPort: 22
      CidrIp: 0.0.0.0/0
```

Mehrere VPC-Sicherheitsgruppen

Dieses Beispiel zeigt eine AWS: :RDS:: DBSecurity Gruppenressource mit Eingangsautorisierung für mehrere Amazon EC2 VPC-Sicherheitsgruppen in AWS: :RDS::. DBSecurity GroupIngress

JSON

```
{
   "Resources" : {
      "DBinstance" : {
```

```
         "Type" : "AWS::RDS::DBInstance",
         "Properties" : {
            "AllocatedStorage" : "5",
            "DBInstanceClass" : "db.t2.small",
           "DBName" : {"Ref": "MyDBName" },
            "DBSecurityGroups" : [ { "Ref" : "DbSecurityByEC2SecurityGroup" } ],
            "DBSubnetGroupName" : { "Ref" : "MyDBSubnetGroup" },
            "Engine" : "MySQL",
           "MasterUserPassword": { "Ref" : "MyDBPassword" },
           "MasterUsername"    : { "Ref" : "MyDBUsername" }
        },
         "DeletionPolicy" : "Snapshot"
      },
      "DbSecurityByEC2SecurityGroup" : {
         "Type" : "AWS::RDS::DBSecurityGroup",
         "Properties" : {
            "GroupDescription" : "Ingress for Amazon EC2 security group",
           "EC2VpcId" : { "Ref" : "MyVPC" },
            "DBSecurityGroupIngress" : [ {
               "EC2SecurityGroupId" : "sg-b0ff1111",
               "EC2SecurityGroupOwnerId" : "111122223333"
            }, {
               "EC2SecurityGroupId" : "sg-ffd722222",
               "EC2SecurityGroupOwnerId" : "111122223333"
            } ]
         }
      }
   }
}
```

YAML

```
Resources:
  DBinstance:
    Type: AWS::RDS::DBInstance
    Properties:
      AllocatedStorage: '5'
      DBInstanceClass: db.t2.small
      DBName:
        Ref: MyDBName
      DBSecurityGroups:
      - Ref: DbSecurityByEC2SecurityGroup
      DBSubnetGroupName:
```

```
        Ref: MyDBSubnetGroup
      Engine: MySQL
      MasterUserPassword:
        Ref: MyDBPassword
      MasterUsername:
        Ref: MyDBUsername
    DeletionPolicy: Snapshot
  DbSecurityByEC2SecurityGroup:
    Type: AWS::RDS::DBSecurityGroup
    Properties:
      GroupDescription: Ingress for Amazon EC2 security group
      EC2VpcId:
        Ref: MyVPC
      DBSecurityGroupIngress:
      - EC2SecurityGroupId: sg-b0ff1111
        EC2SecurityGroupOwnerId: '111122223333'
      - EC2SecurityGroupId: sg-ffd722222
        EC2SecurityGroupOwnerId: '111122223333'
```

Amazon RDS-Datenbank-Instance in einer VPC-Sicherheitsgruppe

Dieses Beispiel zeigt eine Amazon RDS-Datenbank-Instance, die mit einer Amazon EC2 VPC-Sicherheitsgruppe verknüpft ist.

JSON

```
{
  "DBEC2SecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties" : {
      "GroupDescription": "Open database for access",
      "SecurityGroupIngress" : [{
        "IpProtocol" : "tcp",
        "FromPort" : 3306,
        "ToPort" : 3306,
        "SourceSecurityGroupName" : { "Ref" : "WebServerSecurityGroup" }
      }]
    }
  },
  "DBInstance" : {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "DBName"            : { "Ref" : "DBName" },
```

```
      "Engine"            : "MySQL",
      "MultiAZ"           : { "Ref": "MultiAZDatabase" },
      "MasterUsername"    : { "Ref" : "DBUser" },
      "DBInstanceClass"   : { "Ref" : "DBClass" },
      "AllocatedStorage"  : { "Ref" : "DBAllocatedStorage" },
      "MasterUserPassword": { "Ref" : "DBPassword" },
      "VPCSecurityGroups" : [ { "Fn::GetAtt": [ "DBEC2SecurityGroup", "GroupId" ] } ]
    }
  }
}
```

YAML

```
DBEC2SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Open database for access
    SecurityGroupIngress:
    - IpProtocol: tcp
      FromPort: 3306
      ToPort: 3306
      SourceSecurityGroupName:
        Ref: WebServerSecurityGroup
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName:
      Ref: DBName
    Engine: MySQL
    MultiAZ:
      Ref: MultiAZDatabase
    MasterUsername:
      Ref: DBUser
    DBInstanceClass:
      Ref: DBClass
    AllocatedStorage:
      Ref: DBAllocatedStorage
    MasterUserPassword:
      Ref: DBPassword
    VPCSecurityGroups:
    - !GetAtt DBEC2SecurityGroup.GroupId
```

Route-53-Vorlagenausschnitte

Themen

- Amazon Route 53-Ressourcendatensatz mit gehosteten Zone oder ID
- Wird RecordSetGroup zum Einrichten gewichteter Ressourcendatensätze verwendet
- Wird verwendet RecordSetGroup , um einen Alias-Ressourcendatensatz einzurichten
- Alias-Ressourcendatensatz für eine Verteilung CloudFront

Amazon Route 53-Ressourcendatensatz mit gehosteten Zone oder ID

Wenn Sie einen Amazon Route 53-Ressourcendatensatz erstellen, müssen Sie die Hosting-Zone angeben, in der Sie ihn hinzufügen möchten. CloudFormation bietet zwei Möglichkeiten, eine Hosting-Zone anzugeben:

- Sie können die gehostete Zone mit der Eigenschaft `HostedZoneId` explizit angeben.
- Sie können die Hosting-Zone mithilfe der `HostedZoneName` Eigenschaft CloudFormation gefunden haben. Wenn Sie die `HostedZoneName` Eigenschaft verwenden und es mehrere Hosting-Zonen mit demselben Namen gibt, wird der Stack CloudFormation nicht erstellt.

Hinzufügen RecordSet mit HostedZoneId

In diesem Beispiel wird ein Amazon Route 53-Ressourcendatensatz hinzugefügt, der einen `SPF` Datensatz für den Domainnamen enthält`mysite.example.com`, der die `HostedZoneId` Eigenschaft zur Angabe der Hosting-Zone verwendet.

JSON

```
"myDNSRecord" : {
  "Type" : "AWS::Route53::RecordSet",
  "Properties" :
  {
    "HostedZoneId" : "Z3DG6IL3SJCGPX",
    "Name" : "mysite.example.com.",
    "Type" : "SPF",
    "TTL" : "900",
    "ResourceRecords" : [ "\"v=spf1 ip4:192.168.0.1/16 -all\"" ]
  }
}
```

YAML

```
myDNSRecord:
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneId: Z3DG6IL3SJCGPX
    Name: mysite.example.com.
    Type: SPF
    TTL: '900'
    ResourceRecords:
    - '"v=spf1 ip4:192.168.0.1/16 -all"'
```

Hinzufügen RecordSet mit HostedZoneName

In diesem Beispiel wird ein Amazon Route 53-Ressourcendatensatz für den Domänennamen „mysite.example.com“ mit der Eigenschaft `HostedZoneName` zur Angabe der gehosteten Zone hinzugefügt.

JSON

```
"myDNSRecord2" : {
            "Type" : "AWS::Route53::RecordSet",
            "Properties" : {
                "HostedZoneName" : "example.com.",
                "Name" : "mysite.example.com.",
                "Type" : "A",
                "TTL" : "900",
                "ResourceRecords" : [
                    "192.168.0.1",
                    "192.168.0.2"
                ]
            }
        }
```

YAML

```
myDNSRecord2:
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneName: example.com.
    Name: mysite.example.com.
    Type: A
```

```
    TTL: '900'
    ResourceRecords:
    - 192.168.0.1
    - 192.168.0.2
```

Wird RecordSetGroup zum Einrichten gewichteter Ressourcendatensätze verwendet

In diesem Beispiel wird eine AWS::Route53::RecordSetGruppe verwendet, um zwei CNAME-Einträge für „example.com“ einzurichten. Gehostete Zone. Die Eigenschaft `RecordSets` enthält die CNAME-Datensätze für den DNS-Namen „mysite.example.com“. Jeder Datensatz enthält eine Kennung (`SetIdentifier`) und eine Gewichtung (`Weight`). Der Anteil des Internetverkehrs, der zu den Ressourcen geleitet wird, basiert auf den folgenden Berechnungen:

- `Frontend One`: `140/(140+60)` = `140/200` = 70%
- `Frontend Two`: `60/(140+60)` = `60/200` = 30%

Weitere Informationen über gewichtete Ressourcendatensätze finden Sie unter Gewichtetes Routing im Entwicklerhandbuch für Amazon Route 53.

JSON

```
        "myDNSOne" : {
            "Type" : "AWS::Route53::RecordSetGroup",
            "Properties" : {
                "HostedZoneName" : "example.com.",
                "Comment" : "Weighted RR for my frontends.",
                "RecordSets" : [
                  {
                    "Name" : "mysite.example.com.",
                    "Type" : "CNAME",
                    "TTL" : "900",
                    "SetIdentifier" : "Frontend One",
                    "Weight" : "140",
                    "ResourceRecords" : ["example-ec2.amazonaws.com"]
                  },
                  {
                    "Name" : "mysite.example.com.",
                    "Type" : "CNAME",
                    "TTL" : "900",
                    "SetIdentifier" : "Frontend Two",
                    "Weight" : "60",
```

```
                    "ResourceRecords" : ["example-ec2-larger.amazonaws.com"]
                }
                ]
            }
        }
```

YAML

```
myDNSOne:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Weighted RR for my frontends.
    RecordSets:
    - Name: mysite.example.com.
      Type: CNAME
      TTL: '900'
      SetIdentifier: Frontend One
      Weight: '140'
      ResourceRecords:
      - example-ec2.amazonaws.com
    - Name: mysite.example.com.
      Type: CNAME
      TTL: '900'
      SetIdentifier: Frontend Two
      Weight: '60'
      ResourceRecords:
      - example-ec2-larger.amazonaws.com
```

Wird verwendet RecordSetGroup , um einen Alias-Ressourcendatensatz einzurichten

In den folgenden Beispielen wird eine AWS::Route53::RecordSetGruppe verwendet, um einen Alias-Ressourcendatensatz mit dem Namen einzurichten`example.com`, der den Datenverkehr an einen ELB-Load-Balancer der Version 1 (Classic) und einen Load Balancer der Version 2 (Anwendung oder Netzwerk) weiterleitet. Die AliasTargetEigenschaft gibt die Hosting-Zonen-ID und den DNS-Namen für die `myELB` `LoadBalancer` mithilfe der `GetAtt` systemeigenen Funktion an. `GetAtt`ruft verschiedene Eigenschaften der `myELB` Ressource ab, je nachdem, ob Sie den Datenverkehr an einen Load Balancer der Version 1 oder Version 2 weiterleiten:

- Version 1 Load Balancer: `CanonicalHostedZoneNameID` und `DNSName`
- Version 2 Load Balancer: `CanonicalHostedZoneID` und `DNSName`

Weitere Informationen über Alias-Ressourcendatensätze finden Sie unter Wählen zwischen Alias- und Nicht-Alias-Datensätzen im Route-53-Entwicklerhandbuch.

JSON für Version 1 Load Balancer

```
      "myELB" : {
        "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties" : {
            "AvailabilityZones" : [ "us-east-1a" ],
            "Listeners" : [ {
                "LoadBalancerPort" : "80",
                "InstancePort" : "80",
                "Protocol" : "HTTP"
            } ]
        }
      },
      "myDNS" : {
        "Type" : "AWS::Route53::RecordSetGroup",
        "Properties" : {
          "HostedZoneName" : "example.com.",
          "Comment" : "Zone apex alias targeted to myELB LoadBalancer.",
          "RecordSets" : [
            {
              "Name" : "example.com.",
              "Type" : "A",
              "AliasTarget" : {
                  "HostedZoneId" : { "Fn::GetAtt" : ["myELB",
 "CanonicalHostedZoneNameID"] },
                  "DNSName" : { "Fn::GetAtt" : ["myELB","DNSName"] }
              }
            }
          ]
        }
    }
```

YAML für Version 1 Load Balancer

```
myELB:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
    - "us-east-1a"
    Listeners:
```

```
    - LoadBalancerPort: '80'
      InstancePort: '80'
      Protocol: HTTP
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Zone apex alias targeted to myELB LoadBalancer.
    RecordSets:
    - Name: example.com.
      Type: A
      AliasTarget:
        HostedZoneId: !GetAtt 'myELB.CanonicalHostedZoneNameID'
        DNSName: !GetAtt 'myELB.DNSName'
```

JSON für Version 2 Load Balancer

```
      "myELB" : {
        "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties" : {
            "Subnets" : [
                {"Ref": "SubnetAZ1"},
                {"Ref" : "SubnetAZ2"}
            ]
        }
      },
      "myDNS" : {
        "Type" : "AWS::Route53::RecordSetGroup",
        "Properties" : {
          "HostedZoneName" : "example.com.",
          "Comment" : "Zone apex alias targeted to myELB LoadBalancer.",
          "RecordSets" : [
            {
              "Name" : "example.com.",
              "Type" : "A",
              "AliasTarget" : {
                  "HostedZoneId" : { "Fn::GetAtt" : ["myELB",
 "CanonicalHostedZoneID"] },
                  "DNSName" : { "Fn::GetAtt" : ["myELB","DNSName"] }
              }
            }
          ]
        }
```

```
    }
```

YAML für Version 2 Load Balancer

```
myELB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Subnets:
    - Ref: SubnetAZ1
    - Ref: SubnetAZ2
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Zone apex alias targeted to myELB LoadBalancer.
    RecordSets:
    - Name: example.com.
      Type: A
      AliasTarget:
        HostedZoneId: !GetAtt 'myELB.CanonicalHostedZoneID'
        DNSName: !GetAtt 'myELB.DNSName'
```

Alias-Ressourcendatensatz für eine Verteilung CloudFront

Im folgenden Beispiel wird ein Alias-A-Datensatz erstellt, der einen benutzerdefinierten Domänennamen auf eine bestehende CloudFront Distribution verweist. `myHostedZoneID`wird angenommen, dass es sich entweder um einen Verweis auf eine tatsächliche `AWS::Route53::HostedZone` Ressource in derselben Vorlage oder um einen Parameter handelt. `myCloudFrontDistributionbezieht` sich auf eine `AWS::CloudFront::Distribution` Ressource innerhalb derselben Vorlage. Der Aliaseintrag verwendet die standardmäßige CloudFront Hosting-Zonen-ID (`Z2FDTNDATAQYW2`) und löst den Domainnamen der Distribution automatisch anhand von`Fn::GetAtt`. Dieses Setup ermöglicht die Weiterleitung des Webverkehrs von der benutzerdefinierten Domain zur CloudFront Distribution, ohne dass eine IP-Adresse erforderlich ist.

> **Note**
>
> Wenn Sie Alias-Ressourcendatensätze erstellen, müssen Sie dies `Z2FDTNDATAQYW2` für die `HostedZoneId` Eigenschaft angeben. Alias-Ressourcendatensätze für CloudFront können nicht in einer privaten Zone erstellt werden.

JSON

```
{
    "myDNS": {
        "Type": "AWS::Route53::RecordSetGroup",
        "Properties": {
            "HostedZoneId": {
                "Ref": "myHostedZoneID"
            },
            "RecordSets": [
                {
                    "Name": {
                        "Ref": "myRecordSetDomainName"
                    },
                    "Type": "A",
                    "AliasTarget": {
                        "HostedZoneId": "Z2FDTNDATAQYW2",
                        "DNSName": {
                            "Fn::GetAtt": [
                                "myCloudFrontDistribution",
                                "DomainName"
                            ]
                        },
                        "EvaluateTargetHealth": false
                    }
                }
            ]
        }
    }
}
```

YAML

```
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneId: !Ref myHostedZoneID
    RecordSets:
      - Name: !Ref myRecordSetDomainName
        Type: A
        AliasTarget:
          HostedZoneId: Z2FDTNDATAQYW2
          DNSName: !GetAtt
```

```
            - myCloudFrontDistribution
            - DomainName
          EvaluateTargetHealth: false
```

Amazon S3-Vorlagenausschnitte

Verwenden Sie diese Amazon S3 S3-Beispielvorlagen, um Ihre Amazon S3 S3-Buckets zu CloudFormation beschreiben. Weitere Beispiele finden Sie im Abschnitt Beispiele in der `AWS::S3::Bucket` Ressource.

Themen

- Erstellen eines Amazon S3-Buckets mit Standardeinstellungen
- Erstellen eines Amazon-S3-Buckets für das Hosten von Websites inklusive einer DeletionPolicy
- Erstellen einer statischen Website mit einer benutzerdefinierten Domäne

Erstellen eines Amazon S3-Buckets mit Standardeinstellungen

In diesem Beispiel wird mithilfe einer Ressource des Typs AWS::S3::Bucket ein Bucket mit Standardeinstellungen erstellt.

JSON

```
"myS3Bucket" : {
      "Type" : "AWS::S3::Bucket"
      }
```

YAML

```
MyS3Bucket:
    Type: AWS::S3::Bucket
```

Erstellen eines Amazon-S3-Buckets für das Hosten von Websites inklusive einer **`DeletionPolicy`**

In diesem Beispiel wird ein Bucket als Website erstellt und Block Public Access deaktiviert (öffentliche Leseberechtigungen sind für Buckets erforderlich, die für das Hosting von Websites eingerichtet wurden). Dem Bucket wird dann eine öffentliche Bucket-Richtlinie hinzugefügt. Da für diese Bucket-Ressource ein `DeletionPolicy` Attribut auf gesetzt ist`Retain`, CloudFormation wird dieser Bucket

beim Löschen des Stacks nicht gelöscht. Der `Output` Abschnitt dient `Fn::GetAtt` zum Abrufen des `WebsiteURL` Attributs und des `DomainName` Attributs der `S3Bucket` Ressource.

> **Note**
>
> In den folgenden Beispielen wird davon ausgegangen, dass die Einstellungen `BlockPublicPolicy` und „Öffentlichen Zugriff `RestrictPublicBuckets` blockieren“ auf Kontoebene deaktiviert wurden.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "S3Bucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "PublicAccessBlockConfiguration": {
                    "BlockPublicAcls": false,
                    "BlockPublicPolicy": false,
                    "IgnorePublicAcls": false,
                    "RestrictPublicBuckets": false
                },
                "WebsiteConfiguration": {
                    "IndexDocument": "index.html",
                    "ErrorDocument": "error.html"
                }
            },
            "DeletionPolicy": "Retain",
            "UpdateReplacePolicy": "Retain"
        },
        "BucketPolicy": {
            "Type": "AWS::S3::BucketPolicy",
            "Properties": {
                "PolicyDocument": {
                    "Id": "MyPolicy",
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Sid": "PublicReadForGetBucketObjects",
                            "Effect": "Allow",
```

```
                            "Principal": "*",
                            "Action": "s3:GetObject",
                            "Resource": {
                                "Fn::Join": [
                                    "",
                                    [
                                        "arn:aws:s3:::",
                                        {
                                            "Ref": "S3Bucket"
                                        },
                                        "/*"
                                    ]
                                ]
                            }
                        }
                    ]
                },
                "Bucket": {
                    "Ref": "S3Bucket"
                }
            }
        }
    },
    "Outputs": {
        "WebsiteURL": {
            "Value": {
                "Fn::GetAtt": [
                    "S3Bucket",
                    "WebsiteURL"
                ]
            },
            "Description": "URL for website hosted on S3"
        },
        "S3BucketSecureURL": {
            "Value": {
                "Fn::Join": [
                    "",
                    [
                        "https://",
                        {
                            "Fn::GetAtt": [
                                "S3Bucket",
                                "DomainName"
                            ]
```

```
                        }
                    ]
                ]
            },
            "Description": "Name of S3 bucket to hold website content"
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  S3Bucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      PublicAccessBlockConfiguration:
        BlockPublicAcls: false
        BlockPublicPolicy: false
        IgnorePublicAcls: false
        RestrictPublicBuckets: false
      WebsiteConfiguration:
        IndexDocument: index.html
        ErrorDocument: error.html
    DeletionPolicy: Retain
    UpdateReplacePolicy: Retain
  BucketPolicy:
    Type: 'AWS::S3::BucketPolicy'
    Properties:
      PolicyDocument:
        Id: MyPolicy
        Version: 2012-10-17
        Statement:
          - Sid: PublicReadForGetBucketObjects
            Effect: Allow
            Principal: '*'
            Action: 's3:GetObject'
            Resource: !Join
              - ''
              - - 'arn:aws:s3:::'
                - !Ref S3Bucket
                - /*
      Bucket: !Ref S3Bucket
```

```
Outputs:
  WebsiteURL:
    Value: !GetAtt
      - S3Bucket
      - WebsiteURL
    Description: URL for website hosted on S3
  S3BucketSecureURL:
    Value: !Join
      - ''
      - - 'https://'
        - !GetAtt
          - S3Bucket
          - DomainName
    Description: Name of S3 bucket to hold website content
```

Erstellen einer statischen Website mit einer benutzerdefinierten Domäne

Sie können Route 53 mit einer registrierten Domäne verwenden. Im folgenden Beispiel wird vorausgesetzt, dass Sie in Route 53 bereits eine gehostete Zone für Ihre Domäne erstellt haben. Das Beispiel erstellt zwei Buckets für das Hosten von Websites. Der Stamm-Bucket hostet die Inhalte, der andere Bucket leitet Anforderungen des Typs `www.`*`domainname`*`.com` zum Stamm-Bucket um. In den Datensätzen wird der Name Ihrer Domäne Amazon S3-Endpunkten zugewiesen.

Sie müssen auch eine Bucket-Richtlinie hinzufügen, wie in den oben aufgeführten Beispielen dargestellt.

Weitere Informationen zur Verwendung einer benutzerdefinierten Domain finden Sie unter Tutorial: Konfiguration einer statischen Website mithilfe einer bei Route 53 registrierten benutzerdefinierten Domain im Amazon Simple Storage Service-Benutzerhandbuch.

> **Note**
>
> In den folgenden Beispielen wird davon ausgegangen, dass die Einstellungen `BlockPublicPolicy` und `RestrictPublicBuckets` Block Public Access auf Kontoebene deaktiviert wurden.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
```

```
    "Mappings" : {
        "RegionMap" : {
            "us-east-1" : { "S3hostedzoneID" : "Z3AQBSTGFYJSTF", "websiteendpoint" :
 "s3-website-us-east-1.amazonaws.com" },
            "us-west-1" : { "S3hostedzoneID" : "Z2F56UZL2M1ACD", "websiteendpoint" :
 "s3-website-us-west-1.amazonaws.com" },
            "us-west-2" : { "S3hostedzoneID" : "Z3BJ6K6RIION7M", "websiteendpoint" :
 "s3-website-us-west-2.amazonaws.com" },
            "eu-west-1" : { "S3hostedzoneID" : "Z1BKCTXD74EZPE", "websiteendpoint" :
 "s3-website-eu-west-1.amazonaws.com" },
            "ap-southeast-1" : { "S3hostedzoneID" : "Z3O0J2DXBE1FTB",
 "websiteendpoint" : "s3-website-ap-southeast-1.amazonaws.com" },
            "ap-southeast-2" : { "S3hostedzoneID" : "Z1WCIGYICN2BYD",
 "websiteendpoint" : "s3-website-ap-southeast-2.amazonaws.com" },
            "ap-northeast-1" : { "S3hostedzoneID" : "Z2M4EHUR26P7ZW",
 "websiteendpoint" : "s3-website-ap-northeast-1.amazonaws.com" },
            "sa-east-1" : { "S3hostedzoneID" : "Z31GFT0UA1I2HV", "websiteendpoint" :
 "s3-website-sa-east-1.amazonaws.com" }
        }
    },
    "Parameters": {
        "RootDomainName": {
            "Description": "Domain name for your website (example.com)",
            "Type": "String"
        }
    },
    "Resources": {
        "RootBucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "BucketName" : {"Ref":"RootDomainName"},
                "PublicAccessBlockConfiguration": {
                    "BlockPublicAcls": false,
                    "BlockPublicPolicy": false,
                    "IgnorePublicAcls": false,
                    "RestrictPublicBuckets": false
                },
                "WebsiteConfiguration": {
                    "IndexDocument":"index.html",
                    "ErrorDocument":"404.html"
                }
            }
        },
        "WWWBucket": {
```

```
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "BucketName": {
                    "Fn::Join": ["", ["www.", {"Ref":"RootDomainName"}]]
                },
                "AccessControl": "BucketOwnerFullControl",
                "WebsiteConfiguration": {
                    "RedirectAllRequestsTo": {
                        "HostName": {"Ref": "RootBucket"}
                    }
                }
            }
        },
        "myDNS": {
            "Type": "AWS::Route53::RecordSetGroup",
            "Properties": {
                "HostedZoneName": {
                    "Fn::Join": ["", [{"Ref": "RootDomainName"}, "."]]
                },
                "Comment": "Zone apex alias.",
                "RecordSets": [
                    {
                        "Name": {"Ref": "RootDomainName"},
                        "Type": "A",
                        "AliasTarget": {
                            "HostedZoneId": {"Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "S3hostedzoneID"]},
                            "DNSName": {"Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "websiteendpoint"]}
                        }
                    },
                    {
                        "Name": {
                            "Fn::Join": ["", ["www.", {"Ref":"RootDomainName"}]]
                        },
                        "Type": "CNAME",
                        "TTL" : "900",
                        "ResourceRecords" : [
                            {"Fn::GetAtt":["WWWBucket", "DomainName"]}
                        ]
                    }
                ]
            }
        }
```

```
    },
    "Outputs": {
        "WebsiteURL": {
            "Value": {"Fn::GetAtt": ["RootBucket", "WebsiteURL"]},
            "Description": "URL for website hosted on S3"
        }
    }
}
```

YAML

```
Parameters:
  RootDomainName:
    Description: Domain name for your website (example.com)
    Type: String
Mappings:
  RegionMap:
    us-east-1:
      S3hostedzoneID: Z3AQBSTGFYJSTF
      websiteendpoint: s3-website-us-east-1.amazonaws.com
    us-west-1:
      S3hostedzoneID: Z2F56UZL2M1ACD
      websiteendpoint: s3-website-us-west-1.amazonaws.com
    us-west-2:
      S3hostedzoneID: Z3BJ6K6RIION7M
      websiteendpoint: s3-website-us-west-2.amazonaws.com
    eu-west-1:
      S3hostedzoneID: Z1BKCTXD74EZPE
      websiteendpoint: s3-website-eu-west-1.amazonaws.com
    ap-southeast-1:
      S3hostedzoneID: Z3O0J2DXBE1FTB
      websiteendpoint: s3-website-ap-southeast-1.amazonaws.com
    ap-southeast-2:
      S3hostedzoneID: Z1WCIGYICN2BYD
      websiteendpoint: s3-website-ap-southeast-2.amazonaws.com
    ap-northeast-1:
      S3hostedzoneID: Z2M4EHUR26P7ZW
      websiteendpoint: s3-website-ap-northeast-1.amazonaws.com
    sa-east-1:
      S3hostedzoneID: Z31GFT0UA1I2HV
      websiteendpoint: s3-website-sa-east-1.amazonaws.com
Resources:
  RootBucket:
```

```
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Ref RootDomainName
      PublicAccessBlockConfiguration:
        BlockPublicAcls: false
        BlockPublicPolicy: false
        IgnorePublicAcls: false
        RestrictPublicBuckets: false
      WebsiteConfiguration:
        IndexDocument: index.html
        ErrorDocument: 404.html
  WWWBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Sub
        - www.${Domain}
        - Domain: !Ref RootDomainName
      AccessControl: BucketOwnerFullControl
      WebsiteConfiguration:
        RedirectAllRequestsTo:
          HostName: !Ref RootBucket
  myDNS:
    Type: AWS::Route53::RecordSetGroup
    Properties:
      HostedZoneName: !Sub
        - ${Domain}.
        - Domain: !Ref RootDomainName
      Comment: Zone apex alias.
      RecordSets:
        - Name: !Ref RootDomainName
          Type: A
          AliasTarget:
            HostedZoneId: !FindInMap [ RegionMap, !Ref 'AWS::Region', S3hostedzoneID]
            DNSName: !FindInMap [ RegionMap, !Ref 'AWS::Region', websiteendpoint]
        - Name: !Sub
            - www.${Domain}
            - Domain: !Ref RootDomainName
          Type: CNAME
          TTL: 900
          ResourceRecords:
            - !GetAtt WWWBucket.DomainName
Outputs:
  WebsiteURL:
    Value: !GetAtt RootBucket.WebsiteURL
```

```
    Description: URL for website hosted on S3
```

Amazon SNS-Vorlagenausschnitte

Dieses Beispiel zeigt eine Amazon SNS-Themenressource. Es muss eine gültige E-Mail-Adresse vorhanden sein.

JSON

```
"MySNSTopic" : {
    "Type" : "AWS::SNS::Topic",
    "Properties" : {
        "Subscription" : [ {
            "Endpoint" : "add valid email address",
            "Protocol" : "email"
        } ]
    }
}
```

YAML

```
MySNSTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
    - Endpoint: "add valid email address"
      Protocol: email
```

Amazon SQS-Vorlagenausschnitte

Dieses Beispiel zeigt eine Amazon SQS-Warteschlange.

JSON

```
"MyQueue" : {
    "Type" : "AWS::SQS::Queue",
    "Properties" : {
        "VisibilityTimeout" : "value"
    }
}
```

YAML

```
MyQueue:
  Type: AWS::SQS::Queue
  Properties:
    VisibilityTimeout: value
```

Amazon Timestream Timestream-Vorlagenausschnitte

Amazon Timestream for InfluxDB macht es Anwendungsentwicklern und DevOps Teams leicht, vollständig verwaltete InfluxDB-Datenbanken AWS für Echtzeit-Zeitreihenanwendungen mithilfe von Open Source auszuführen. APIs Sie können schnell eine InfluxDB-Datenbank erstellen, die anspruchsvolle Zeitreihen-Workloads bewältigt. Mit ein paar einfachen API-Aufrufen können Sie eine InfluxDB-Datenbank AWS mit automatisiertem Software-Patching, Backups und Recovery einrichten, migrieren, betreiben und skalieren. Sie finden diese Beispiele auch unter awslabs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream -influxdb on. GitHub

Themen

- Minimale Stichprobe mit Standardwerten
- Vollständigeres Beispiel mit Parametern

Diese AWS CloudFormation Vorlagen erstellen die folgenden Ressourcen, die benötigt werden, um eine Amazon Timestream for InfluxDB-Instance erfolgreich zu erstellen, eine Verbindung zu ihr herzustellen und sie zu überwachen:

Amazon VPC

- `VPC`
- Eine oder mehrere `Subnet`
- `InternetGateway`
- `RouteTable`
- `SecurityGroup`

Amazon S3

- `Bucket`

Amazon Timestream

- `InfluxDBInstance`

Minimale Stichprobe mit Standardwerten

In diesem Beispiel wird eine öffentlich zugängliche Multi-AZ-Instance bereitgestellt, wobei nach Möglichkeit Standardwerte verwendet werden.

JSON

```
{
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {"default": "Amazon Timestream for InfluxDB Configuration"},
          "Parameters": [
            "DbInstanceName",
            "InfluxDBPassword"
          ]
        }
      ],
      "ParameterLabels": {
        "VPCCIDR": {"default": "VPC CIDR"}
      }
    }
  },
  "Parameters": {
    "DbInstanceName": {
      "Description": "The name that uniquely identifies the DB instance when
 interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name
 will also be a prefix included in the endpoint. DB instance names must be unique per
 customer and per Region.",
      "Type": "String",
      "Default": "mydbinstance",
      "MinLength": 3,
      "MaxLength": 40,
      "AllowedPattern": "^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$"
    },
    "InfluxDBPassword": {
      "Description": "The password of the initial admin user created in InfluxDB. This
 password will allow you to access the InfluxDB UI to perform various administrative
```

```
tasks and also use the InfluxDB CLI to create an operator token. These attributes will
be stored in a Secret created in AWS Secrets Manager in your account.",
     "Type": "String",
     "NoEcho": true,
     "MinLength": 8,
     "MaxLength": 64,
     "AllowedPattern": "^[a-zA-Z0-9]+$"
   }
 },
 "Resources": {
   "VPC": {
     "Type": "AWS::EC2::VPC",
     "Properties": {"CidrBlock": "10.0.0.0/16"}
   },
   "InternetGateway": {"Type": "AWS::EC2::InternetGateway"},
   "InternetGatewayAttachment": {
     "Type": "AWS::EC2::VPCGatewayAttachment",
     "Properties": {
       "InternetGatewayId": {"Ref": "InternetGateway"},
       "VpcId": {"Ref": "VPC"}
     }
   },
   "Subnet1": {
     "Type": "AWS::EC2::Subnet",
     "Properties": {
       "VpcId": {"Ref": "VPC"},
       "AvailabilityZone": {
         "Fn::Select": [
           0,
           {"Fn::GetAZs": ""}
         ]
       },
       "CidrBlock": {
         "Fn::Select": [
           0,
           {
             "Fn::Cidr": [
               {
                 "Fn::GetAtt": [
                   "VPC",
                   "CidrBlock"
                 ]
               },
               2,
```

```
                12
              ]
            }
          ]
        },
        "MapPublicIpOnLaunch": true
      }
    },
    "Subnet2": {
      "Type": "AWS::EC2::Subnet",
      "Properties": {
        "VpcId": {"Ref": "VPC"},
        "AvailabilityZone": {
          "Fn::Select": [
            1,
            {"Fn::GetAZs": ""}
          ]
        },
        "CidrBlock": {
          "Fn::Select": [
            1,
            {
              "Fn::Cidr": [
                {
                  "Fn::GetAtt": [
                    "VPC",
                    "CidrBlock"
                  ]
                },
                2,
                12
              ]
            }
          ]
        },
        "MapPublicIpOnLaunch": true
      }
    },
    "RouteTable": {
      "Type": "AWS::EC2::RouteTable",
      "Properties": {
        "VpcId": {"Ref": "VPC"}
      }
    },
```

```
    "DefaultRoute": {
      "Type": "AWS::EC2::Route",
      "DependsOn": "InternetGatewayAttachment",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": {"Ref": "InternetGateway"}
      }
    },
    "Subnet1RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet1"}
      }
    },
    "Subnet2RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet2"}
      }
    },
    "InfluxDBSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupName": "influxdb-sg",
        "GroupDescription": "Security group allowing port 8086 ingress for InfluxDB",
        "VpcId": {"Ref": "VPC"}
      }
    },
    "InfluxDBSecurityGroupIngress": {
      "Type": "AWS::EC2::SecurityGroupIngress",
      "Properties": {
        "GroupId": {"Ref": "InfluxDBSecurityGroup"},
        "IpProtocol": "tcp",
        "CidrIp": "0.0.0.0/0",
        "FromPort": 8086,
        "ToPort": 8086
      }
    },
    "InfluxDBLogsS3Bucket": {
      "Type": "AWS::S3::Bucket",
      "DeletionPolicy": "Retain"
```

```
    },
    "InfluxDBLogsS3BucketPolicy": {
      "Type": "AWS::S3::BucketPolicy",
      "Properties": {
        "Bucket": {"Ref": "InfluxDBLogsS3Bucket"},
        "PolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "s3:PutObject",
              "Effect": "Allow",
              "Resource": {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/
*"},
              "Principal": {"Service": "timestream-influxdb.amazonaws.com"}
            },
            {
              "Action": "s3:*",
              "Effect": "Deny",
              "Resource": [
                {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/*"},
                {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}"}
              ],
              "Principal": "*",
              "Condition": {
                "Bool": {"aws:SecureTransport": false}
              }
            }
          ]
        }
      }
    },
    "DbInstance": {
      "Type": "AWS::Timestream::InfluxDBInstance",
      "DependsOn": "InfluxDBLogsS3BucketPolicy",
      "Properties": {
        "AllocatedStorage": 20,
        "DbInstanceType": "db.influx.medium",
        "Name": {"Ref": "DbInstanceName"},
        "Password": {"Ref": "InfluxDBPassword"},
        "PubliclyAccessible": true,
        "DeploymentType": "WITH_MULTIAZ_STANDBY",
        "VpcSecurityGroupIds": [
          {"Ref": "InfluxDBSecurityGroup"}
        ],
```

```
        "VpcSubnetIds": [
          {"Ref": "Subnet1"},
          {"Ref": "Subnet2"}
        ],
        "LogDeliveryConfiguration": {
          "S3Configuration": {
            "BucketName": {"Ref": "InfluxDBLogsS3Bucket"},
            "Enabled": true
          }
        }
      }
    }
  },
  "Outputs": {
    "VPC": {
      "Description": "A reference to the VPC used to create network resources",
      "Value": {"Ref": "VPC"}
    },
    "Subnets": {
      "Description": "A list of the subnets created",
      "Value": {
        "Fn::Join": [
          ",",
          [
            {"Ref": "Subnet1"},
            {"Ref": "Subnet2"}
          ]
        ]
      }
    },
    "Subnet1": {
      "Description": "A reference to the subnet in the 1st Availability Zone",
      "Value": {"Ref": "Subnet1"}
    },
    "Subnet2": {
      "Description": "A reference to the subnet in the 2nd Availability Zone",
      "Value": {"Ref": "Subnet2"}
    },
    "InfluxDBSecurityGroup": {
      "Description": "Security group with port 8086 ingress rule",
      "Value": {"Ref": "InfluxDBSecurityGroup"}
    },
    "InfluxDBLogsS3Bucket": {
      "Description": "S3 Bucket containing InfluxDB logs from the DB instance",
```

```
      "Value": {"Ref": "InfluxDBLogsS3Bucket"}
    },
    "DbInstance": {
      "Description": "A reference to the Timestream for InfluxDB DB instance",
      "Value": {"Ref": "DbInstance"}
    },
    "InfluxAuthParametersSecretArn": {
      "Description": "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
 containing the initial InfluxDB authorization parameters. The secret value is a JSON
 formatted key-value pair holding InfluxDB authorization values: organization, bucket,
 username, and password.",
      "Value": {
        "Fn::GetAtt": [
          "DbInstance",
          "InfluxAuthParametersSecretArn"
        ]
      }
    },
    "Endpoint": {
      "Description": "The endpoint URL to connect to InfluxDB",
      "Value": {
        "Fn::Join": [
          "",
          [
            "https://",
            {
              "Fn::GetAtt": [
                "DbInstance",
                "Endpoint"
              ]
            },
            ":8086"
          ]
        ]
      }
    }
  }
}
```

YAML

```
Metadata:
  AWS::CloudFormation::Interface:
```

```
    ParameterGroups:
      -
        Label:
          default: "Amazon Timestream for InfluxDB Configuration"
        Parameters:
          - DbInstanceName
          - InfluxDBPassword
    ParameterLabels:
      VPCCIDR:
        default: VPC CIDR

Parameters:
  DbInstanceName:
    Description: The name that uniquely identifies the DB instance when interacting
 with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a
 prefix included in the endpoint. DB instance names must be unique per customer and per
 Region.
    Type: String
    Default: mydbinstance
    MinLength: 3
    MaxLength: 40
    AllowedPattern: ^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$
  InfluxDBPassword:
    Description: The password of the initial admin user created in InfluxDB. This
 password will allow you to access the InfluxDB UI to perform various administrative
 tasks and also use the InfluxDB CLI to create an operator token. These attributes will
 be stored in a Secret created in AWS Secrets Manager in your account.
    Type: String
    NoEcho: true
    MinLength: 8
    MaxLength: 64
    AllowedPattern: ^[a-zA-Z0-9]+$

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
  InternetGateway:
    Type: AWS::EC2::InternetGateway
  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
```

```
      VpcId: !Ref VPC
  Subnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [0, !GetAZs '']
      CidrBlock: !Select [0, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
      MapPublicIpOnLaunch: true
  Subnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [1, !GetAZs  '']
      CidrBlock: !Select [1, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
      MapPublicIpOnLaunch: true
  RouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
  DefaultRoute:
    Type: AWS::EC2::Route
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref RouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway
  Subnet1RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref RouteTable
      SubnetId: !Ref Subnet1
  Subnet2RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref RouteTable
      SubnetId: !Ref Subnet2
  InfluxDBSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: "influxdb-sg"
      GroupDescription: "Security group allowing port 8086 ingress for InfluxDB"
      VpcId: !Ref VPC
  InfluxDBSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
```

```
    Properties:
      GroupId: !Ref InfluxDBSecurityGroup
      IpProtocol: tcp
      CidrIp: 0.0.0.0/0
      FromPort: 8086
      ToPort: 8086
  InfluxDBLogsS3Bucket:
    Type: AWS::S3::Bucket
    DeletionPolicy: Retain
  InfluxDBLogsS3BucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      Bucket: !Ref InfluxDBLogsS3Bucket
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Action: "s3:PutObject"
            Effect: Allow
            Resource: !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*
            Principal:
              Service: timestream-influxdb.amazonaws.com
          - Action: "s3:*"
            Effect: Deny
            Resource:
              - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/*
              - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}
            Principal: "*"
            Condition:
              Bool:
                aws:SecureTransport: false
  DbInstance:
    Type: AWS::Timestream::InfluxDBInstance
    DependsOn: InfluxDBLogsS3BucketPolicy
    Properties:
      AllocatedStorage: 20
      DbInstanceType: db.influx.medium
      Name: !Ref DbInstanceName
      Password: !Ref InfluxDBPassword
      PubliclyAccessible: true
      DeploymentType: WITH_MULTIAZ_STANDBY
      VpcSecurityGroupIds:
        - !Ref InfluxDBSecurityGroup
      VpcSubnetIds:
        - !Ref Subnet1
```

```
        - !Ref Subnet2
      LogDeliveryConfiguration:
        S3Configuration:
          BucketName: !Ref InfluxDBLogsS3Bucket
          Enabled: true

Outputs:
  # Network Resources
  VPC:
    Description: A reference to the VPC used to create network resources
    Value: !Ref VPC
  Subnets:
    Description: A list of the subnets created
    Value: !Join [",", [!Ref Subnet1, !Ref Subnet2]]
  Subnet1:
    Description: A reference to the subnet in the 1st Availability Zone
    Value: !Ref Subnet1
  Subnet2:
    Description: A reference to the subnet in the 2nd Availability Zone
    Value: !Ref Subnet2
  InfluxDBSecurityGroup:
    Description: Security group with port 8086 ingress rule
    Value: !Ref InfluxDBSecurityGroup

  # Timestream for InfluxDB Resources
  InfluxDBLogsS3Bucket:
    Description: S3 Bucket containing InfluxDB logs from the DB instance
    Value: !Ref InfluxDBLogsS3Bucket
  DbInstance:
    Description: A reference to the Timestream for InfluxDB DB instance
    Value: !Ref DbInstance
  InfluxAuthParametersSecretArn:
    Description: "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
 containing the initial InfluxDB authorization parameters. The secret value is a JSON
 formatted key-value pair holding InfluxDB authorization values: organization, bucket,
 username, and password."
    Value: !GetAtt DbInstance.InfluxAuthParametersSecretArn
  Endpoint:
    Description: The endpoint URL to connect to InfluxDB
    Value: !Join ["", ["https://", !GetAtt DbInstance.Endpoint, ":8086"]]
```

Vollständigeres Beispiel mit Parametern

Diese Beispielvorlage ändert die Netzwerkressourcen dynamisch auf der Grundlage der angegebenen Parameter. Zu den Parametern gehören `PubliclyAccessible` und`DeploymentType`.

JSON

```
{
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {"default": "Network Configuration"},
          "Parameters": ["VPCCIDR"]
        },
        {
          "Label": {"default": "Amazon Timestream for InfluxDB Configuration"},
          "Parameters": [
            "DbInstanceName",
            "InfluxDBUsername",
            "InfluxDBPassword",
            "InfluxDBOrganization",
            "InfluxDBBucket",
            "DbInstanceType",
            "DbStorageType",
            "AllocatedStorage",
            "PubliclyAccessible",
            "DeploymentType"
          ]
        }
      ],
      "ParameterLabels": {
        "VPCCIDR": {"default": "VPC CIDR"}
      }
    }
  },
  "Parameters": {
    "VPCCIDR": {
      "Description": "Please enter the IP range (CIDR notation) for the new VPC",
      "Type": "String",
      "Default": "10.0.0.0/16"
    },
```

```
   "DbInstanceName": {
     "Description": "The name that uniquely identifies the DB instance when
interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name
will also be a prefix included in the endpoint. DB instance names must be unique per
customer and per Region.",
     "Type": "String",
     "Default": "mydbinstance",
     "MinLength": 3,
     "MaxLength": 40,
     "AllowedPattern": "^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$"
   },
   "InfluxDBUsername": {
     "Description": "The username of the initial admin user created in InfluxDB. Must
start with a letter and can't end with a hyphen or contain two consecutive hyphens.
For example, my-user1. This username will allow you to access the InfluxDB UI to
perform various administrative tasks and also use the InfluxDB CLI to create an
operator token. These attributes will be stored in a Secret created in AWS Secrets
Manager in your account.",
     "Type": "String",
     "Default": "admin",
     "MinLength": 1,
     "MaxLength": 64
   },
   "InfluxDBPassword": {
     "Description": "The password of the initial admin user created in InfluxDB. This
password will allow you to access the InfluxDB UI to perform various administrative
tasks and also use the InfluxDB CLI to create an operator token. These attributes will
be stored in a Secret created in AWS Secrets Manager in your account.",
     "Type": "String",
     "NoEcho": true,
     "MinLength": 8,
     "MaxLength": 64,
     "AllowedPattern": "^[a-zA-Z0-9]+$"
   },
   "InfluxDBOrganization": {
     "Description": "The name of the initial organization for the initial admin user
in InfluxDB. An InfluxDB organization is a workspace for a group of users.",
     "Type": "String",
     "Default": "org",
     "MinLength": 1,
     "MaxLength": 64
   },
   "InfluxDBBucket": {
```

```
      "Description": "The name of the initial InfluxDB bucket. All InfluxDB data
 is stored in a bucket. A bucket combines the concept of a database and a retention
 period (the duration of time that each data point persists). A bucket belongs to an
 organization.",
      "Type": "String",
      "Default": "bucket",
      "MinLength": 2,
      "MaxLength": 64,
      "AllowedPattern": "^[^_\\\"][^\\\"]*$"
    },
    "DeploymentType": {
      "Description": "Specifies whether the Timestream for InfluxDB is deployed as
 Single-AZ or with a MultiAZ Standby for High availability",
      "Type": "String",
      "Default": "WITH_MULTIAZ_STANDBY",
      "AllowedValues": [
        "SINGLE_AZ",
        "WITH_MULTIAZ_STANDBY"
      ]
    },
    "AllocatedStorage": {
      "Description": "The amount of storage to allocate for your DB storage type in GiB
 (gibibytes).",
      "Type": "Number",
      "Default": 400,
      "MinValue": 20,
      "MaxValue": 16384
    },
    "DbInstanceType": {
      "Description": "The Timestream for InfluxDB DB instance type to run InfluxDB
 on.",
      "Type": "String",
      "Default": "db.influx.medium",
      "AllowedValues": [
        "db.influx.medium",
        "db.influx.large",
        "db.influx.xlarge",
        "db.influx.2xlarge",
        "db.influx.4xlarge",
        "db.influx.8xlarge",
        "db.influx.12xlarge",
        "db.influx.16xlarge"
      ]
    },
```

```
    "DbStorageType": {
      "Description": "The Timestream for InfluxDB DB storage type to read and write
 InfluxDB data.",
      "Type": "String",
      "Default": "InfluxIOIncludedT1",
      "AllowedValues": [
        "InfluxIOIncludedT1",
        "InfluxIOIncludedT2",
        "InfluxIOIncludedT3"
      ]
    },
    "PubliclyAccessible": {
      "Description": "Configures the DB instance with a public IP to facilitate
 access.",
      "Type": "String",
      "Default": true,
      "AllowedValues": [
        true,
        false
      ]
    }
  },
  "Conditions": {
    "IsMultiAZ": {
      "Fn::Equals": [
        {"Ref": "DeploymentType"},
        "WITH_MULTIAZ_STANDBY"
      ]
    },
    "IsPublic": {
      "Fn::Equals": [
        {"Ref": "PubliclyAccessible"},
        true
      ]
    }
  },
  "Resources": {
    "VPC": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": {"Ref": "VPCCIDR"}
      }
    },
    "InternetGateway": {
```

```
      "Type": "AWS::EC2::InternetGateway",
      "Condition": "IsPublic"
    },
    "InternetGatewayAttachment": {
      "Type": "AWS::EC2::VPCGatewayAttachment",
      "Condition": "IsPublic",
      "Properties": {
        "InternetGatewayId": {"Ref": "InternetGateway"},
        "VpcId": {"Ref": "VPC"}
      }
    },
    "Subnet1": {
      "Type": "AWS::EC2::Subnet",
      "Properties": {
        "VpcId": {"Ref": "VPC"},
        "AvailabilityZone": {
          "Fn::Select": [
            0,
            {"Fn::GetAZs": ""}
          ]
        },
        "CidrBlock": {
          "Fn::Select": [
            0,
            {
              "Fn::Cidr": [
                {
                  "Fn::GetAtt": [
                    "VPC",
                    "CidrBlock"
                  ]
                },
                2,
                12
              ]
            }
          ]
        },
        "MapPublicIpOnLaunch": {
          "Fn::If": [
            "IsPublic",
            true,
            false
          ]
```

```
        }
      }
    },
    "Subnet2": {
      "Type": "AWS::EC2::Subnet",
      "Condition": "IsMultiAZ",
      "Properties": {
        "VpcId": {"Ref": "VPC"},
        "AvailabilityZone": {
          "Fn::Select": [
            1,
            {"Fn::GetAZs": ""}
          ]
        },
        "CidrBlock": {
          "Fn::Select": [
            1,
            {
              "Fn::Cidr": [
                {
                  "Fn::GetAtt": [
                    "VPC",
                    "CidrBlock"
                  ]
                },
                2,
                12
              ]
            }
          ]
        },
        "MapPublicIpOnLaunch": {
          "Fn::If": [
            "IsPublic",
            true,
            false
          ]
        }
      }
    },
    "RouteTable": {
      "Type": "AWS::EC2::RouteTable",
      "Properties": {
        "VpcId": {"Ref": "VPC"}
```

```
      }
    },
    "DefaultRoute": {
      "Type": "AWS::EC2::Route",
      "Condition": "IsPublic",
      "DependsOn": "InternetGatewayAttachment",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": {"Ref": "InternetGateway"}
      }
    },
    "Subnet1RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet1"}
      }
    },
    "Subnet2RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",
      "Condition": "IsMultiAZ",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet2"}
      }
    },
    "InfluxDBSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupName": "influxdb-sg",
        "GroupDescription": "Security group allowing port 8086 ingress for InfluxDB",
        "VpcId": {"Ref": "VPC"}
      }
    },
    "InfluxDBSecurityGroupIngress": {
      "Type": "AWS::EC2::SecurityGroupIngress",
      "Properties": {
        "GroupId": {"Ref": "InfluxDBSecurityGroup"},
        "IpProtocol": "tcp",
        "CidrIp": "0.0.0.0/0",
        "FromPort": 8086,
        "ToPort": 8086
      }
```

```
    },
    "InfluxDBLogsS3Bucket": {
      "Type": "AWS::S3::Bucket",
      "DeletionPolicy": "Retain"
    },
    "InfluxDBLogsS3BucketPolicy": {
      "Type": "AWS::S3::BucketPolicy",
      "Properties": {
        "Bucket": {"Ref": "InfluxDBLogsS3Bucket"},
        "PolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "s3:PutObject",
              "Effect": "Allow",
              "Resource": {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/
*"},
              "Principal": {"Service": "timestream-influxdb.amazonaws.com"}
            },
            {
              "Action": "s3:*",
              "Effect": "Deny",
              "Resource": [
                {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/*"},
                {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}"}
              ],
              "Principal": "*",
              "Condition": {
                "Bool": {"aws:SecureTransport": false}
              }
            }
          ]
        }
      }
    },
    "DbInstance": {
      "Type": "AWS::Timestream::InfluxDBInstance",
      "DependsOn": "InfluxDBLogsS3BucketPolicy",
      "Properties": {
        "DbStorageType": {"Ref": "DbStorageType"},
        "AllocatedStorage": {"Ref": "AllocatedStorage"},
        "DbInstanceType": {"Ref": "DbInstanceType"},
        "Name": {"Ref": "DbInstanceName"},
        "Username": {"Ref": "InfluxDBUsername"},
```

```
        "Password": {"Ref": "InfluxDBPassword"},
        "Organization": {"Ref": "InfluxDBOrganization"},
        "Bucket": {"Ref": "InfluxDBBucket"},
        "PubliclyAccessible": {
          "Fn::If": [
            "IsPublic",
            true,
            false
          ]
        },
        "DeploymentType": {"Ref": "DeploymentType"},
        "VpcSecurityGroupIds": [
          {"Ref": "InfluxDBSecurityGroup"}
        ],
        "VpcSubnetIds": {
          "Fn::If": [
            "IsMultiAZ",
            [
              {"Ref": "Subnet1"},
              {"Ref": "Subnet2"}
            ],
            [
              {"Ref": "Subnet1"}
            ]
          ]
        },
        "LogDeliveryConfiguration": {
          "S3Configuration": {
            "BucketName": {"Ref": "InfluxDBLogsS3Bucket"},
            "Enabled": true
          }
        }
      }
    }
  },
  "Outputs": {
    "VPC": {
      "Description": "A reference to the VPC used to create network resources",
      "Value": {"Ref": "VPC"}
    },
    "Subnets": {
      "Description": "A list of the subnets created",
      "Value": {
        "Fn::If": [
```

```
          "IsMultiAZ",
          {
            "Fn::Join": [
              ",",
              [
                {"Ref": "Subnet1"},
                {"Ref": "Subnet2"}
              ]
            ]
          },
          {"Ref": "Subnet1"}
        ]
      }
    },
    "Subnet1": {
      "Description": "A reference to the subnet in the 1st Availability Zone",
      "Value": {"Ref": "Subnet1"}
    },
    "Subnet2": {
      "Condition": "IsMultiAZ",
      "Description": "A reference to the subnet in the 2nd Availability Zone",
      "Value": {"Ref": "Subnet2"}
    },
    "InfluxDBSecurityGroup": {
      "Description": "Security group with port 8086 ingress rule",
      "Value": {"Ref": "InfluxDBSecurityGroup"}
    },
    "InfluxDBLogsS3Bucket": {
      "Description": "S3 Bucket containing InfluxDB logs from the DB instance",
      "Value": {"Ref": "InfluxDBLogsS3Bucket"}
    },
    "DbInstance": {
      "Description": "A reference to the Timestream for InfluxDB DB instance",
      "Value": {"Ref": "DbInstance"}
    },
    "InfluxAuthParametersSecretArn": {
      "Description": "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password.",
      "Value": {
        "Fn::GetAtt": [
          "DbInstance",
          "InfluxAuthParametersSecretArn"
```

```
        ]
      }
    },
    "Endpoint": {
      "Description": "The endpoint URL to connect to InfluxDB",
      "Value": {
        "Fn::Join": [
          "",
          [
            "https://",
            {
              "Fn::GetAtt": [
                "DbInstance",
                "Endpoint"
              ]
            },
            ":8086"
          ]
        ]
      }
    }
  }
}
```

YAML

```
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      -
        Label:
          default: "Network Configuration"
        Parameters:
          - VPCCIDR
      -
        Label:
          default: "Amazon Timestream for InfluxDB Configuration"
        Parameters:
          - DbInstanceName
          - InfluxDBUsername
          - InfluxDBPassword
          - InfluxDBOrganization
          - InfluxDBBucket
```

```
          - DbInstanceType
          - DbStorageType
          - AllocatedStorage
          - PubliclyAccessible
          - DeploymentType
    ParameterLabels:
      VPCCIDR:
        default: VPC CIDR

Parameters:
  # Network Configuration
  VPCCIDR:
    Description: Please enter the IP range (CIDR notation) for the new VPC
    Type: String
    Default: 10.0.0.0/16
  # Timestream for InfluxDB Configuration
  DbInstanceName:
    Description: The name that uniquely identifies the DB instance when interacting
 with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a
 prefix included in the endpoint. DB instance names must be unique per customer and per
 Region.
    Type: String
    Default: mydbinstance
    MinLength: 3
    MaxLength: 40
    AllowedPattern: ^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$
  # InfluxDB initial user configurations
  InfluxDBUsername:
    Description: The username of the initial admin user created in InfluxDB. Must start
 with a letter and can't end with a hyphen or contain two consecutive hyphens. For
 example, my-user1. This username will allow you to access the InfluxDB UI to perform
 various administrative tasks and also use the InfluxDB CLI to create an operator
 token. These attributes will be stored in a Secret created in AWS Secrets Manager in
 your account.
    Type: String
    Default: admin
    MinLength: 1
    MaxLength: 64
  InfluxDBPassword:
    Description: The password of the initial admin user created in InfluxDB. This
 password will allow you to access the InfluxDB UI to perform various administrative
 tasks and also use the InfluxDB CLI to create an operator token. These attributes will
 be stored in a Secret created in AWS in your account.
    Type: String
```

```
    NoEcho: true
    MinLength: 8
    MaxLength: 64
    AllowedPattern: ^[a-zA-Z0-9]+$
  InfluxDBOrganization:
    Description: The name of the initial organization for the initial admin user in
 InfluxDB. An InfluxDB organization is a workspace for a group of users.
    Type: String
    Default: org
    MinLength: 1
    MaxLength: 64
  InfluxDBBucket:
    Description: The name of the initial InfluxDB bucket. All InfluxDB data is stored
 in a bucket. A bucket combines the concept of a database and a retention period (the
 duration of time that each data point persists). A bucket belongs to an organization.
    Type: String
    Default: bucket
    MinLength: 2
    MaxLength: 64
    AllowedPattern: ^[^_\"][^\"]*$
  DeploymentType:
    Description: Specifies whether the Timestream for InfluxDB is deployed as Single-AZ
 or with a MultiAZ Standby for High availability
    Type: String
    Default: WITH_MULTIAZ_STANDBY
    AllowedValues:
      - SINGLE_AZ
      - WITH_MULTIAZ_STANDBY
  AllocatedStorage:
    Description: The amount of storage to allocate for your DB storage type in GiB
 (gibibytes).
    Type: Number
    Default: 400
    MinValue: 20
    MaxValue: 16384
  DbInstanceType:
    Description: The Timestream for InfluxDB DB instance type to run InfluxDB on.
    Type: String
    Default: db.influx.medium
    AllowedValues:
      - db.influx.medium
      - db.influx.large
      - db.influx.xlarge
      - db.influx.2xlarge
```

```
      - db.influx.4xlarge
      - db.influx.8xlarge
      - db.influx.12xlarge
      - db.influx.16xlarge
  DbStorageType:
    Description: The Timestream for InfluxDB DB storage type to read and write InfluxDB
 data.
    Type: String
    Default: InfluxIOIncludedT1
    AllowedValues:
      - InfluxIOIncludedT1
      - InfluxIOIncludedT2
      - InfluxIOIncludedT3
  PubliclyAccessible:
    Description: Configures the DB instance with a public IP to facilitate access.
    Type: String
    Default: true
    AllowedValues:
      - true
      - false

Conditions:
  IsMultiAZ: !Equals [!Ref DeploymentType, WITH_MULTIAZ_STANDBY]
  IsPublic: !Equals [!Ref PubliclyAccessible, true]

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VPCCIDR
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Condition: IsPublic
  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Condition: IsPublic
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC
  Subnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [0, !GetAZs '']
```

```
      CidrBlock: !Select [0, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
      MapPublicIpOnLaunch: !If [IsPublic, true, false]
  Subnet2:
    Type: AWS::EC2::Subnet
    Condition: IsMultiAZ
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [1, !GetAZs  '']
      CidrBlock: !Select [1, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
      MapPublicIpOnLaunch: !If [IsPublic, true, false]
  RouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
  DefaultRoute:
    Type: AWS::EC2::Route
    Condition: IsPublic
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref RouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway
  Subnet1RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref RouteTable
      SubnetId: !Ref Subnet1
  Subnet2RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Condition: IsMultiAZ
    Properties:
      RouteTableId: !Ref RouteTable
      SubnetId: !Ref Subnet2
  InfluxDBSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: "influxdb-sg"
      GroupDescription: "Security group allowing port 8086 ingress for InfluxDB"
      VpcId: !Ref VPC
  InfluxDBSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: !Ref InfluxDBSecurityGroup
      IpProtocol: tcp
```

```
      CidrIp: 0.0.0.0/0
      FromPort: 8086
      ToPort: 8086
  InfluxDBLogsS3Bucket:
    Type: AWS::S3::Bucket
    DeletionPolicy: Retain
  InfluxDBLogsS3BucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      Bucket: !Ref InfluxDBLogsS3Bucket
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Action: "s3:PutObject"
            Effect: Allow
            Resource: !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*
            Principal:
              Service: timestream-influxdb.amazonaws.com
          - Action: "s3:*"
            Effect: Deny
            Resource:
              - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/*
              - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}
            Principal: "*"
            Condition:
              Bool:
                aws:SecureTransport: false
  DbInstance:
    Type: AWS::Timestream::InfluxDBInstance
    DependsOn: InfluxDBLogsS3BucketPolicy
    Properties:
      DbStorageType: !Ref DbStorageType
      AllocatedStorage: !Ref AllocatedStorage
      DbInstanceType: !Ref DbInstanceType
      Name: !Ref DbInstanceName
      Username: !Ref InfluxDBUsername
      Password: !Ref InfluxDBPassword
      Organization: !Ref InfluxDBOrganization
      Bucket: !Ref InfluxDBBucket
      PubliclyAccessible: !If [IsPublic, true, false]
      DeploymentType: !Ref DeploymentType
      VpcSecurityGroupIds:
        - !Ref InfluxDBSecurityGroup
      VpcSubnetIds: !If
```

```
        - IsMultiAZ
        -
          - !Ref Subnet1
          - !Ref Subnet2
        -
          - !Ref Subnet1
      LogDeliveryConfiguration:
        S3Configuration:
          BucketName: !Ref InfluxDBLogsS3Bucket
          Enabled: true

Outputs:
  # Network Resources
  VPC:
    Description: A reference to the VPC used to create network resources
    Value: !Ref VPC
  Subnets:
    Description: A list of the subnets created
    Value: !If
      - IsMultiAZ
      - !Join [",", [!Ref Subnet1, !Ref Subnet2]]
      - !Ref Subnet1
  Subnet1:
    Description: A reference to the subnet in the 1st Availability Zone
    Value: !Ref Subnet1
  Subnet2:
    Condition: IsMultiAZ
    Description: A reference to the subnet in the 2nd Availability Zone
    Value: !Ref Subnet2
  InfluxDBSecurityGroup:
    Description: Security group with port 8086 ingress rule
    Value: !Ref InfluxDBSecurityGroup

  # Timestream for InfluxDB Resources
  InfluxDBLogsS3Bucket:
    Description: S3 Bucket containing InfluxDB logs from the DB instance
    Value: !Ref InfluxDBLogsS3Bucket
  DbInstance:
    Description: A reference to the Timestream for InfluxDB DB instance
    Value: !Ref DbInstance
  InfluxAuthParametersSecretArn:
    Description: "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
 containing the initial InfluxDB authorization parameters. The secret value is a JSON
```

```
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password."
   Value: !GetAtt DbInstance.InfluxAuthParametersSecretArn
 Endpoint:
   Description: The endpoint URL to connect to InfluxDB
   Value: !Join ["", ["https://", !GetAtt DbInstance.Endpoint, ":8086"]]
```

Bereitstellen von Windows basierten Stacks mit CloudFormation

Diese Seite enthält Links zur technischen Referenzdokumentation für CloudFormation Ressourcen, die häufig in Windows basierten Bereitstellungen verwendet werden.

CloudFormation bietet Unterstützung für die Bereitstellung und Verwaltung von Microsoft Windows Stacks über. Infrastructure as Code (IaC) Sie können es CloudFormation für die automatisierte Bereitstellung von Windows basierten EC2 Instances SQL Server auf Amazon RDS und Microsoft Active Directory über AWS Directory Service Amazon RDS verwenden.

AWS bietet vorkonfigurierte Amazon Machine Images (AMIs), die speziell für Windows Plattformen entwickelt wurden, um Sie bei der schnellen Bereitstellung von Anwendungen auf Amazon EC2 zu unterstützen. Dazu AMIs gehören Microsoft Standardeinstellungen und AWS spezifische Anpassungen. Mit CloudFormation können Sie ein geeignetes AMI auswählen, eine Instance starten und über Remote Desktop Connection darauf zugreifen, genau wie bei jeder anderenWindows Server. Sie AMIs enthalten wichtige Softwarekomponenten, darunter EC2Launch (Versionen variieren je nach Windows Server Edition) AWS Systems Manager, CloudFormation, AWS -Tools für PowerShell, und verschiedene Netzwerk-, Speicher- und Grafiktreiber, um eine optimale Leistung und Kompatibilität mit AWS Diensten zu gewährleisten. Weitere Informationen finden Sie in der AWSWindowsAMI-Referenz.

CloudFormation unterstützt auch Softwarekonfigurationstools wie `UserData` Skripts, die Befehle ausführen PowerShell oder Batch-Befehle ausführen können, wenn eine EC2 Instance zum ersten Mal gestartet wird. Es bietet auch Hilfsskripte (`cfn-initcfn-signal`,`cfn-get-metadata`, und`cfn-hup`) und unterstützt die `AWS::CloudFormation::Init` Metadaten für die Verwaltung von Paketen, Dateien und Diensten auf Windows Instanzen.

Für Unternehmensumgebungen CloudFormation ermöglicht es den Beitritt zu Domänen, die Windows Lizenzverwaltung mithilfe von EC2 Lizenzmodellen und die sichere Handhabung von Anmeldeinformationen mit AWS Secrets Manager. In Kombination mit versionsgesteuerten Vorlagen und wiederholbaren Bereitstellungen CloudFormation unterstützt es Unternehmen dabei, konsistente,

sichere und skalierbare Windows Umgebungen für mehrere Konten aufrechtzuerhalten. AWS-Regionen

Einzelheiten zu CloudFormation Ressourcen, die häufig in Windows basierten Bereitstellungen verwendet werden, finden Sie in den folgenden technischen Referenzthemen.

Ressourcentyp	Beschreibung
AWS::EC2::Instance	Zum Starten von Windows EC2 Instances.
AWS::EC2::SecurityGroup	Um Firewall-Regeln für Windows Workloads zu definieren.
AWS::AutoScaling::AutoScalingGroup (Gruppieren) AWS::EC2::LaunchTemplate	Zum Skalieren von Windows EC2 Instanzen.
AWS::DirectoryService::MicrosoftAD	Zum Bereitstellen MicrosoftActive Directory.
AWS::FSx::FileSystem	Zur Bereitstellung FSx für Windows Dateiserver.
AWS: :RDS:: DBInstance	Für die Bereitstellung SQL Server auf Amazon RDS.
AWS::CloudFormation::Init	Wird in EC2 Metadaten für die Konfiguration von Instances verwendet. Weitere Informationen finden Sie unter Bootstrapping-basierte Stacks Windows CloudFormation .
AWS::SecretsManager::Secret	Für die sichere Verwaltung von Anmeldeinformationen und Windows Passwörtern.
AWS::SSM::Parameter	Zum sicheren Speichern von Konfigurationswerten.
AWS::IAM::InstanceProfile AWS::IAM::Role	Zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Instanzen ausgeführt werden.

Bootstrapping-basierte Stacks Windows CloudFormation

In diesem Thema wird beschrieben, wie Sie einen Windows Stack booten und Probleme bei der Stapelerstellung beheben.

Themen

- Benutzerdaten in Instanzen EC2
- CloudFormation Hilfsskripte
- Beispiel für das Bootstrapping eines Stacks Windows
- Umgekehrte Schrägstriche in Dateipfaden umgehen Windows
- Dienste verwalten Windows
- Beheben Sie Probleme bei der Stack-Erstellung

Benutzerdaten in Instanzen EC2

Benutzerdaten sind eine EC2 Amazon-Funktion, mit der Sie Skripts oder Konfigurationsinformationen an eine EC2 Instance übergeben können, wenn diese gestartet wird.

Für Windows EC2 Instances:

- Sie können entweder Batch-Skripten (mit `<script>` Tags) oder PowerShell Skripten (mit `<powershell>` Tags) verwenden.
- Die Ausführung des Skripts erfolgt vonEC2Launch.

> **Important**
>
> Wenn Sie Ihr eigenes Windows AMI zur Verwendung mit erstellen CloudFormation, stellen Sie sicher, dass EC2Launch v2 ordnungsgemäß konfiguriert ist. EC2Launchv2 ist erforderlich, damit CloudFormation Bootstrapping-Tools Windows Instanzen während der Stack-Erstellung ordnungsgemäß initialisieren und konfigurieren können. Weitere Informationen finden Sie unter Verwenden des EC2Launch v2-Agenten zur Ausführung von Aufgaben beim Start von EC2 Windows-Instances im EC2 Amazon-Benutzerhandbuch. Informationen dazu AWS Windows AMIs finden Sie in der AWSWindowsAMI-Referenz.

CloudFormation Hilfsskripte

Hilfsskripte sind Hilfsprogramme für die Konfiguration von Instanzen während des Bootstrapping-Prozesses. In Verbindung mit EC2 Amazon-Benutzerdaten bieten sie leistungsstarke Konfigurationsoptionen.

CloudFormation stellt die folgenden Python-Hilfsskripte bereit, mit denen Sie Software installieren und Dienste auf einer EC2 Amazon-Instance starten können, die Sie als Teil Ihres Stacks erstellen:

- `cfn-init`— Wird verwendet, um Ressourcenmetadaten abzurufen und zu interpretieren, Pakete zu installieren, Dateien zu erstellen und Dienste zu starten.
- `cfn-signal`— Wird verwendet, um mit einem zu signalisieren`CreationPolicy`, sodass Sie andere Ressourcen im Stack synchronisieren können, wenn die erforderliche Ressource oder Anwendung bereit ist.
- `cfn-get-metadata`— Wird verwendet, um Metadaten für eine Ressource oder einen Pfad zu einem bestimmten Schlüssel abzurufen.
- `cfn-hup`— Wird verwendet, um nach Aktualisierungen von Metadaten zu suchen und benutzerdefinierte Hooks auszuführen, wenn Änderungen erkannt werden.

Sie können die Skripts direkt aus Ihrer Vorlage aufrufen. Die Skripts funktionieren in Verbindung mit Ressourcen-Metadaten, die in derselben Vorlage festgelegt werden. Die Skripts werden während der Stack-Erstellung auf der EC2 Amazon-Instance ausgeführt.

Weitere Informationen finden Sie in der Referenz zu CloudFormation Hilfsskripten im AWS CloudFormation Template Reference Guide.

Beispiel für das Bootstrapping eines Stacks Windows

Schauen wir uns Beispielausschnitte aus einer Windows Server Vorlage an, die die folgenden Aktionen ausführt:

- Startet eine EC2 Instance, die `TestInstance` nach einem AMI Windows Server 2022 benannt ist.
- Erstellt eine einfache Testdatei, um zu überprüfen, `cfn-init` ob sie funktioniert.
- Konfiguriert `cfn-hup` für die fortlaufende Konfigurationsverwaltung.
- Verwendet a`CreationPolicy`, um sicherzustellen, dass die Instanz den erfolgreichen Abschluss signalisiert.

Das `cfn-init` Hilfsskript wird verwendet, um jede dieser Aktionen auf der Grundlage der Informationen in der `AWS::CloudFormation::Init` Ressource in der Vorlage auszuführen.

Der `AWS::CloudFormation::Init` Abschnitt ist benannt `TestInstance` und beginnt mit der folgenden Deklaration.

```
TestInstance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      configSets:
        default:
          - create_files
          - start_services
```

Danach wird der `files` Abschnitt von `AWS::CloudFormation::Init` deklariert.

```
      create_files:
        files:
          c:\cfn\test.txt:
            content: !Sub |
              Hello from ${AWS::StackName}
          c:\cfn\cfn-hup.conf:
            content: !Sub |
              [main]
              stack=${AWS::StackName}
              region=${AWS::Region}
              interval=2
          c:\cfn\hooks.d\cfn-auto-reloader.conf:
            content: !Sub |
              [cfn-auto-reloader-hook]
              triggers=post.update
              path=Resources.TestInstance.Metadata.AWS::CloudFormation::Init
              action=cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --
region ${AWS::Region}
```

Drei Dateien werden hier erstellt und im `C:\cfn` Verzeichnis auf der Serverinstanz abgelegt:

- `test.txt`, eine einfache Testdatei, die überprüft, ob sie `cfn-init` korrekt funktioniert und Dateien mit dynamischem Inhalt erstellen kann.
- `cfn-hup.conf`, die Konfigurationsdatei für `cfn-hup` mit einem Prüfintervall von 2 Minuten.

- `cfn-auto-reloader.conf`, die Konfigurationsdatei für den Hook, der von verwendet wird`cfn-hup`, um ein Update (Aufruf`cfn-init`) zu initiieren, wenn sich die Metadaten `AWS::CloudFormation::Init` ändern.

Als Nächstes folgt der `start_services` Abschnitt, in dem Windows Dienste konfiguriert werden.

```
      start_services:
        services:
          windows:
            cfn-hup:
              enabled: true
              ensureRunning: true
              files:
                - c:\cfn\cfn-hup.conf
                - c:\cfn\hooks.d\cfn-auto-reloader.conf
```

Dieser Abschnitt stellt sicher, dass der `cfn-hup` Dienst gestartet wird und automatisch neu gestartet wird, wenn die Konfigurationsdateien geändert werden. Der Dienst sucht nach Änderungen an den CloudFormation Metadaten und wird erneut ausgeführt`cfn-init`, wenn Updates erkannt werden.

Als Nächstes folgt der `Properties` Abschnitt.

```
TestInstance:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT20M
  Metadata:
    AWS::CloudFormation::Init:
      # ... metadata configuration ...
  Properties:
    InstanceType: t2.large
    ImageId: '{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2022-
English-Full-Base}}'
    SecurityGroupIds:
      - !Ref InstanceSecurityGroup
    KeyName: !Ref KeyPairName
    UserData:
      Fn::Base64: !Sub |
        <powershell>
        cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --region
 ${AWS::Region}
```

```
        cfn-signal.exe -e $lastexitcode --stack ${AWS::StackName} --resource
 TestInstance --region ${AWS::Region}
        </powershell>
```

In diesem Abschnitt enthält die `UserData` Eigenschaft ein PowerShell Skript, das von `<powershell>` Tags umgeben EC2Launch ausgeführt wird. Das Skript wird `cfn-init` mit dem `default` configSet ausgeführt und verwendet dann, `cfn-signal` um den Exit-Code an CloudFormation zurückzumelden. Das `CreationPolicy` wird verwendet, um sicherzustellen, dass die Instanz ordnungsgemäß konfiguriert ist, bevor die Stack-Erstellung als abgeschlossen betrachtet wird.

Die `ImageId` Eigenschaft verwendet einen öffentlichen Parameter von Systems Manager Parameter Store, um automatisch die neueste AMI-ID Windows Server 2022 abzurufen. Dieser Ansatz macht regionsspezifische AMI-Zuordnungen überflüssig und stellt sicher, dass Sie immer das neueste AMI erhalten. Die Verwendung von Systems Manager Manager-Parametern für AMI IDs ist eine bewährte Methode zur Pflege aktueller AMI-Referenzen. Wenn Sie eine Verbindung zu Ihrer Instance herstellen möchten, stellen Sie sicher, dass die `SecurityGroupIds` Eigenschaft auf eine Sicherheitsgruppe verweist, die den RDP-Zugriff ermöglicht.

Die `CreationPolicy` wird als Teil der Ressourceneigenschaften deklariert und gibt einen Timeout-Zeitraum an. Der `cfn-signal` Befehl in den Benutzerdaten signalisiert, wenn die Instanzkonfiguration abgeschlossen ist:

```
TestInstance:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT20M
  Properties:
    # ... other properties ...
```

Da der Bootstrapping-Vorgang minimal ist und nur Dateien erstellt und Dienste gestartet werden, `CreationPolicy` wartet der Server 20 Minuten (PT20M), bevor es zu einem Timeout kommt. Das Timeout wird im ISO 8601-Dauerformat angegeben. Beachten Sie, dass der Start von Windows Instances in der Regel länger dauert als der von Linux-Instances. Testen Sie daher gründlich, um die für Ihre Anforderungen am besten geeigneten Timeout-Werte zu ermitteln.

Wenn alles gut geht, wird die Instance erfolgreich `CreationPolicy` abgeschlossen und Sie können über ihre öffentliche IP-Adresse auf die Windows Server Instance zugreifen. Sobald die

Stack-Erstellung abgeschlossen ist, werden die Instanz-ID und die öffentliche IP-Adresse auf der Registerkarte Ausgaben der CloudFormation Konsole angezeigt.

```
Outputs:
  InstanceId:
    Value: !Ref TestInstance
    Description: Instance ID of the Windows Server
  PublicIP:
    Value: !GetAtt TestInstance.PublicIp
    Description: Public IP address of the Windows Server
```

Sie können auch manuell überprüfen, ob das Bootstrapping ordnungsgemäß funktioniert hat, indem Sie über RDP eine Verbindung zur Instance herstellen und überprüfen, ob die Datei `C:\cfn\test.txt` vorhanden ist und den erwarteten Inhalt enthält. Weitere Informationen zum Herstellen einer Verbindung zu Windows Instances finden Sie unter Verbindung zu Ihrer Windows Instance mithilfe von RDP herstellen im EC2 Amazon-Benutzerhandbuch.

Umgekehrte Schrägstriche in Dateipfaden umgehen Windows

Denken Sie beim Verweisen auf Windows Pfade in CloudFormation Vorlagen immer daran, Backslashes (\) entsprechend dem verwendeten Vorlagenformat korrekt zu maskieren.

- Bei JSON-Vorlagen müssen Sie doppelte Backslashes in Windows Dateipfaden verwenden, da JSON Backslash als Escape-Zeichen behandelt. Der erste Backslash maskiert den zweiten, was zur Interpretation eines einzelnen, wörtlichen Backslashs führt.

  ```
  "commands" : {
    "1-extract" : {
      "command" : "C:\\SharePoint\\SharePointFoundation2010.exe /extract:C:\\SharePoint
  \\SPF2010 /quiet /log:C:\\SharePoint\\SharePointFoundation2010-extract.log"
    }
  }
  ```

- Für YAML-Vorlagen sind einzelne Backslashes in der Regel ausreichend.

  ```
  commands:
    1-extract:
      command: C:\SharePoint\SharePointFoundation2010.exe /extract:C:\SharePoint
  \SPF2010 /quiet /log:C:\SharePoint\SharePointFoundation2010-extract.log
  ```

Dienste verwalten Windows

Sie verwalten Windows Dienste auf die gleiche Weise wie Linux-Dienste, außer dass Sie stattdessen einen `windows` Schlüssel verwendensysvinit. Im folgenden Beispiel wird der `cfn-hup` Dienst gestartet, auf Automatisch gesetzt und der Dienst neu gestartet, wenn die `cfn-init` `c:\cfn\hooks.d\cfn-auto-reloader.conf` Konfigurationsdateien `c:\cfn\cfn-hup.conf` oder geändert werden.

```
services:
  windows:
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - c:\cfn\cfn-hup.conf
        - c:\cfn\hooks.d\cfn-auto-reloader.conf
```

Sie können andere Windows Dienste auf die gleiche Weise verwalten, indem Sie den Namen und nicht den Anzeigenamen verwenden, um auf den Dienst zu verweisen.

Beheben Sie Probleme bei der Stack-Erstellung

Wenn Ihr Stack bei der Erstellung fehlschlägt, besteht das Standardverhalten darin, bei einem Fehler ein Rollback durchzuführen. Dies ist zwar in der Regel ein gutes Standardverhalten, da unnötige Kosten vermieden werden, es ist aber schwierig zu debuggen, warum bei der Stack-Erstellung ein Fehler auftrat.

Um dieses Verhalten zu deaktivieren, wenn Sie Ihren Stack mit der CloudFormation Konsole erstellen oder aktualisieren, wählen Sie unter Optionen für Stapelfehler die Option Erfolgreich bereitgestellte Ressourcen beibehalten aus. Weitere Informationen finden Sie unter Wählen Sie aus, wie mit Ausfällen bei der Bereitstellung von Ressourcen umgegangen werden soll. Auf diese Weise können Sie sich bei Ihrer Instance anmelden und die Protokolldateien einsehen, um Probleme zu ermitteln, die bei der Ausführung Ihrer Startskripts aufgetreten sind.

Wichtige Protokolle, die sich Sie ansehen sollten:

- Das EC2 Konfigurationsprotokoll unter `%ProgramData%\Amazon\EC2Launch\log\agent.log`
- Das cfn-init Protokoll unter `C:\cfn\log\cfn-init.log` (überprüfen Sie die Exit-Codes und Fehlermeldungen für bestimmte Fehlerpunkte)

Weitere Protokolle finden Sie unter den folgenden Themen im EC2 Amazon-Benutzerhandbuch:

- EC2LaunchVerzeichnisstruktur
- EC2Launchv2-Verzeichnisstruktur

Weitere Informationen zur Behebung von Bootstrapping-Problemen finden Sie unter Wie behebe ich Probleme mit Hilfsskripten, die in einem CloudFormation Stapel mit Instanzen kein Bootstrap durchführen können? Windows .

Erweitern Sie die Funktionen Ihrer Vorlage mit CloudFormation bereitgestellten Ressourcentypen

CloudFormation bietet mehrere Ressourcentypen, die Sie in Ihrer Stack-Vorlage verwenden können, um deren Funktionen über die einer einfachen Stack-Vorlage hinaus zu erweitern.

Zu diesen Ressourcentypen gehören:

Ressourcentyp	Beschreibung	Dokumentation
Benutzerdefinierte Ressourcen	Mit dem `AWS::CloudFormation::CustomResource` Ressourcentyp können Sie benutzerdefinierte Ressourcen erstellen, die bestimmte Bereitstellungsaufgaben ausführen können oder Ressourcen einschließen, die nicht als CloudFormation Ressourcentypen verfügbar sind.	Benutzerdefinierte Ressourcen
Makros	Der `AWS::CloudFormation::Macro` Ressourcentyp definiert einen wiederverwendbaren Code, der eine benutzerdefinierte Verarbeitung von CloudFormation Vorlagen durchführen kann. Makros können Ihre Vorlagen ändern, zusätzliche Ressourcen generieren oder andere benutzerdefinierte Operationen während der Stack-Erstellung oder Aktualisierung ausführen.	Vorlagenmakros

Ressourcentyp	Beschreibung	Dokumentation
Verschachtelte Stacks	Der `AWS::CloudFormation::Stack` Ressourcentyp ermöglicht es Ihnen, verschachtelte Stapel innerhalb Ihrer CloudFormation Vorlagen zu erstellen, um modularere und wiederverwendbare Stack-Architekturen zu erhalten.	Verschachtelte Stacks
StackSet	Der `AWS::CloudFormation::StackSet` Ressourcentyp erstellt oder aktualisiert einen. CloudFormation StackSet Dabei handelt es sich um einen Container für Stacks, die in mehreren Regionen eingesetzt werden können. AWS-Konten	Verwaltung von Stacks mit StackSets
Wartebedingung	Der `AWS::CloudFormation::WaitCondition` Ressourcentyp unterbricht die Erstellung oder Aktualisierung von Stacks, bis eine bestimmte Bedingung erfüllt ist, z. B. der erfolgreiche Abschluss eines lang andauernden Prozesses oder die Verfügbarkeit externer Ressourcen.	Wartebedingungen
Warten Sie, ob die Bedingung erfüllt	Der `AWS::CloudFormation::WaitConditionHandle` Ressourcentyp arbeitet mit dem `AWS::CloudFormation::WaitCondition` Ressourcentyp zusammen. Es stellt eine vordefinierte URL bereit, die verwendet wird, um Signale zu senden, die angeben, dass eine bestimmte Bedingung erfüllt ist. Diese Signale ermöglichen es, mit der Erstellung oder Aktualisierung des Stacks fortzufahren.	Wartebedingungen

Erstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen

Benutzerdefinierte Ressourcen bieten Ihnen die Möglichkeit, benutzerdefinierte Bereitstellungslogik in Ihre CloudFormation Vorlagen zu schreiben und sie jedes Mal CloudFormation auszuführen, wenn Sie einen Stack erstellen, aktualisieren (falls Sie die benutzerdefinierte Ressource geändert haben) oder löschen. Dies kann nützlich sein, wenn Ihre Bereitstellungsanforderungen komplexe Logiken oder Workflows beinhalten, die mit CloudFormation den integrierten Ressourcentypen nicht ausgedrückt werden können.

Beispielsweise möchten Sie möglicherweise Ressourcen einbeziehen, die nicht als CloudFormation Ressourcentypen verfügbar sind. Sie können diese Ressourcen mithilfe von benutzerdefinierten Ressourcen einschließen. Auf diese Weise können Sie weiterhin alle zugehörigen Ressourcen in einem einzigen Stapel verwalten.

Um eine benutzerdefinierte Ressource in Ihrer CloudFormation Vorlage zu definieren, verwenden Sie den `Custom::MyCustomResourceTypeName`Ressourcentyp `AWS::CloudFormation::CustomResource`oder. Benutzerdefinierte Ressourcen benötigen eine Eigenschaft, das Service-Token, das angibt, wohin Anfragen CloudFormation gesendet werden, z. B. ein Amazon SNS SNS-Thema oder eine Lambda-Funktion.

Die folgenden Themen enthalten Informationen zur Verwendung benutzerdefinierter Ressourcen.

Themen

- So funktionieren benutzerdefinierte Ressourcen
- Timeout für die Antwort
- Amazon SNS-gestützte benutzerdefinierte Ressourcen
- Lambda-gestützte benutzerdefinierte Ressourcen
- Referenz für benutzerdefinierte Ressourcen

> **Note**
>
> Die CloudFormation Registrierung und die benutzerdefinierten Ressourcen bieten jeweils ihre eigenen Vorteile. Benutzerdefinierte Ressourcen bieten die folgenden Vorteile:
>
> - Sie müssen die Ressource nicht registrieren.

- Sie können eine gesamte Ressource als Teil einer Vorlage verwenden, ohne sich zu registrieren.
- Unterstützt die `Delete` Operationen `CreateUpdate`, und

Zu den Vorteilen, die registrierungsbasierte Ressourcen bieten, gehören:

- Unterstützt die Modellierung, Provisionierung und Verwaltung von Anwendungsressourcen von Drittanbietern
- Unterstützt die Vorgänge `Create`, `Read`, `Update`, `Delete` und `List` (`CRUDL`)
- Unterstützt die Drifterkennung für Ressourcentypen privater und Drittanbieter

Im Gegensatz zu benutzerdefinierten Ressourcen müssen registrierungsbasierte Ressourcen kein Amazon SNS-Thema oder Lambda-Funktion zur Durchführung von `CRUDL`-Vorgängen verknüpfen. Weitere Informationen finden Sie unter Erweiterungen mit der CloudFormation Registry verwalten.

So funktionieren benutzerdefinierte Ressourcen

Das allgemeine Verfahren zum Einrichten einer neuen benutzerdefinierten Ressource umfasst die folgenden Schritte. Diese Schritte umfassen zwei Rollen: den Anbieter für benutzerdefinierte Ressourcen, der Eigentümer der benutzerdefinierten Ressource ist, und den Vorlagenentwickler, der eine Vorlage erstellt, die einen benutzerdefinierten Ressourcentyp enthält. Dabei kann es sich um dieselbe Person handeln, aber wenn nicht, sollte der Anbieter für benutzerdefinierte Ressourcen mit dem Vorlagenentwickler zusammenarbeiten.

1. Der benutzerdefinierte Ressourcenanbieter schreibt eine Logik, die bestimmt, wie Anfragen von der benutzerdefinierten Ressource behandelt CloudFormation und Aktionen für diese ausgeführt werden.
2. Der benutzerdefinierte Ressourcenanbieter erstellt das Amazon SNS SNS-Thema oder die Lambda-Funktion, an die Anfragen gesendet CloudFormation werden können. Das Amazon SNS SNS-Thema oder die Lambda-Funktion müssen sich in derselben Region befinden, in der der Stack erstellt wird.
3. Der benutzerdefinierte Ressourcenanbieter gibt dem Vorlagenentwickler den ARN für das Amazon SNS SNS-Thema oder die Lambda-Funktion.

4. Der Vorlagenentwickler definiert die benutzerdefinierte Ressource in seiner CloudFormation Vorlage. Dazu gehören ein Service-Token und alle Eingabedatenparameter. Das Service-Token und die Struktur der Eingabedaten werden vom benutzerdefinierten Ressourcenanbieter definiert. Das Service-Token spezifiziert den ARN für das Amazon SNS SNS-Thema oder die Lambda-Funktion und ist immer erforderlich, aber die Eingabedaten sind je nach benutzerdefinierter Ressource optional.

Wenn nun jemand die Vorlage verwendet, um die benutzerdefinierte Ressource zu erstellen, zu aktualisieren oder zu löschen, CloudFormation sendet er eine Anfrage an das angegebene Service-Token und wartet dann auf eine Antwort, bevor er mit dem Stack-Vorgang fortfährt.

Im Folgenden wird der Ablauf für die Erstellung eines Stacks aus der Vorlage zusammengefasst:

1. CloudFormation sendet eine Anfrage an das angegebene Service-Token. Die Anfrage enthält Informationen wie den Anforderungstyp und eine vorsignierte Amazon Simple Storage Service-URL, an die die benutzerdefinierte Ressource Antworten sendet. Weitere Informationen zu den in der Anforderung enthaltenen Informationen finden Sie unter Anforderungsobjekte für benutzerdefinierte Ressourcen.

   Die folgenden Beispieldaten zeigen, was in einer `Create` Anfrage CloudFormation enthalten ist. In diesem Beispiel `ResourceProperties` können CloudFormation Sie eine benutzerdefinierte Nutzlast erstellen, die an die Lambda-Funktion gesendet werden soll.

```
{
   "RequestType" : "Create",
   "ResponseURL" : "http://pre-signed-S3-url-for-response",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "RequestId" : "unique id for this create request",
   "ResourceType" : "Custom::TestResource",
   "LogicalResourceId" : "MyTestResource",
   "ResourceProperties" : {
      "Name" : "Value",
      "List" : [ "1", "2", "3" ]
   }
}
```

2. Der benutzerdefinierte Ressourcenanbieter verarbeitet die CloudFormation Anfrage und gibt eine Antwort von `SUCCESS` oder `FAILED` an die vorsignierte URL zurück. Der custom resource provider gibt die Antwort in einer JSON-formatierten Datei zurück und lädt diese zu der vorsignierten S3-

URL hoch. Weitere Informationen finden Sie unter Hochladen von Objekten mit Vorsignierung URLs im Amazon Simple Storage Service-Benutzerhandbuch.

Der custom resource provider kann in der Antwort auch Name-Wert-Paare angeben, auf die der template developer zugreifen kann. Die Antwort kann beispielsweise Ausgabedaten enthalten, wenn die Anforderung erfolgreich war, oder eine Fehlermeldung, wenn die Anforderung fehlgeschlagen ist. Weitere Informationen zu Antworten finden Sie unter Benutzerdefinierte Ressourcenantwortobjekte.

> **Important**
>
> Wenn die Namen-Wert-Paare vertrauliche Informationen enthalten, sollten Sie das `NoEcho`-Feld benutzen, um die Ausgabe der benutzerdefinierten Ressource zu maskieren. Andernfalls sind die Werte durch die Eigenschaftswerte APIs dieser Oberfläche sichtbar (z. B.`DescribeStackEvents`).
> Weitere Informationen zur Verwendung `NoEcho` zum Maskieren vertraulicher Informationen finden Sie in der Keine Anmeldeinformationen in Vorlagen einbetten bewährten Methode.

Der custom resource provider ist für das Erkennen und Beantworten der Anforderung verantwortlich. Bei Amazon SNS SNS-Benachrichtigungen muss der Anbieter benutzerdefinierter Ressourcen beispielsweise Benachrichtigungen abhören und beantworten, die an einen bestimmten Themen-ARN gesendet werden. CloudFormation wartet und wartet auf eine Antwort an der Stelle mit der vorab signierten URL.

In den folgenden Beispieldaten wird gezeigt, was eine benutzerdefinierte Ressourcen in eine Antwort einschließen kann:

```
{
   "Status" : "SUCCESS",
   "PhysicalResourceId" : "TestResource1",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "RequestId" : "unique id for this create request",
   "LogicalResourceId" : "MyTestResource",
   "Data" : {
      "OutputName1" : "Value1",
      "OutputName2" : "Value2",
```

```
    }
}
```

3. Nachdem Sie eine `SUCCESS` Antwort erhalten haben, wird mit dem CloudFormation Stack-Vorgang fortgefahren. Wenn keine `FAILED`-Antwort oder gar keine Antwort zurückgegeben wird, schlägt der Vorgang fehl. Alle Ausgabedaten von der benutzerdefinierten Ressource werden am Speicherort der vorsignierten URL gespeichert. Der Vorlagenentwickler kann diese Daten mithilfe der GetAtt Funktion Fn:: abrufen.

> **Note**
>
> Wenn Sie verwenden AWS PrivateLink, müssen benutzerdefinierte Ressourcen in der VPC Zugriff auf CloudFormation -spezifische S3-Buckets haben. Benutzerdefinierte Ressourcen müssen Antworten an eine vorsignierte Amazon S3-URL senden. Wenn sie keine Antworten an Amazon S3 senden können, erhalten sie CloudFormation keine Antwort und der Stack-Vorgang schlägt fehl. Weitere Informationen finden Sie unter Zugriff CloudFormation über einen Schnittstellenendpunkt (AWS PrivateLink).

Timeout für die Antwort

Das Standard-Timeout für Ihre benutzerdefinierte Ressource beträgt 3600 Sekunden (1 Stunde). Wenn während dieser Zeit keine Antwort eingeht, schlägt der Stack-Vorgang fehl.

Sie können den Timeout-Wert anpassen, je nachdem, wie lange Sie erwarten, dass die Antwort von der benutzerdefinierten Ressource dauert. Wenn Sie beispielsweise eine benutzerdefinierte Ressource bereitstellen, die eine Lambda-Funktion aufruft, von der erwartet wird, dass sie innerhalb von fünf Minuten reagiert, können Sie in der Stack-Vorlage ein Timeout von fünf Minuten festlegen, indem Sie die Eigenschaft angeben. `ServiceTimeout` Weitere Informationen finden Sie unter Anforderungsobjekte für benutzerdefinierte Ressourcen. Auf diese Weise schlägt der Stack-Vorgang nach fünf Minuten fehl, wenn in der Lambda-Funktion ein Fehler auftritt, der dazu führt, dass sie hängen bleibt, CloudFormation anstatt die ganze Stunde zu warten.

Achten Sie jedoch darauf, den Timeout-Wert nicht zu niedrig einzustellen. Um unerwartete Timeouts zu vermeiden, stellen Sie sicher, dass Ihre benutzerdefinierte Ressource genügend Zeit hat, um die erforderlichen Aktionen durchzuführen und eine Antwort zurückzugeben.

Amazon SNS-gestützte benutzerdefinierte Ressourcen

Das folgende Thema zeigt Ihnen, wie Sie eine benutzerdefinierte Ressource mit einem Service-Token konfigurieren, das das Amazon SNS SNS-Thema angibt, CloudFormation an das Anfragen gesendet werden. Sie lernen auch die Reihenfolge der Ereignisse und Nachrichten kennen, die als Ergebnis der Erstellung, Aktualisierung und Löschung eines benutzerdefinierten Ressourcenstapels gesendet und empfangen wurden.

Bei benutzerdefinierten Ressourcen und Amazon SNS können Sie Szenarien wie das Hinzufügen neuer Ressourcen zu einem Stack und das Einfügen dynamischer Daten in einen Stack ermöglichen. Wenn Sie beispielsweise einen Stack erstellen, CloudFormation können Sie eine `Create` Anfrage an ein Thema senden, das von einer Anwendung überwacht wird, die auf einer EC2 Amazon-Instance ausgeführt wird. Die Amazon-SNS-Benachrichtigung löst aus, dass die Anwendung weitere Bereitstellungsaufgaben ausführt, z. B. das Abrufen eines Pools von erlaubten Elastic-IP-Adressen. Danach sendet die Anwendung eine Antwort (und alle Ausgabedaten), in der Sie aufgefordert werden, mit dem Stack-Vorgang CloudFormation fortzufahren.

Wenn Sie ein Amazon SNS SNS-Thema als Ziel einer benutzerdefinierten Ressource angeben, CloudFormation sendet bei Stack-Vorgängen, die die benutzerdefinierte Ressource betreffen, Nachrichten an das angegebene SNS-Thema. Um diese Nachrichten verarbeiten und die erforderlichen Aktionen ausführen zu können, muss ein unterstützter Endpunkt das SNS-Thema abonniert haben.

Eine Einführung in benutzerdefinierte Ressourcen und deren Funktionsweise finden Sie unter. Erstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen Informationen zu Amazon SNS und seiner Funktionsweise finden Sie im Amazon Simple Notification Service Developer Guide.

Verwenden von Amazon SNS zum Erstellen benutzerdefinierter Ressourcen

Themen

- Schritt 1: Erstellen des Stacks
- Schritt 2: Stack-Updates
- Schritt 3: Löschen des Stacks

Schritt 1: Erstellen des Stacks

1. Der Vorlagenentwickler erstellt einen CloudFormation Stapel, der eine benutzerdefinierte Ressource enthält.

   Im folgenden Vorlagenbeispiel verwenden wir den Namen des benutzerdefinierten Ressourcentyps `Custom::SeleniumTester` für die benutzerdefinierte Ressource mit logischer ID`MySeleniumTest`. Benutzerdefinierte Ressourcentypnamen müssen alphanumerisch sein und können maximal 60 Zeichen enthalten.

   Der benutzerdefinierte Ressourcentyp wird mit einem Diensttoken, optionalen anbieterspezifischen Eigenschaften und optionalen Fn:: GetAtt -Attributen deklariert, die vom benutzerdefinierten Ressourcenanbieter definiert werden. Diese Eigenschaften und Attribute können verwendet werden, um Informationen vom template developer an custom resource provider zu übergeben und umgekehrt. Das Service-Token gibt ein Amazon SNS SNS-Thema an, das der Ressourcenanbieter konfiguriert hat.

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "MySeleniumTest" : {
         "Type": "Custom::SeleniumTester",
         "Version" : "1.0",
         "Properties" : {
            "ServiceToken": "arn:aws:sns:us-west-2:123456789012:CRTest",
            "seleniumTester" : "SeleniumTest()",
            "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
 "http://search.mysite.com" ],
            "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
         }
      }
   },
   "Outputs" : {
      "topItem" : {
         "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
      },
      "numRespondents" : {
         "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
      }
   }
}
```

> **Note**
>
> Die Namen und Werte der Daten, mit `Fn::GetAtt` denen zugegriffen wird, werden vom benutzerdefinierten Ressourcenanbieter während der Antwort des Anbieters auf CloudFormation zurückgegeben. Wenn der custom resource provider von einem Drittanbieter stammt, muss der template developer die Namen dieser Rückgabewerte vom custom resource provider abrufen.

2. CloudFormation sendet eine Amazon SNS SNS-Benachrichtigung an den Ressourcenanbieter mit einer`"RequestType" : "Create"`, die Informationen über den Stack, die benutzerdefinierten Ressourceneigenschaften aus der Stack-Vorlage und eine S3-URL für die Antwort enthält.

   Das SNS-Thema, das zum Senden der Benachrichtigung verwendet wird, ist in die Vorlage in der `ServiceToken`-Eigenschaft eingebettet. Um die Verwendung eines hartcodierten Werts zu vermeiden, kann ein Vorlagen-Entwickler einen Vorlagenparameter verwenden, sodass der Wert zu dem Zeitpunkt eingegeben wird, wenn der Stack gelauncht wird.

   Das folgende Beispiel zeigt eine `Create`-Anforderung einer benutzerdefinierten Ressource, die einen benutzerdefinierten Ressourcentypnamen, `Custom::SeleniumTester`, enthält, der mit der `LogicalResourceId` `MySeleniumTester` erstellt wurde:

```
{
   "RequestType" : "Create",
   "ResponseURL" : "http://pre-signed-S3-url-for-response",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "RequestId" : "unique id for this create request",
   "ResourceType" : "Custom::SeleniumTester",
   "LogicalResourceId" : "MySeleniumTester",
   "ResourceProperties" : {
      "seleniumTester" : "SeleniumTest()",
      "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com" ],
      "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
   }
}
```

   Ausführliche Informationen zum Anforderungsobjekt für `Create` Anfragen finden Sie im Anfrage für CloudFormation benutzerdefinierte Ressourcen erstellen Thema.

3. Der custom resource provider verarbeitet die Daten, die von template developer gesendet werden und bestimmt, ob die `Create`-Anforderung erfolgreich war. Der Ressourcenanbieter verwendet dann die von gesendete S3-URL CloudFormation , um eine Antwort entweder `SUCCESS` oder zu senden`FAILED`.

   Je nach Antworttyp werden unterschiedliche Antwortfelder von CloudFormation erwartet. Informationen zu den Antwortfeldern für einen bestimmten Anforderungstyp finden Sie in der Dokumentation zu diesem Anforderungstyp im Anforderungstypen für benutzerdefinierte Ressourcen Abschnitt.

   Als Antwort auf eine Anforderung zum Erstellen oder Aktualisieren kann custom resource provider Datenelemente im Feld `Data` der Antwort zurückgeben. Dabei handelt es sich um Name-Wert-Paare und die Namen entsprechen den `Fn::GetAtt`-Attributen mit der benutzerdefinierten Ressource in der Stack-Vorlage. Die Werte sind die Daten, die zurückgegeben werden, wenn der Vorlagen-Entwickler `Fn::GetAtt` für die Ressource mit dem Attributnamen aufruft.

   Nachfolgend finden Sie eine Beispielantwort einer benutzerdefinierten Ressource:

```
{
   "Status" : "SUCCESS",
   "PhysicalResourceId" : "Tester1",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "RequestId" : "unique id for this create request",
   "LogicalResourceId" : "MySeleniumTester",
   "Data" : {
      "resultsPage" : "http://www.myexampledomain/test-results/guid",
      "lastUpdate" : "2012-11-14T03:30Z"
   }
}
```

   Ausführliche Informationen zum Antwortobjekt für `Create` Anfragen finden Sie im Anfrage für CloudFormation benutzerdefinierte Ressourcen erstellen Thema.

   Die `StackId`, `RequestId` und `LogicalResourceId`-Felder müssen wörtlich aus der Anforderung kopiert werden.

4. CloudFormation deklariert den Stack-Status als `CREATE_COMPLETE` oder`CREATE_FAILED`. Wenn der Stack erfolgreich erstellt wurde, kann der Vorlagenentwickler die Ausgabewerte der erstellten benutzerdefinierten Ressource verwenden, indem er mit Fn:: GetAtt auf sie zugreift.

Die benutzerdefinierte Ressourcenvorlage, die zur Veranschaulichung verwendet wird, verwendete `Fn::GetAtt` zum Kopieren von Ressourcenausgaben in die Stack-Ausgabe:

```
"Outputs" : {
   "topItem" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
   },
   "numRespondents" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
   }
}
```

Schritt 2: Stack-Updates

Um einen vorhandenen Stack zu aktualisieren, müssen Sie eine Vorlage übermitteln, die Updates für die Eigenschaften von Ressourcen im Stack angibt, wie im folgenden Beispiel dargestellt. CloudFormation aktualisiert nur die Ressourcen, für die Änderungen in der Vorlage angegeben sind. Weitere Informationen finden Sie unter Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen.

Sie können benutzerdefinierte Ressourcen aktualisieren, bei denen eine der zugrunde liegenden physischen Ressourcen ersetzt werden muss. Wenn Sie eine benutzerdefinierte Ressource in einer CloudFormation Vorlage aktualisieren, CloudFormation sendet eine Aktualisierungsanforderung an diese benutzerdefinierte Ressource. Wenn eine benutzerdefinierte Ressource ersetzt werden muss, muss die neu benutzerdefinierte Ressource eine Antwort mit der neuen physischen ID senden. When CloudFormation die Antwort empfängt, wird die `PhysicalResourceId` der alten und der neuen benutzerdefinierten Ressource verglichen. Wenn sie unterschiedlich sind, CloudFormation erkennt das Update als Ersatz und sendet eine Löschanfrage an die alte Ressource, wie unter gezeigtSchritt 3: Löschen des Stacks.

> **Note**
>
> Wenn Sie keine Änderungen an der benutzerdefinierten Ressource vorgenommen haben, CloudFormation werden während eines Stack-Updates keine Anfragen an sie gesendet.

1. Der template developer initiiert ein Update für den Stack, der eine benutzerdefinierte Ressource enthält. Während eines Update kann template developer neue Eigenschaften in der Stack-Vorlage angeben.

   Nachfolgend sehen Sie ein Beispiel für ein `Update` am der Stack-Vorlage mithilfe eines benutzerdefinierten Ressourcentyps:

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "MySeleniumTest" : {
         "Type": "Custom::SeleniumTester",
         "Version" : "1.0",
         "Properties" : {
            "ServiceToken": "arn:aws:sns:us-west-2:123456789012:CRTest",
            "seleniumTester" : "SeleniumTest()",
            "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
 "http://search.mysite.com",
               "http://mynewsite.com" ],
            "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
         }
      }
   },
   "Outputs" : {
      "topItem" : {
         "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
      },
      "numRespondents" : {
         "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
      }
   }
}
```

2. CloudFormation sendet eine Amazon SNS SNS-Benachrichtigung an den Ressourcenanbieter mit einer`"RequestType" : "Update"`, die ähnliche Informationen wie der `Create` Anruf enthält, außer dass das `OldResourceProperties` Feld die alten Ressourceneigenschaften und die aktualisierten (falls vorhanden) Ressourceneigenschaften ResourceProperties enthält.

   Es folgt ein Beispiel für eine `Update`-Anforderung.

```
{
   "RequestType" : "Update",
```

```
   "ResponseURL" : "http://pre-signed-S3-url-for-response",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "RequestId" : "uniqueid for this update request",
   "LogicalResourceId" : "MySeleniumTester",
   "ResourceType" : "Custom::SeleniumTester",
   "PhysicalResourceId" : "Tester1",
   "ResourceProperties" : {
      "seleniumTester" : "SeleniumTest()",
      "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com",
         "http://mynewsite.com" ],
      "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
   },
   "OldResourceProperties" : {
      "seleniumTester" : "SeleniumTest()",
      "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com" ],
      "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
   }
}
```

Ausführliche Informationen zum Anforderungsobjekt für `Update` Anfragen finden Sie im Aktualisierungsanfrage für CloudFormation benutzerdefinierte Ressourcen Thema.

3. Der benutzerdefinierte Ressourcenanbieter verarbeitet die von gesendeten Daten CloudFormation. Die benutzerdefinierte Ressource führt das Update durch und sendet eine Antwort entweder `SUCCESS` oder `FAILED` an die S3-URL. CloudFormation vergleicht dann `PhysicalResourceIDs` die alten und neuen benutzerdefinierten Ressourcen. Wenn sie unterschiedlich sind, CloudFormation erkennt, dass für das Update ein Ersatz erforderlich ist, und sendet eine Löschanforderung an die alte Ressource. Das folgende Beispiel zeigt die custom resource provider-Antwort auf eine `Update`-Anforderung.

```
{
   "Status" : "SUCCESS",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "RequestId" : "uniqueid for this update request",
   "LogicalResourceId" : "MySeleniumTester",
   "PhysicalResourceId" : "Tester2"
}
```

Ausführliche Informationen zum Antwortobjekt für `Update` Anfragen finden Sie im Aktualisierungsanfrage für CloudFormation benutzerdefinierte Ressourcen Thema.

Die `StackId`, `RequestId` und `LogicalResourceId`-Felder müssen wörtlich aus der Anforderung kopiert werden.

4. CloudFormation deklariert den Stack-Status als `UPDATE_COMPLETE` oder`UPDATE_FAILED`. Wenn das Update fehlschlägt, wird der Stack zurückgesetzt. Wenn der Stack erfolgreich aktualisiert wurde, kann der template developer auf neue Ausgabewerte der erstellten benutzerdefinierten mit `Fn::GetAtt` zugreifen.

Schritt 3: Löschen des Stacks

1. Der Vorlagen-Entwickler löscht einen Stack, der eine benutzerdefinierte Ressource enthält. CloudFormation ruft die aktuellen Eigenschaften, die in der Stack-Vorlage angegeben sind, zusammen mit dem SNS-Thema ab und bereitet sich darauf vor, eine Anforderung an den benutzerdefinierten Ressourcenanbieter zu stellen.
2. CloudFormation sendet eine Amazon SNS SNS-Benachrichtigung an den Ressourcenanbieter mit einer`"RequestType" : "Delete"`, die aktuelle Informationen über den Stack, die benutzerdefinierten Ressourceneigenschaften aus der Stack-Vorlage und eine S3-URL für die Antwort enthält.

   Immer wenn Sie einen Stapel löschen oder ein Update vornehmen, bei dem die benutzerdefinierte Ressource entfernt oder ersetzt wird, werden `PhysicalResourceId` die alten und neuen benutzerdefinierten Ressourcen CloudFormation verglichen. Wenn sie unterschiedlich sind, CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung für die alte Ressource (`OldPhysicalResource`), wie im folgenden Beispiel für eine `Delete` Anfrage gezeigt.

```
{
   "RequestType" : "Delete",
   "ResponseURL" : "http://pre-signed-S3-url-for-response",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "RequestId" : "unique id for this delete request",
   "ResourceType" : "Custom::SeleniumTester",
   "LogicalResourceId" : "MySeleniumTester",
   "PhysicalResourceId" : "Tester1",
   "ResourceProperties" : {
```

```
      "seleniumTester" : "SeleniumTest()",
      "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com",
         "http://mynewsite.com" ],
      "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
   }
}
```

Ausführliche Informationen zum Anforderungsobjekt für `Delete` Anfragen finden Sie im Anfrage für CloudFormation benutzerdefinierte Ressourcen löschen Thema.

`DescribeStackResource`, `DescribeStackResources` und `ListStackResources` zeigen den benutzerdefinierten Namen an, wenn er angegeben wurde.

3. Der benutzerdefinierte Ressourcenanbieter verarbeitet die von gesendeten Daten CloudFormation und stellt fest, ob die `Delete` Anfrage erfolgreich war. Der Ressourcenanbieter verwendet dann die von gesendete S3-URL CloudFormation , um eine Antwort entweder `SUCCESS` oder zu senden`FAILED`. Wenn Sie eine benutzerdefinierte Ressource löschen möchten, muss custom resource provider erfolgreich auf eine Löschenanfrage antworten.

   Das folgende Beispiel zeigt die custom resource provider-Antwort auf eine `Delete`-Anforderung.

```
{
   "Status" : "SUCCESS",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "RequestId" : "unique id for this delete request",
   "LogicalResourceId" : "MySeleniumTester",
   "PhysicalResourceId" : "Tester1"
}
```

   Ausführliche Informationen zum Antwortobjekt für `Delete` Anfragen finden Sie im Anfrage für CloudFormation benutzerdefinierte Ressourcen löschen Thema.

   Die `StackId`, `RequestId` und `LogicalResourceId`-Felder müssen wörtlich aus der Anforderung kopiert werden.

4. CloudFormation deklariert den Stack-Status als `DELETE_COMPLETE` oder`DELETE_FAILED`.

Lambda-gestützte benutzerdefinierte Ressourcen

Wenn Sie eine Lambda-Funktion einer benutzerdefinierten Ressource zuordnen, wird die Funktion immer dann aufgerufen, wenn die benutzerdefinierte Ressource erstellt, aktualisiert oder gelöscht wird. CloudFormation ruft eine Lambda-API auf, um die Funktion aufzurufen und alle Anforderungsdaten (z. B. Anforderungstyp und Ressourceneigenschaften) an die Funktion zu übergeben. Die Leistungsfähigkeit und Anpassungsfähigkeit von Lambda-Funktionen in Kombination mit CloudFormation ermöglichen eine Vielzahl von Szenarien, z. B. das dynamische Nachschlagen von AMI IDs während der Stack-Erstellung oder die Implementierung und Verwendung von Hilfsfunktionen wie String-Reversal-Funktionen.

Eine Einführung in benutzerdefinierte Ressourcen und deren Funktionsweise finden Sie unter. Erstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen

Themen

- Exemplarische Vorgehensweise: Erstellen Sie einen Verzögerungsmechanismus mit einer Lambda-gestützten benutzerdefinierten Ressource
- cfn-response-Modul

Exemplarische Vorgehensweise: Erstellen Sie einen Verzögerungsmechanismus mit einer Lambda-gestützten benutzerdefinierten Ressource

In dieser exemplarischen Vorgehensweise wird anhand einer Beispielvorlage gezeigt, wie Sie eine Lambda-gestützte benutzerdefinierte Ressource konfigurieren und starten. CloudFormation Diese Vorlage erstellt einen Verzögerungsmechanismus, der Stack-Bereitstellungen für eine bestimmte Zeit pausiert. Dies kann nützlich sein, wenn Sie bei der Bereitstellung von Ressourcen bewusst Verzögerungen einführen müssen, z. B. wenn Sie darauf warten, dass sich Ressourcen stabilisieren, bevor abhängige Ressourcen erstellt werden.

> **Note**
>
> Während zuvor Lambda-gestützte benutzerdefinierte Ressourcen für das Abrufen von AMI empfohlen wurden IDs, empfehlen wir jetzt die Verwendung von Parametern. AWS Systems Manager Dieser Ansatz macht Ihre Vorlagen wiederverwendbarer und einfacher zu verwalten. Weitere Informationen finden Sie unter Ruft einen Klartext-Wert aus dem Systems Manager Parameter Store ab.

Themen

- Übersicht
- Beispielvorlage
- Exemplarische Vorgehensweise für eine Beispielvorlage
- Voraussetzungen
- Den Stack starten
- Bereinigen von Ressourcen
- Ähnliche Informationen

Übersicht

Die in dieser exemplarischen Vorgehensweise verwendete Beispielstapelvorlage erstellt eine Lambda-gestützte benutzerdefinierte Ressource. Diese benutzerdefinierte Ressource führt zu einer konfigurierbaren Verzögerung (standardmäßig 60 Sekunden) bei der Stack-Erstellung. Die Verzögerung tritt bei Stack-Aktualisierungen nur auf, wenn die Eigenschaften der benutzerdefinierten Ressource geändert werden.

Die Vorlage stellt die folgenden Ressourcen bereit:

- eine benutzerdefinierte Ressource,
- eine Lambda-Funktion und
- eine IAM-Rolle, in die Lambda Logs schreiben kann. CloudWatch

Sie definiert auch zwei Ausgaben:

- Die tatsächliche Zeit, zu der die Funktion gewartet hat.
- Eine eindeutige Kennung, die bei jeder Ausführung der Lambda-Funktion generiert wird.

> **Note**
>
> CloudFormation ist ein kostenloser Dienst, aber Lambda berechnet Gebühren auf der Grundlage der Anzahl der Anfragen für Ihre Funktionen und der Zeit, in der Ihr Code ausgeführt wird. Weitere Informationen zu den Lambda-Preisen finden Sie unter AWS Lambda Preise.

Beispielvorlage

Nachfolgend finden Sie die von Lambda unterstützte benutzerdefinierte Ressourcenbeispielvorlage mit dem Verzögerungsmechanismus:

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "LambdaExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [{
            "Effect": "Allow",
            "Principal": { "Service": ["lambda.amazonaws.com"] },
            "Action": ["sts:AssumeRole"]
          }]
        },
        "Path": "/",
        "Policies": [{
          "PolicyName": "AllowLogs",
          "PolicyDocument": {
            "Statement": [{
              "Effect": "Allow",
              "Action": ["logs:*"],
              "Resource": "*"
            }]
          }
        }]
      }
    },
    "CFNWaiter": {
      "Type": "AWS::Lambda::Function",
      "Properties": {
        "Handler": "index.handler",
        "Runtime": "python3.9",
        "Timeout": 900,
        "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
        "Code": {
          "ZipFile": { "Fn::Join": ["\n", [
            "from time import sleep",
            "import json",
```

```
            "import cfnresponse",
            "import uuid",
            "",
            "def handler(event, context):",
            "  wait_seconds = 0",
            "  id = str(uuid.uuid1())",
            "  if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
            "    wait_seconds = int(event[\"ResourceProperties\"].get(\"WaitSeconds\",
 0))",
            "    sleep(wait_seconds)",
            "  response = {",
            "    \"TimeWaited\": wait_seconds,",
            "    \"Id\": id ",
            "  }",
            "  cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
 \"Waiter-\"+id)"
          ]]}
        }
      }
    },
    "CFNWaiterCustomResource": {
      "Type": "AWS::CloudFormation::CustomResource",
      "Properties": {
        "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
        "WaitSeconds": 60
      }
    }
  },
  "Outputs": {
    "TimeWaited": {
      "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "TimeWaited"] },
      "Export": { "Name": "TimeWaited" }
    },
    "WaiterId": {
      "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "Id"] },
      "Export": { "Name": "WaiterId" }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: "2010-09-09"
```

```
Resources:
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "lambda.amazonaws.com"
            Action:
              - "sts:AssumeRole"
      Path: "/"
      Policies:
        - PolicyName: "AllowLogs"
          PolicyDocument:
            Statement:
              - Effect: "Allow"
                Action:
                  - "logs:*"
                Resource: "*"
  CFNWaiter:
    Type: AWS::Lambda::Function
    Properties:
      Handler: index.handler
      Runtime: python3.9
      Timeout: 900
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile:
          !Sub |
          from time import sleep
          import json
          import cfnresponse
          import uuid

          def handler(event, context):
            wait_seconds = 0
            id = str(uuid.uuid1())
            if event["RequestType"] in ["Create", "Update"]:
              wait_seconds = int(event["ResourceProperties"].get("WaitSeconds", 0))
              sleep(wait_seconds)
            response = {
              "TimeWaited": wait_seconds,
```

```
              "Id": id
            }
            cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
 "Waiter-"+id)
  CFNWaiterCustomResource:
    Type: "AWS::CloudFormation::CustomResource"
    Properties:
      ServiceToken: !GetAtt CFNWaiter.Arn
      WaitSeconds: 60
Outputs:
  TimeWaited:
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited
    Export:
      Name: TimeWaited
  WaiterId:
    Value: !GetAtt CFNWaiterCustomResource.Id
    Export:
      Name: WaiterId
```

Exemplarische Vorgehensweise für eine Beispielvorlage

In den folgenden Ausschnitten werden relevante Teile der Beispielvorlage erläutert, damit Sie verstehen, wie die Lambda-Funktion mit einer benutzerdefinierten Ressource verknüpft ist, und um die Ausgabe zu verstehen.

AWS::Lambda::FunctionRessource `CFNWaiter`

Die `AWS::Lambda::Function` Ressource gibt den Quellcode, den Handlernamen, die Laufzeitumgebung und die Ausführungsrolle Amazon Resource Name (ARN) der Funktion an.

Die `Handler` Eigenschaft ist auf gesetzt, `index.handler` da sie einen Python-Quellcode verwendet. Weitere Informationen zu akzeptierten Handler-Identifikatoren bei der Verwendung von Quellcodes für Inline-Funktionen finden Sie unter AWS::Lambda::Function Code.

Das `Runtime` ist so angegeben, `python3.9` da die Quelldatei ein Python-Code ist.

Der `Timeout` ist auf 900 Sekunden eingestellt.

Die `Role` Eigenschaft verwendet die `Fn::GetAtt` Funktion, um den ARN der `LambdaExecutionRole` Ausführungsrolle abzurufen, die in der `AWS::IAM::Role` Ressource in der Vorlage deklariert ist.

Die `Code` Eigenschaft definiert den Funktionscode inline mithilfe einer Python-Funktion. Die Python-Funktion in der Beispielvorlage macht Folgendes:

- Erstellen Sie mit der UUID eine eindeutige ID
- Prüfen Sie, ob es sich bei der Anfrage um eine Anfrage zum Erstellen oder Aktualisieren handelt
- Schlafen Sie für die Dauer, die für `WaitSeconds` während `Create` oder `Update` Anfragen angegeben wurde
- Gibt die Wartezeit und die eindeutige ID zurück

JSON

```
...
    "CFNWaiter": {
      "Type": "AWS::Lambda::Function",
      "Properties": {
        "Handler": "index.handler",
        "Runtime": "python3.9",
        "Timeout": 900,
        "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
        "Code": {
          "ZipFile": { "Fn::Join": ["\n", [
            "from time import sleep",
            "import json",
            "import cfnresponse",
            "import uuid",
            "",
            "def handler(event, context):",
            "  wait_seconds = 0",
            "  id = str(uuid.uuid1())",
            "  if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
            "    wait_seconds = int(event[\"ResourceProperties\"].get(\"WaitSeconds\",
 0))",
            "    sleep(wait_seconds)",
            "  response = {",
            "    \"TimeWaited\": wait_seconds,",
            "    \"Id\": id ",
            "  }",
            "  cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
 \"Waiter-\"+id)"
          ]]}
```

```
        }
      }
    },
...
```

YAML

```
...
  CFNWaiter:
    Type: AWS::Lambda::Function
    Properties:
      Handler: index.handler
      Runtime: python3.9
      Timeout: 900
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile:
          !Sub |
          from time import sleep
          import json
          import cfnresponse
          import uuid

          def handler(event, context):
            wait_seconds = 0
            id = str(uuid.uuid1())
            if event["RequestType"] in ["Create", "Update"]:
              wait_seconds = int(event["ResourceProperties"].get("WaitSeconds", 0))
              sleep(wait_seconds)
            response = {
              "TimeWaited": wait_seconds,
              "Id": id
            }
            cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
 "Waiter-"+id)
...
```

AWS::IAM::RoleRessource `LambdaExecutionRole`

Die `AWS::IAM:Role` Ressource erstellt eine Ausführungsrolle für die Lambda-Funktion, die eine Richtlinie zur Übernahme der Rolle enthält, die es Lambda ermöglicht, sie zu verwenden. Sie enthält auch eine Richtlinie, die den Zugriff auf CloudWatch Protokolle ermöglicht.

JSON

```
...
    "LambdaExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [{
            "Effect": "Allow",
            "Principal": { "Service": ["lambda.amazonaws.com"] },
            "Action": ["sts:AssumeRole"]
          }]
        },
        "Path": "/",
        "Policies": [{
          "PolicyName": "AllowLogs",
          "PolicyDocument": {
            "Statement": [{
              "Effect": "Allow",
              "Action": ["logs:*"],
              "Resource": "*"
            }]
          }
        }]
      }
    },
...
```

YAML

```
...
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "lambda.amazonaws.com"
            Action:
              - "sts:AssumeRole"
      Path: "/"
```

```
      Policies:
        - PolicyName: "AllowLogs"
          PolicyDocument:
            Statement:
              - Effect: "Allow"
                Action:
                  - "logs:*"
                Resource: "*"
...
```

AWS::CloudFormation::CustomResourceRessource `CFNWaiterCustomResource`

Die benutzerdefinierte Ressource ist mit der Lambda-Funktion verknüpft, deren ARN verwendet `!GetAtt CFNWaiter.Arn` wird. Es wird eine Wartezeit von 60 Sekunden für Erstellungs- und Aktualisierungsvorgänge wie in `WaitSeconds` beschrieben implementiert. Die Ressource wird nur für einen Aktualisierungsvorgang aufgerufen, wenn die Eigenschaften geändert werden.

JSON

```
...
    "CFNWaiterCustomResource": {
      "Type": "AWS::CloudFormation::CustomResource",
      "Properties": {
        "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
        "WaitSeconds": 60
      }
    }
  },
...
```

YAML

```
...
  CFNWaiterCustomResource:
    Type: "AWS::CloudFormation::CustomResource"
    Properties:
      ServiceToken: !GetAtt CFNWaiter.Arn
      WaitSeconds: 60
...
```

`Outputs`

Die `Outputs` dieser Vorlage sind die `TimeWaited` und die`WaiterId`. Der `TimeWaited` Wert verwendet eine `Fn::GetAtt` Funktion, um anzugeben, wie lange die Kellnerressource tatsächlich gewartet hat. Der `WaiterId` verwendet eine `Fn::GetAtt` Funktion, um die eindeutige ID bereitzustellen, die generiert und der Ausführung zugeordnet wurde.

JSON

```
...
  "Outputs": {
    "TimeWaited": {
      "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "TimeWaited"] },
      "Export": { "Name": "TimeWaited" }
    },
    "WaiterId": {
      "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "Id"] },
      "Export": { "Name": "WaiterId" }
    }
  }
}
...
```

YAML

```
...
Outputs:
  TimeWaited:
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited
    Export:
      Name: TimeWaited
  WaiterId:
    Value: !GetAtt CFNWaiterCustomResource.Id
    Export:
      Name: WaiterId
...
```

Voraussetzungen

Sie benötigen IAM-Berechtigungen, um alle entsprechenden Dienste wie Lambda und verwenden zu können. CloudFormation

Den Stack starten

So erstellen Sie den Stack

1. Suchen Sie im Beispielvorlage Abschnitt nach der Vorlage Ihrer Wahl (YAML oder JSON) und speichern Sie sie mit dem Namen `samplelambdabackedcustomresource.template` auf Ihrem Computer.
2. Öffnen Sie die CloudFormation Konsole unter. https://console.aws.amazon.com/cloudformation/
3. Wählen Sie auf der Seite „Stacks“ oben rechts die Option „Stack erstellen“ und dann „Mit neuen Ressourcen (Standard)“ aus.
4. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Bestehende Vorlage auswählen aus.
5. Wählen Sie für Vorlage angeben die Option Vorlagendatei hochladen und anschließend Datei auswählen aus.
6. Wählen Sie die `samplelambdabackedcustomresource.template` Vorlagendatei aus, die Sie zuvor gespeichert haben.
7. Wählen Sie Weiter aus.
8. Geben Sie als Stapelname den Namen ein **`SampleCustomResourceStack`** und wählen Sie Weiter aus.
9. Für diese Anleitung müssen Sie keine Tags hinzufügen oder erweiterte Einstellungen angeben, wählen Sie daher Weiter aus.
10. Stellen Sie sicher, dass der Stack-Name korrekt aussieht, und wählen Sie dann Create aus.

Die Erstellung Ihres Stacks kann mehrere Minuten CloudFormation dauern. Betrachten Sie die Stack-Ereignisse, um den Fortschritt zu überwachen. Weitere Informationen finden Sie unter Stack-Informationen von der CloudFormation Konsole aus anzeigen.

Wenn die Stack-Erstellung erfolgreich ist, wurden alle Ressourcen im Stack, z. B. die Lambda-Funktion und die benutzerdefinierte Ressource, erstellt. Sie haben erfolgreich eine Lambda-Funktion und eine benutzerdefinierte Ressource verwendet.

Wenn die Lambda-Funktion einen Fehler zurückgibt, sehen Sie sich die Protokolle der Funktion in der CloudWatch Logs-Konsole an. Der Name des Protokollstreams ist die physische ID der benutzerdefinierten Ressource, die Sie durch Anzeigen der Ressourcen des Stacks finden können. Weitere Informationen finden Sie unter Protokolldaten anzeigen im CloudWatch Amazon-Benutzerhandbuch.

Bereinigen von Ressourcen

Löschen Sie den Stack, um alle Stack-Ressourcen zu bereinigen, die Sie erstellt haben, damit Ihnen keine unnötigen Ressourcen in Rechnung gestellt werden.

So löschen Sie den Stack

1. Wählen Sie in der CloudFormation Konsole den SampleCustomResourceStackStack aus.
2. Wählen Sie die Option Actions und anschließend Delete Stack aus.
3. Wählen Sie im Bestätigungsdialogfeld Yes, Delete aus.

Alle erstellten Ressourcen werden gelöscht.

Nachdem Sie nun verstanden haben, wie Lambda-gestützte benutzerdefinierte Ressourcen erstellt und verwendet werden, können Sie die Beispielvorlage und den Code aus dieser exemplarischen Vorgehensweise verwenden, um andere Stacks und Funktionen zu erstellen und mit ihnen zu experimentieren.

Ähnliche Informationen

- CloudFormation Referenz für benutzerdefinierte Ressourcen
- AWS::CloudFormation::CustomResource

`cfn-response`-Modul

In Ihrer CloudFormation Vorlage können Sie eine Lambda-Funktion als Ziel einer benutzerdefinierten Ressource angeben. Wenn Sie die `ZipFile` Eigenschaft verwenden, um den Quellcode Ihrer Funktion anzugeben, können Sie das `cfn-response` Modul laden, um Antworten von Ihrer Lambda-Funktion an eine benutzerdefinierte Ressource zu senden. Das `cfn-response` Modul ist eine Bibliothek, die das Senden von Antworten an die benutzerdefinierte Ressource vereinfacht, die Ihre Lambda-Funktion aufgerufen hat. Das Modul verfügt über eine `send` Methode, die ein Antwortobjekt über eine vorsignierte Amazon S3 S3-URL (die`ResponseURL`) an eine benutzerdefinierte Ressource sendet.

Das `cfn-response`-Modul ist nur verfügbar, wenn Sie die `ZipFile`-Eigenschaft verwenden, um Ihren Quell-Code zu schreiben. Es ist für Quellcode, der in Amazon S3-Buckets gespeichert wird, nicht verfügbar. Für Code in -Buckets müssen Sie eigene Funktionen schreiben um Antworten senden zu können.

> **Note**
>
> Nach dem Ausführen der `send`-Methode wird die Lambda-Funktion beendet, und alles, was Sie nach dieser Methode schreiben, wird ignoriert.

Laden des **`cfn-response`**-Moduls

Verwenden Sie bei Node.js-Funktionen die `require()`-Funktion, um das `cfn-response`-Modul zu laden. Im folgenden Codebeispiel wird ein `cfn-response`-Objekt mit dem Namen `response` erstellt:

```
var response = require('cfn-response');
```

Verwenden Sie bei Python die `import`-Anweisung, um das `cfnresponse`-Modul zu laden, wie im folgenden Beispiel veranschaulicht:

> **Note**
>
> Verwenden Sie genau dieses Import-Statement Wenn Sie andere Varianten desselben Import-Anweisung verwenden, beachtet CloudFormation das Antwortmodul nicht.

```
import cfnresponse
```

Parameter für die **`send`**-Methode

Mit der `send`-Methode können Sie die folgenden Parameter verwenden.

`event`

Die Felder in einer benutzerdefinierten Ressourcenanforderung.

`context`

Ein Objekt, das für Lambda-Funktionen festgelegt wurde und das Sie verwenden können, um festzulegen, wann die Funktion und alle Callbacks abschließend ausgeführt wurden, oder um aus der Lambda-Ausführungsumgebung auf Informationen zuzugreifen. Weitere Informationen finden Sie unter Erstellen von Lambda-Funktionen mit Node.js im AWS Lambda Entwicklerhandbuch.

`responseStatus`

Gibt an, ob die Funktion erfolgreich abgeschlossen wurde. Verwenden Sie die `cfnresponse`-Modulkonstanten, um den Status festzulegen: `SUCCESS` für erfolgreiche Ausführungen und `FAILED` für fehlgeschlagene Ausführungen.

`responseData`

Das `Data`-Feld für ein benutzerdefiniertes Ressourcen-Antwortobjekt. Die Daten sind eine Liste von Name-Wert-Paaren.

`physicalResourceId`

Optional. Die eindeutige Kennung der benutzerdefinierten Ressource, die die Funktion aufgerufen hat. Standardmäßig verwendet das Modul den Namen des Amazon CloudWatch Logs-Protokollstreams, der mit der Lambda-Funktion verknüpft ist.

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

`noEcho`

Optional. Gibt an, ob die Ausgabe der benutzerdefinierten Ressource maskiert wird, wenn sie mithilfe der Funktion `Fn::GetAtt` abgerufen wird. Wird „`true`“ festgelegt, werden alle zurückgegebenen Werte mit Sternchen (*****) maskiert, mit Ausnahme der Informationen, die an den folgenden Speicherorten gespeichert sind. Dieser Wert ist standardmäßig `false`.

> **Important**
>
> Durch die Verwendung des `NoEcho`-Attributs werden keine Informationen maskiert, die im Folgenden gespeichert sind:
>
> - Der `Metadata` Vorlagenbereich. CloudFormation transformiert, modifiziert oder redigiert keine Informationen, die Sie in den `Metadata` Abschnitt aufnehmen. Weitere Informationen finden Sie unter Metadaten.
> - Der `Outputs`-Vorlagenabschnitt. Weitere Informationen finden Sie unter Ausgaben:
> - Das `Metadata`-Attribut einer Ressourcendefinition. Weitere Informationen finden Sie unter `Metadata`-Attribut.

Es wird dringend empfohlen, diese Mechanismen nicht zu verwenden, um vertrauliche Informationen wie Passwörter oder Geheimnisse einzugeben.

Weitere Informationen zur Verwendung `NoEcho` zum Maskieren vertraulicher Informationen finden Sie in der Keine Anmeldeinformationen in Vorlagen einbetten bewährten Methode.

Beispiele

Node.js

Im folgenden Beispiel für Node.js verwendet die eingebettete Lambda-Funktion einen Eingabewert und multipliziert diesen mit 5. Eingebettete bzw. Inline-Funktionen sind besonders nützlich für kleinere Funktionen, da Sie den Quellcode hiermit direkt in einer Vorlage festlegen können, statt ein Paket zu erstellen und dieses in einen Amazon S3-Bucket hochzuladen. Die Funktion verwendet die `cfn-response` `send`-Methode zum Senden des Ergebnisses zurück an die benutzerdefinierte Ressource, die sie aufgerufen hat.

JSON

```
"ZipFile": { "Fn::Join": ["", [
  "var response = require('cfn-response');",
  "exports.handler = function(event, context) {",
  "  var input = parseInt(event.ResourceProperties.Input);",
  "  var responseData = {Value: input * 5};",
  "  response.send(event, context, response.SUCCESS, responseData);",
  "};"
]]}
```

YAML

```
ZipFile: >
  var response = require('cfn-response');
  exports.handler = function(event, context) {
    var input = parseInt(event.ResourceProperties.Input);
    var responseData = {Value: input * 5};
    response.send(event, context, response.SUCCESS, responseData);
  };
```

Python

Im folgenden Beispiel für Python verwendet die Lambda-Inline-Funktion einen Ganzzahlwert und multipliziert diesen mit 5.

JSON

```
"ZipFile" : { "Fn::Join" : ["\n", [
  "import json",
  "import cfnresponse",
  "def handler(event, context):",
  "   responseValue = int(event['ResourceProperties']['Input']) * 5",
  "   responseData = {}",
  "   responseData['Data'] = responseValue",
  "   cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
 \"CustomResourcePhysicalID\")"
]]}
```

YAML

```
ZipFile: |
  import json
  import cfnresponse
  def handler(event, context):
    responseValue = int(event['ResourceProperties']['Input']) * 5
    responseData = {}
    responseData['Data'] = responseValue
    cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
 "CustomResourcePhysicalID")
```

Modul-Quellcode

Themen

- Asynchroner Quellcode von Node.js
- Der Quellcode von Node.js
- Python Quellcode

Asynchroner Quellcode von Node.js

Im Folgenden finden Sie den Quellcode des Antwortmoduls für die Funktionen von Node.js, wenn der Handler asynchron ist. Überprüfen Sie den Code, um zu verstehen, was das Modul tut, und um die Implementierung in Ihre eigenen Antwortfunktionen zu vereinfachen.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0

exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

exports.send = function(event, context, responseStatus, responseData,
 physicalResourceId, noEcho) {

    return new Promise((resolve, reject) => {
        var responseBody = JSON.stringify({
            Status: responseStatus,
            Reason: "See the details in CloudWatch Log Stream: " +
 context.logStreamName,
            PhysicalResourceId: physicalResourceId || context.logStreamName,
            StackId: event.StackId,
            RequestId: event.RequestId,
            LogicalResourceId: event.LogicalResourceId,
            NoEcho: noEcho || false,
            Data: responseData
        });

        console.log("Response body:\n", responseBody);

        var https = require("https");
        var url = require("url");

        var parsedUrl = url.parse(event.ResponseURL);
        var options = {
            hostname: parsedUrl.hostname,
            port: 443,
            path: parsedUrl.path,
            method: "PUT",
            headers: {
                "content-type": "",
                "content-length": responseBody.length
            }
```

```
        };

        var request = https://request(options, function(response) {
            console.log("Status code: " + parseInt(response.statusCode));
            resolve(context.done());
        });

        request.on("error", function(error) {
            console.log("send(..) failed executing https.request(..): " +
 maskCredentialsAndSignature(error));
            reject(context.done(error));
        });

        request.write(responseBody);
        request.end();
    })
}

function maskCredentialsAndSignature(message) {
    return message.replace(/X-Amz-Credential=[^&\s]+/i, 'X-Amz-Credential=*****')
        .replace(/X-Amz-Signature=[^&\s]+/i, 'X-Amz-Signature=*****');
}
```

Der Quellcode von Node.js

Im Folgenden finden Sie den Quellcode des Antwortmoduls für die Funktionen von Node.js, falls der Handler nicht asynchron ist. Überprüfen Sie den Code, um zu verstehen, was das Modul tut, und um die Implementierung in Ihre eigenen Antwortfunktionen zu vereinfachen.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0

exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

exports.send = function(event, context, responseStatus, responseData,
 physicalResourceId, noEcho) {

    var responseBody = JSON.stringify({
        Status: responseStatus,
        Reason: "See the details in CloudWatch Log Stream: " + context.logStreamName,
        PhysicalResourceId: physicalResourceId || context.logStreamName,
        StackId: event.StackId,
```

```
        RequestId: event.RequestId,
        LogicalResourceId: event.LogicalResourceId,
        NoEcho: noEcho || false,
        Data: responseData
    });

    console.log("Response body:\n", responseBody);

    var https = require("https");
    var url = require("url");

    var parsedUrl = url.parse(event.ResponseURL);
    var options = {
        hostname: parsedUrl.hostname,
        port: 443,
        path: parsedUrl.path,
        method: "PUT",
        headers: {
            "content-type": "",
            "content-length": responseBody.length
        }
    };

    var request = https.request(options, function(response) {
        console.log("Status code: " + parseInt(response.statusCode));
        context.done();
    });

    request.on("error", function(error) {
        console.log("send(..) failed executing https.request(..): " +
 maskCredentialsAndSignature(error));
        context.done();
    });

    request.write(responseBody);
    request.end();
}
```

Python Quellcode

Im Folgenden finden Sie den Quellcode des Antwortmoduls für Python-Funktionen:

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
```

```
from __future__ import print_function
import urllib3
import json
import re

SUCCESS = "SUCCESS"
FAILED = "FAILED"

http = urllib3.PoolManager()


def send(event, context, responseStatus, responseData, physicalResourceId=None,
 noEcho=False, reason=None):
    responseUrl = event['ResponseURL']

    responseBody = {
        'Status' : responseStatus,
        'Reason' : reason or "See the details in CloudWatch Log Stream:
 {}".format(context.log_stream_name),
        'PhysicalResourceId' : physicalResourceId or context.log_stream_name,
        'StackId' : event['StackId'],
        'RequestId' : event['RequestId'],
        'LogicalResourceId' : event['LogicalResourceId'],
        'NoEcho' : noEcho,
        'Data' : responseData
    }

    json_responseBody = json.dumps(responseBody)

    print("Response body:")
    print(json_responseBody)

    headers = {
        'content-type' : '',
        'content-length' : str(len(json_responseBody))
    }

    try:
        response = http.request('PUT', responseUrl, headers=headers,
 body=json_responseBody)
        print("Status code:", response.status)
```

```
    except Exception as e:

        print("send(..) failed executing http.request(..):",
 mask_credentials_and_signature(e))


def mask_credentials_and_signature(message):
    message = re.sub(r'X-Amz-Credential=[^&\s]+', 'X-Amz-Credential=*****', message,
 flags=re.IGNORECASE)
    return re.sub(r'X-Amz-Signature=[^&\s]+', 'X-Amz-Signature=*****', message,
 flags=re.IGNORECASE)
```

Referenz für benutzerdefinierte Ressourcen

Eine Einführung in benutzerdefinierte Ressourcen und deren Funktionsweise finden Sie unterErstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen.

Dieser Abschnitt enthält Details zu folgenden Themen:

- Die JSON-Anforderung und die Antwortfelder in an und von CloudFormation gesendeten Nachrichten, wenn eine benutzerdefinierte Ressource bereitgestellt wird.
- Erwartete Felder für Anforderungen an und Antworten auf den custom resource provider als Reaktion auf Stack-Erstellung, Stack-Aktualisierungen und Stack-Löschung.

Themen

- Anforderungsobjekte für benutzerdefinierte Ressourcen
- Benutzerdefinierte Ressourcenantwortobjekte
- Anforderungstypen für benutzerdefinierte Ressourcen

Anforderungsobjekte für benutzerdefinierte Ressourcen

In diesem Thema werden die Eigenschaften des Anforderungsobjekts für eine CloudFormation benutzerdefinierte Ressource beschrieben.

Eine Einführung in benutzerdefinierte Ressourcen und deren Funktionsweise finden Sie unterErstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen.

Eigenschaften von Vorlagenentwickleranforderungen

Der Vorlagenentwickler verwendet die CloudFormation Ressource AWS::CloudFormation::CustomResource,, um eine benutzerdefinierte Ressource in einer Vorlage anzugeben.

In `AWS::CloudFormation::CustomResource` werden alle Eigenschaften durch custom resource provider definiert. Es gibt nur eine erforderliche Eigenschaft: `ServiceToken`.

`ServiceTimeout`

Die maximale Zeit in Sekunden, die vergehen kann, bis das Timeout eines benutzerdefinierten Ressourcenvorgangs überschritten wird.

Der Wert muss eine Ganzzahl zwischen 1 und 3600 sein. Der Standardwert ist 3600 Sekunden (1 Stunde).

Required: No

Typ: Zeichenfolge

`ServiceToken`

Das Service-Token, z. B. ein Amazon SNS SNS-Themen-ARN oder ein Lambda-Funktions-ARN. Das Service-Token muss aus derselben Region stammen wie der Stack.

Erforderlich: Ja

Typ: Zeichenfolge

Alle anderen Felder in den Ressourceneigenschaften sind optional und werden im Feld custom resource provider der Anforderung wortgetreu an den `ResourceProperties` gesendet. Der Anbieter definiert sowohl die Namen als auch die gültigen Inhalte für diese Felder.

Anforderungs-Felder für den benutzerdefinierten Ressourcenanbieter

Diese Felder werden in JSON-Anfragen von CloudFormation an den benutzerdefinierten Ressourcenanbieter im SNS-Thema gesendet, das der Anbieter für diesen Zweck konfiguriert hat.

`RequestType`

Der Anforderungstyp wird durch den CloudFormation Stack-Vorgang (Create-Stack, Update-Stack oder Delete-Stack) festgelegt, der vom Vorlagenentwickler für den Stack initiiert wurde, der die benutzerdefinierte Ressource enthält.

Zulässige Werte sind: `Create`, `Update` und `Delete`. Weitere Informationen finden Sie unter Anforderungstypen für benutzerdefinierte Ressourcen.

Erforderlich: Ja

Typ: Zeichenfolge

`ResponseURL`

Die Antwort-URL identifiziert einen vorsignierten S3-Bucket, der Antworten vom Anbieter der benutzerdefinierten Ressource auf empfängt. AWS CloudFormation

Erforderlich: Ja

Typ: Zeichenfolge

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält.

Kombinieren Sie `StackId` mit `RequestId` und es bildet sich ein Wert, mit dem Sie eine Anforderung für eine bestimmte benutzerdefinierte Ressource eindeutig identifizieren können.

Erforderlich: Ja

Typ: Zeichenfolge

`RequestId`

Eine eindeutige ID für die Anforderung.

Kombinieren Sie `StackId` mit `RequestId` und es bildet sich ein Wert, mit dem Sie eine Anforderung für eine bestimmte benutzerdefinierte Ressource eindeutig identifizieren können.

Erforderlich: Ja

Typ: Zeichenfolge

`ResourceType`

Der vom Developer der Vorlagen gewählte Ressourcentyp der benutzerdefinierten Ressource in der CloudFormation -Vorlage. Namen für benutzerdefinierte Ressourcentypen dürfen bis zu 60 Zeichen lang sein und sowohl alphanumerische Zeichen als auch folgende Zeichen enthalten: `_@-`.

Erforderlich: Ja

Typ: Zeichenfolge

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage. Er wird angegeben, um die Kommunikation zwischen dem benutzerdefinierten Ressourcenanbieter und dem Vorlagen-Entwickler zu erleichtern.

Erforderlich: Ja

Typ: Zeichenfolge

`PhysicalResourceId`

Eine erforderliche, vom custom resource provider definierte physische ID, die für diesen Anbieter eindeutig ist

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, AWS CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

Erforderlich: Sie wird immer mit Anforderungen des Typs `Update` und Anforderungen des Typs `Delete` gesendet, niemals jedoch mit Anforderungen des Typs `Create`.

Typ: Zeichenfolge

`ResourceProperties`

Dieses Feld enthält den Inhalt des `Properties`-Objekts, das vom Vorlagen-Entwickler gesendet wurde. Sein Inhalt wird durch benutzerdefinierte Ressourcenanbieter definiert.

Required: No

Typ: JSON-Objekt

`OldResourceProperties`

Wird nur für `Update`-Anforderungen verwendet. Enthält die Ressourceneigenschaften, die vor der Aktualisierungs-Anforderung deklariert wurden.

Erforderlich: Ja

Typ: JSON-Objekt

Benutzerdefinierte Ressourcenantwortobjekte

In diesem Thema werden die Eigenschaften des Antwortobjekts für eine CloudFormation benutzerdefinierte Ressource beschrieben.

Eine Einführung in benutzerdefinierte Ressourcen und deren Funktionsweise finden Sie unterErstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen.

Antwort-Felder für den benutzerdefinierten Ressourcenanbieter

Die folgenden Eigenschaften werden vom benutzerdefinierten Ressourcenanbieter eingeschlossen, wenn er die JSON-Datei an die vorsignierte URL sendet. Weitere Informationen zum Hochladen von Objekten mithilfe von presigned URLs finden Sie unter Uploading objects with presigned URLs im Amazon Simple Storage Service-Benutzerhandbuch.

> **Note**
>
> Die Gesamtgröße des Antworttexts darf 4 096 Bytes nicht überschreiten.

`Status`

Der Statuswert, der vom Anbieter benutzerdefinierter Ressourcen als Antwort auf eine -generierte Anfrage gesendet wurde. AWS CloudFormation

Es muss entweder `SUCCESS` oder `FAILED` sein.

Erforderlich: Ja

Typ: Zeichenfolge

`Reason`

Beschreibt den Grund für eine Reaktion auf einen Fehler.

Erforderlich: Erforderlich wenn `Status` `FAILED` ist. Er ist andernfalls optional.

Typ: Zeichenfolge

`PhysicalResourceId`

Dieser Wert sollte eine eindeutige Kennung für den Anbieter der benutzerdefinierten Ressource sein und kann bis zu 1 KB groß sein. Der Wert muss eine nicht leere Zeichenfolge sein und muss für alle Antworten für dieselbe Ressource identisch sein.

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, AWS CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

Erforderlich: Ja

Typ: Zeichenfolge

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

Erforderlich: Ja

Typ: Zeichenfolge

`RequestId`

Eine eindeutige ID für die Anforderung. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

Erforderlich: Ja

Typ: Zeichenfolge

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

Erforderlich: Ja

Typ: Zeichenfolge

`NoEcho`

Optional. Gibt an, ob die Ausgabe der benutzerdefinierten Ressource maskiert wird, wenn sie mithilfe der `Fn::GetAtt`-Funktion abgerufen wird. Wenn auf gesetzt`true`, werden alle zurückgegebenen Werte mit Sternchen (*****) maskiert, mit Ausnahme der Werte, die im *`Metadata`* Abschnitt der Vorlage gespeichert sind. AWS CloudFormation transformiert, modifiziert oder redigiert keine Informationen, die Sie in den Abschnitt aufnehmen. `Metadata` Der Standardwert ist `false`.

Weitere Informationen zum Maskieren vertraulicher Daten mit `NoEcho` enthält die bewährte Methode Keine Anmeldeinformationen in Ihre Vorlagen einbetten.

Required: No

Typ: Boolesch

`Data`

Optional. Die vom benutzerdefinierten Ressourcenanbieter definierten Namen-Wert-Paare, die mit der Antwort gesendet werden. Sie können auf die hier namentlich angegebenen Werte in der Vorlage mit `Fn::GetAtt` zugreifen.

> **Important**
>
> Wenn die Namen-Wert-Paare vertrauliche Informationen enthalten, sollten Sie das `NoEcho`-Feld benutzen, um die Ausgabe der benutzerdefinierten Ressource zu maskieren. Andernfalls sind die Werte durch die Eigenschaftswerte APIs dieser Oberfläche sichtbar (z. B.`DescribeStackEvents`).

Required: No

Typ: JSON-Objekt

Anforderungstypen für benutzerdefinierte Ressourcen

Der Anforderungstyp wird in dem `RequestType` Feld im Anforderungsobjekt gesendet, das gesendet wird, AWS CloudFormation wenn der Vorlagenentwickler einen Stapel erstellt, aktualisiert oder löscht, der eine benutzerdefinierte Ressource enthält.

Jeder Anforderungstyp verfügt über einen bestimmten Satz von Feldern, die zusammen mit der Anforderung gesendet werden, darunter eine Amazon-S3-URL für die Antwort durch den benutzerdefinierten Ressourcenanbieter. Der Anbieter muss auf den S3-Bucket entweder mit einem `SUCCESS` oder einem `FAILED` Ergebnis antworten, bevor der Timeout-Zeitraum endet. Wenn vor Ablauf des Timeout-Zeitraums keine Antwort eingeht, gilt die Anfrage als nicht erfolgreich und der Stack-Vorgang schlägt fehl. Jedes Ergebnis hat außerdem einen bestimmten Satz von Feldern, von denen erwartet wird. CloudFormation

Dieser Abschnitt enthält Informationen über die Anforderungs- und Antwortfelder mit Beispielen für jeden Anforderungstyp.

Themen

- Anfrage für CloudFormation benutzerdefinierte Ressourcen erstellen
- Anfrage für CloudFormation benutzerdefinierte Ressourcen löschen
- Aktualisierungsanfrage für CloudFormation benutzerdefinierte Ressourcen

Anfrage für CloudFormation benutzerdefinierte Ressourcen erstellen

Wenn der Vorlagenentwickler einen Stapel mit einer benutzerdefinierten Ressource erstellt, CloudFormation sendet er eine Anfrage mit der `RequestType` Einstellung auf an den Anbieter benutzerdefinierter Ressourcen`Create`. Diese Anfrage erfolgt speziell, wenn die benutzerdefinierte Ressource erstellt wird.

Eine Einführung in benutzerdefinierte Ressourcen und deren Funktionsweise finden Sie unterErstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen.

Anforderung

Erstellenanforderungen enthalten die folgenden Felder:

`RequestType`

`Create`.

`RequestId`

Eine eindeutige ID für die Anforderung.

`ResponseURL`

Die Antwort-URL identifiziert einen vorsignierten S3-Bucket, der Antworten vom Anbieter für benutzerdefinierte Ressourcen empfängt. AWS CloudFormation

`ResourceType`

Der vom Developer der Vorlagen gewählte Ressourcentyp der benutzerdefinierten Ressource in der CloudFormation -Vorlage. Namen für benutzerdefinierte Ressourcentypen dürfen bis zu 60 Zeichen lang sein und sowohl alphanumerische Zeichen als auch folgende Zeichen enthalten: `_@-`.

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage.

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält.

`ResourceProperties`

Dieses Feld enthält den Inhalt des `Properties`-Objekts, das vom Vorlagen-Entwickler gesendet wurde. Sein Inhalt wird durch benutzerdefinierte Ressourcenanbieter definiert.

Beispiel

```
{
   "RequestType" : "Create",
   "RequestId" : "unique id for this create request",
   "ResponseURL" : "pre-signed-url-for-create-response",
   "ResourceType" : "Custom::MyCustomResourceType",
   "LogicalResourceId" : "name of resource in template",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "ResourceProperties" : {
      "key1" : "string",
```

```
      "key2" : [ "list" ],
      "key3" : { "key4" : "map" }
   }
}
```

Antworten

Herzlichen Glückwunsch

Wenn die Erstellungs-Anforderung erfolgreich ist, muss eine Antwort an den Amazon-S3-Bucket gesendet werden, die die folgenden Felder enthält:

`Status`

Der Wert muss `SUCCESS` sein.

`RequestId`

Eine eindeutige ID für die Anforderung. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`PhysicalResourceId`

Dieser Wert sollte eine eindeutige Kennung für den Anbieter der benutzerdefinierten Ressource sein und kann bis zu 1 KB groß sein. Der Wert muss eine nicht leere Zeichenfolge sein und muss für alle Antworten für dieselbe Ressource identisch sein.

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, AWS CloudFormation erkennt das Update als Ersatz und sendet

eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

`NoEcho`

Optional. Gibt an, ob die Ausgabe der benutzerdefinierten Ressource maskiert wird, wenn sie mithilfe der `Fn::GetAtt`-Funktion abgerufen wird. Wenn auf gesetzt`true`, werden alle zurückgegebenen Werte mit Sternchen (*****) maskiert, mit Ausnahme der Werte, die im *`Metadata`* Abschnitt der Vorlage gespeichert sind. AWS CloudFormation transformiert, modifiziert oder redigiert keine Informationen, die Sie in den Abschnitt aufnehmen. `Metadata` Der Standardwert ist `false`.

Weitere Informationen zur Verwendung `NoEcho` zum Maskieren vertraulicher Informationen finden Sie in der Keine Anmeldeinformationen in Vorlagen einbetten bewährten Methode.

`Data`

Optional. Die vom benutzerdefinierten Ressourcenanbieter definierten Namen-Wert-Paare, die mit der Antwort gesendet werden. Sie können auf die hier namentlich angegebenen Werte in der Vorlage mit `Fn::GetAtt` zugreifen.

> **Important**
>
> Wenn die Namen-Wert-Paare vertrauliche Informationen enthalten, sollten Sie das `NoEcho`-Feld benutzen, um die Ausgabe der benutzerdefinierten Ressource zu maskieren. Andernfalls sind die Werte durch APIs die Eigenschaftswerte dieser Oberfläche sichtbar (z. B.`DescribeStackEvents`).

Beispiel

```
{
   "Status" : "SUCCESS",
   "RequestId" : "unique id for this create request (copied from request)",
   "LogicalResourceId" : "name of resource in template (copied from request)",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
   "PhysicalResourceId" : "required vendor-defined physical id that is unique for that
 vendor",
   "Data" : {
      "keyThatCanBeUsedInGetAtt1" : "data for key 1",
      "keyThatCanBeUsedInGetAtt2" : "data for key 2"
```

```
    }
}
```

Fehlgeschlagen

Wenn die Erstellenanfrage fehlschlägt, muss eine Antwort an den S3-Bucket gesendet werden, die die folgenden Felder enthält:

`Status`

Der Wert muss `FAILED` sein.

`Reason`

Beschreibt den Grund für eine Reaktion auf einen Fehler.

`RequestId`

Eine eindeutige ID für die Anforderung. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`PhysicalResourceId`

Dieser Wert sollte eine eindeutige Kennung für den Anbieter der benutzerdefinierten Ressource sein und kann bis zu 1 KB groß sein. Der Wert muss eine nicht leere Zeichenfolge sein und muss für alle Antworten für dieselbe Ressource identisch sein.

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, AWS CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

Beispiel

```
{
   "Status" : "FAILED",
   "Reason" : "Required failure reason string",
   "RequestId" : "unique id for this create request (copied from request)",
   "LogicalResourceId" : "name of resource in template (copied from request)",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
   "PhysicalResourceId" : "required vendor-defined physical id that is unique for that
 vendor"
}
```

Anfrage für CloudFormation benutzerdefinierte Ressourcen löschen

Wenn der Vorlagenentwickler den Stapel löscht oder die benutzerdefinierte Ressource aus dem Stapel entfernt, CloudFormation sendet er eine Anfrage an den Anbieter der benutzerdefinierten Ressourcen mit der `RequestType` Einstellung auf`Delete`. Wenn Sie eine benutzerdefinierte Ressource löschen möchten, muss custom resource provider erfolgreich auf eine Löschenanfrage antworten.

Eine Einführung in benutzerdefinierte Ressourcen und deren Funktionsweise finden Sie unterErstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen.

Anforderung

Anforderungen zum Löschen enthalten die folgenden Felder:

`RequestType`

`Delete`.

`RequestId`

Eine eindeutige ID für die Anforderung.

`ResponseURL`

Die Antwort-URL identifiziert einen vorsignierten S3-Bucket, der Antworten vom Anbieter für benutzerdefinierte Ressourcen empfängt. AWS CloudFormation

`ResourceType`

Der vom Developer der Vorlagen gewählte Ressourcentyp der benutzerdefinierten Ressource in der CloudFormation -Vorlage. Namen für benutzerdefinierte Ressourcentypen dürfen bis zu 60

Zeichen lang sein und sowohl alphanumerische Zeichen als auch folgende Zeichen enthalten: `_@-`.

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage.

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält.

`PhysicalResourceId`

Eine erforderliche, vom custom resource provider definierte physische ID, die für diesen Anbieter eindeutig ist

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, AWS CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

`ResourceProperties`

Dieses Feld enthält den Inhalt des `Properties`-Objekts, das vom Vorlagen-Entwickler gesendet wurde. Sein Inhalt wird durch benutzerdefinierte Ressourcenanbieter definiert.

Beispiel

```
{
   "RequestType" : "Delete",
   "RequestId" : "unique id for this delete request",
   "ResponseURL" : "pre-signed-url-for-delete-response",
   "ResourceType" : "Custom::MyCustomResourceType",
   "LogicalResourceId" : "name of resource in template",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "PhysicalResourceId" : "custom resource provider-defined physical id",
   "ResourceProperties" : {
      "key1" : "string",
```

```
      "key2" : [ "list" ],
      "key3" : { "key4" : "map" }
   }
}
```

Antworten

Herzlichen Glückwunsch

Wenn die Löschenanfrage erfolgreich ist, muss eine Antwort an den S3-Bucket gesendet werden, die die folgenden Felder enthält:

`Status`

Der Wert muss `SUCCESS` sein.

`RequestId`

Eine eindeutige ID für die Anforderung. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`PhysicalResourceId`

Dieser Wert sollte eine eindeutige Kennung für den Anbieter der benutzerdefinierten Ressource sein und kann bis zu 1 KB groß sein. Der Wert muss eine nicht leere Zeichenfolge sein und muss für alle Antworten für dieselbe Ressource identisch sein.

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, AWS CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

Beispiel

```
{
   "Status" : "SUCCESS",
   "RequestId" : "unique id for this delete request (copied from request)",
   "LogicalResourceId" : "name of resource in template (copied from request)",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
   "PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

Fehlgeschlagen

Wenn die Löschenanfrage fehlschlägt, muss eine Antwort an den S3-Bucket gesendet werden, die die folgenden Felder enthält:

`Status`

Der Wert muss `FAILED` sein.

`Reason`

Der Grund für den Fehlschlag.

`RequestId`

Der Wert `RequestId`, der aus der Löschenanfrage kopiert wurde.

`LogicalResourceId`

Der Wert `LogicalResourceId`, der aus der Löschenanfrage kopiert wurde.

`StackId`

Der Wert `StackId`, der aus der Löschenanfrage kopiert wurde.

`PhysicalResourceId`

Eine erforderliche, vom benutzerdefinierten Ressourcenanbieter definierte physische ID, die für diesen Anbieter eindeutig ist.

Beispiel

```
{
  "Status" : "FAILED",
```

```
  "Reason" : "Required failure reason string",
  "RequestId" : "unique id for this delete request (copied from request)",
  "LogicalResourceId" : "name of resource in template (copied from request)",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
  "PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

Aktualisierungsanfrage für CloudFormation benutzerdefinierte Ressourcen

Wenn der Vorlagenentwickler Änderungen an den Eigenschaften einer benutzerdefinierten Ressource innerhalb der Vorlage vornimmt und den Stapel aktualisiert, CloudFormation sendet er eine Anfrage an den Anbieter für benutzerdefinierte Ressourcen mit der `RequestType` Einstellung aufUpdate. Das bedeutet, dass Ihr benutzerdefinierter Ressourcencode keine Änderungen an Ressourcen erkennen muss, da er weiß, dass sich seine Eigenschaften geändert haben, wenn der Anforderungstyp `Update`

Eine Einführung in benutzerdefinierte Ressourcen und deren Funktionsweise finden Sie unterErstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen.

Anforderung

Aktualisierungsanforderungen enthalten die folgenden Felder:

`RequestType`

`Update`.

`RequestId`

Eine eindeutige ID für die Anforderung.

`ResponseURL`

Die Antwort-URL identifiziert einen vorsignierten S3-Bucket, der Antworten vom Anbieter für benutzerdefinierte Ressourcen empfängt. AWS CloudFormation

`ResourceType`

Der vom Developer der Vorlagen gewählte Ressourcentyp der benutzerdefinierten Ressource in der CloudFormation -Vorlage. Namen für benutzerdefinierte Ressourcentypen dürfen bis zu 60 Zeichen lang sein und sowohl alphanumerische Zeichen als auch folgende Zeichen enthalten: `_@-`. Bei einer Aktualisierung kann der Typ nicht geändert werden.

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage.

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält.

`PhysicalResourceId`

Eine erforderliche, vom custom resource provider definierte physische ID, die für diesen Anbieter eindeutig ist

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, AWS CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

`ResourceProperties`

Die neuen Eigenschaftswerte der Ressource, die vom Vorlagenentwickler in der aktualisierten CloudFormation Vorlage deklariert wurden.

`OldResourceProperties`

Die Ressourceneigenschaftswerte, die zuvor vom Vorlagenentwickler in der CloudFormation Vorlage deklariert wurden.

Beispiel

```
{
   "RequestType" : "Update",
   "RequestId" : "unique id for this update request",
   "ResponseURL" : "pre-signed-url-for-update-response",
   "ResourceType" : "Custom::MyCustomResourceType",
   "LogicalResourceId" : "name of resource in template",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
   "PhysicalResourceId" : "custom resource provider-defined physical id",
```

```
   "ResourceProperties" : {
      "key1" : "new-string",
      "key2" : [ "new-list" ],
      "key3" : { "key4" : "new-map" }
   },
   "OldResourceProperties" : {
      "key1" : "string",
      "key2" : [ "list" ],
      "key3" : { "key4" : "map" }
   }
}
```

Antworten

Herzlichen Glückwunsch

Wenn der benutzerdefinierte Ressourcenanbieter die Ressource erfolgreich aktualisieren kann, CloudFormation erwartet er, dass der Status `SUCCESS` in der Antwort auf gesetzt wird.

`Status`

Der Wert muss `SUCCESS` sein.

`RequestId`

Eine eindeutige ID für die Anforderung. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`PhysicalResourceId`

Dieser Wert sollte eine eindeutige Kennung für den Anbieter der benutzerdefinierten Ressource sein und kann bis zu 1 KB groß sein. Der Wert muss eine nicht leere Zeichenfolge sein und muss für alle Antworten für dieselbe Ressource identisch sein.

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, AWS CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

`NoEcho`

Optional. Gibt an, ob die Ausgabe der benutzerdefinierten Ressource maskiert wird, wenn sie mithilfe der `Fn::GetAtt`-Funktion abgerufen wird. Wenn auf gesetzt`true`, werden alle zurückgegebenen Werte mit Sternchen (*****) maskiert, mit Ausnahme der Werte, die im *`Metadata`* Abschnitt der Vorlage gespeichert sind. AWS CloudFormation transformiert, modifiziert oder redigiert keine Informationen, die Sie in den Abschnitt aufnehmen. `Metadata` Der Standardwert ist `false`.

Weitere Informationen zur Verwendung `NoEcho` zum Maskieren vertraulicher Informationen finden Sie in der Keine Anmeldeinformationen in Vorlagen einbetten bewährten Methode.

`Data`

Optional. Die vom benutzerdefinierten Ressourcenanbieter definierten Namen-Wert-Paare, die mit der Antwort gesendet werden. Sie können auf die hier namentlich angegebenen Werte in der Vorlage mit `Fn::GetAtt` zugreifen.

> **Important**
>
> Wenn die Namen-Wert-Paare vertrauliche Informationen enthalten, sollten Sie das `NoEcho`-Feld benutzen, um die Ausgabe der benutzerdefinierten Ressource zu maskieren. Andernfalls sind die Werte durch APIs die Eigenschaftswerte dieser Oberfläche sichtbar (z. B.`DescribeStackEvents`).

Beispiel

```
{
   "Status" : "SUCCESS",
   "RequestId" : "unique id for this update request (copied from request)",
   "LogicalResourceId" : "name of resource in template (copied from request)",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
```

```
   "PhysicalResourceId" : "custom resource provider-defined physical id",
   "Data" : {
      "keyThatCanBeUsedInGetAtt1" : "data for key 1",
      "keyThatCanBeUsedInGetAtt2" : "data for key 2"
   }
}
```

Fehlgeschlagen

Wenn die Ressource nicht mit einem neuen Satz von Eigenschaften aktualisiert werden kann, wird CloudFormation erwartet, dass der Status auf gesetzt wird`FAILED`, zusammen mit einer Fehlerursache in der Antwort.

`Status`

Der Wert muss `FAILED` sein.

`Reason`

Beschreibt den Grund für eine Reaktion auf einen Fehler.

`RequestId`

Eine eindeutige ID für die Anforderung. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der AWS CloudFormation -Vorlage. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält. Dieser Rückgabewert sollte wörtlich auf Basis der Anforderung kopiert werden.

`PhysicalResourceId`

Dieser Wert sollte eine eindeutige Kennung für den Anbieter der benutzerdefinierten Ressource sein und kann bis zu 1 KB groß sein. Der Wert muss eine nicht leere Zeichenfolge sein und muss für alle Antworten für dieselbe Ressource identisch sein.

Der für eine `PhysicalResourceId` zurückgegebene Wert kann benutzerdefinierte Vorgänge zur Ressourcenaktualisierung ändern. Wenn der zurückgegebene Wert

derselbe ist, wird es als normale Aktualisierung betrachtet. Wenn der zurückgegebene Wert anders ist, AWS CloudFormation erkennt das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter `AWS::CloudFormation::CustomResource`.

Beispiel

```
{
   "Status" : "FAILED",
   "Reason" : "Required failure reason string",
   "RequestId" : "unique id for this update request (copied from request)",
   "LogicalResourceId" : "name of resource in template (copied from request)",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
   "PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

Führen Sie eine benutzerdefinierte Verarbeitung von CloudFormation Vorlagen mit Vorlagenmakros durch

Mit Makros können Sie benutzerdefinierte Verarbeitungsvorgänge für Vorlagen durchführen, von einfachen Aktionen wie find-and-replace Operationen bis hin zu umfangreichen Transformationen ganzer Vorlagen.

Um sich ein Bild von der Bandbreite der Möglichkeiten zu machen, sollten Sie sich die `AWS::Serverless` Transformationen `AWS::Include` und die Transformationen ansehen. Dabei handelt es sich um Makros, die von folgenden Anbietern bereitgestellt werden: CloudFormation

- AWS::Includetransform ermöglicht es Ihnen, Textbausteine von Vorlagen in Ihre Vorlagen einzufügen.
- AWS::Serverlesstransform verwendet eine gesamte Vorlage, die in der AWS SAM-Syntax (AWS Serverless Application Model) geschrieben ist, transformiert und erweitert sie in eine konforme Vorlage. CloudFormation Weitere Informationen zu serverlosen Anwendungen finden Sie AWS Serverless Application Model im AWS SAM Entwicklerhandbuch.

Themen

- Fakturierung

- Makrobeispiele
- Zugehörige Ressourcen
- Überblick über CloudFormation Makros
- Erstellen Sie eine CloudFormation Makrodefinition
- Beispiel für ein einfaches Makro zum Ersetzen von Zeichenketten
- Fehler bei der verarbeiteten Vorlage beheben

Fakturierung

Wenn ein Makro ausgeführt wird, werden dem Besitzer der Lambda-Funktion alle mit der Ausführung dieser Funktion verbundenen Kosten in Rechnung gestellt.

Die `AWS::Serverless` Transformationen `AWS::Include` und sind Makros, die von gehostet werden. CloudFormation Ihre Nutzung ist kostenlos.

Makrobeispiele

Zusätzlich zu den Beispielen in diesem Abschnitt finden Sie in unserem Repository Beispielmakros, einschließlich Quellcode und Vorlagen. GitHub Diese Beispiele werden "wie besehen" für Schulungszwecke zur Verfügung gestellt.

Zugehörige Ressourcen

- AWS::CloudFormation::Macro
- CloudFormation TransformAbschnitt „Vorlage“
- Fn::Transform
- AWS::Serverless-Transformation
- AWS::Include-Transformation

Überblick über CloudFormation Makros

Die Bearbeitung von Vorlagen mit Makros besteht aus zwei wesentlichen Schritten: Erstellen des Makros selbst und dann Verwenden des Makros, um die Verarbeitung Ihrer Vorlagen durchzuführen.

Um eine Makrodefinition zu erstellen, müssen Sie Folgendes erstellen:

- Eine Lambda-Funktion zur Durchführung der Vorlagenverarbeitung. Diese Lambda-Funktion akzeptiert entweder einen Ausschnitt oder eine ganze Vorlage und alle zusätzlichen Parameter, die Sie definieren. Sie gibt den bearbeiteten Vorlagenausschnitt oder die gesamte Vorlage als Antwort zurück.
- Eine Ressource vom Typ AWS::CloudFormation::Macro, die es Benutzern ermöglicht, die Lambda-Funktion aus CloudFormation Vorlagen heraus aufzurufen. Diese Ressource spezifiziert den ARN der Lambda-Funktion, die für dieses Makro aufgerufen werden soll, und zusätzliche optionale Eigenschaften, die beim Debuggen helfen. Um diese Ressource in einem Konto zu erstellen, erstellen Sie eine Vorlage, die die `AWS::CloudFormation::Macro` Ressource enthält, und erstellen Sie dann entweder einen Stack oder ein Stack-Set mit selbstverwalteten Berechtigungen aus der Vorlage. AWS CloudFormation StackSetsunterstützt derzeit nicht das Erstellen oder Aktualisieren von Stack-Sets mit vom Dienst verwalteten Berechtigungen anhand von Vorlagen, die auf Makros verweisen.

Um ein Makro zu verwenden, referenzieren Sie das Makro in Ihrer Vorlage:

- Um einen Abschnitt oder Teil einer Vorlage zu verarbeiten, verweisen Sie auf das Makro in einer `Fn::Transform` Funktion, die sich relativ zum Vorlageninhalt befindet, den Sie transformieren möchten. Bei Verwendung von `Fn::Transform` können Sie außerdem beliebige Parameter übergeben.
- Um eine gesamte Vorlage zu bearbeiten, referenzieren Sie das Makro im Abschnitt Transform der Vorlage.

Als Nächstes erstellen Sie normalerweise einen Änderungssatz und führen diesen aus. (Durch die Verarbeitung von Makros können mehrere Ressourcen hinzugefügt werden, die Ihnen möglicherweise nicht bekannt sind. Um sicherzustellen, dass Sie alle durch Makros eingeführten Änderungen kennen, wird dringend empfohlen, Änderungssätze zu verwenden.) CloudFormation übergibt den angegebenen Vorlageninhalt zusammen mit allen zusätzlich angegebenen Parametern an die in der Makro-Ressource angegebene Lambda-Funktion. Die Lambda-Funktion gibt den bearbeiteten Vorlageninhalt zurück, sei es ein Ausschnitt oder eine ganze Vorlage.

Nachdem alle Makros in der Vorlage aufgerufen wurden, wird ein Änderungssatz CloudFormation generiert, der den verarbeiteten Vorlageninhalt enthält. Nachdem Sie den Änderungssatz überprüft haben, führen Sie ihn aus, um die Änderungen anzuwenden.

Wie erstellt man Stacks direkt

Um einen Stack mit einer Vorlage zu erstellen oder zu aktualisieren, die auf Makros verweist, erstellen Sie normalerweise einen Änderungssatz und führen ihn dann aus. Ein Änderungssatz beschreibt die Aktionen, die CloudFormation basierend auf der verarbeiteten Vorlage ausführen wird. Verarbeitungsmakros können mehrere Ressourcen hinzufügen, die Ihnen möglicherweise nicht bekannt sind. Um sicherzustellen, dass Sie alle Änderungen kennen, die durch Makros eingeführt werden, wird dringend die Verwendung von Änderungssätzen empfohlen. Nachdem Sie den Änderungssatz überprüft haben, können Sie ihn ausführen, um die Änderungen tatsächlich anzuwenden.

Ein Makro kann Ihrer Vorlage IAM-Ressourcen hinzufügen. Für diese Ressourcen erfordert CloudFormation, dass Sie ihre Funktionen angeben. Da Sie vor der Verarbeitung Ihrer Vorlage nicht wissen CloudFormation können, welche Ressourcen hinzugefügt wurden, müssen Sie bei der Erstellung des Änderungssatzes möglicherweise die IAM-Funktionen bestätigen, je nachdem, ob die referenzierten Makros IAM-Ressourcen enthalten. Wenn Sie den Änderungssatz ausführen, CloudFormation verfügt er auf diese Weise über die erforderlichen Funktionen, um IAM-Ressourcen zu erstellen.

Um einen Stack direkt aus einer verarbeiteten Vorlage zu erstellen oder zu aktualisieren, ohne zuerst die vorgeschlagenen Änderungen in einem Änderungssatz zu überprüfen, geben Sie die `CAPABILITY_AUTO_EXPAND` Funktion während einer `CreateStack` `UpdateStack` Oder-Anfrage an. Sie sollten Stacks nur dann direkt von einer Stack-Vorlage mit Makros erstellen, wenn Sie wissen, welche Verarbeitungsschritte das Makro ausführt. Sie können Änderungssätze nicht mit Stack-Set-Makros verwenden; Sie müssen Ihr Stack-Set direkt aktualisieren.

Weitere Informationen finden Sie unter CreateStackoder UpdateStackin der AWS CloudFormation API-Referenz.

> **Important**
>
> Wenn Ihre Stack-Set-Vorlage auf ein oder mehrere Makros verweist, müssen Sie das Stack-Set direkt aus der verarbeiteten Vorlage erstellen, ohne vorher die resultierenden Änderungen in einem Änderungssatz zu überprüfen. Verarbeitungsmakros können mehrere Ressourcen hinzufügen, die Ihnen möglicherweise nicht bekannt sind. Bevor Sie einen Stack-Set aus einer Vorlage erstellen oder aktualisieren, die direkt auf Makros verweist, sollten Sie sicherstellen, dass Sie wissen, welche Verarbeitung die Makros ausführen.

Um die Anzahl der Schritte beim Starten von Stacks aus Vorlagen, die auf Makros verweisen, zu reduzieren, können Sie die Befehle `package` und `deploy` AWS CLI verwenden. Weitere Informationen erhalten Sie unter Laden Sie lokale Artefakte in einen S3-Bucket hoch mit dem AWS CLI und Erstellen Sie einen Stapel, der Transformationen enthält.

Überlegungen

Beachten Sie bei der Arbeit mit Makros die folgenden Hinweise und Einschränkungen:

- Makros werden nur dort unterstützt AWS-Regionen , wo Lambda verfügbar ist. Eine Liste der Regionen, in denen Lambda verfügbar ist, finden Sie unter AWS Lambda Endpunkte und Kontingente.
- Alle verarbeiteten Vorlagenausschnitte müssen gültiges -JSON sein.
- Alle verarbeiteten Vorlagenausschnitte müssen Validierungsprüfungen für eine Stack-Erstellung, Stack-Aktualisierung, Stack-Set-Erstellung oder einen Stack-Set-Aktualisierungs-Vorgang durchlaufen.
- CloudFormation löst zuerst Makros auf und verarbeitet dann die Vorlage. Die resultierende Vorlage muss gültiger JSON-Code sein und darf nicht größer als das Vorlagengrößenlimit sein.
- Aufgrund der Reihenfolge, in der Elemente in einer Vorlage CloudFormation verarbeitet werden, kann ein Makro keine Module in den verarbeiteten Vorlageninhalt aufnehmen, zu dem es zurückkehrt. CloudFormation Weitere Informationen finden Sie unter Reihenfolge der Makro-Evaluierung.
- Wenn Sie die Funktion zum Aktualisieren des Rollbacks verwenden, CloudFormation wird eine Kopie der Originalvorlage verwendet. Es wird ein Rollback auf die ursprüngliche Vorlage ausgeführt, selbst wenn der Codeausschnitt geändert wurde.
- Das Einfügen von Makros in Makros funktioniert nicht, da wir Makros nicht rekursiv verarbeiten.
- Die intrinsische `Fn::ImportValue`-Funktion wird derzeit in Makros nicht unterstützt.
- Die in der Vorlage enthaltenen intrinsischen Funktionen werden nach allen Makros ausgewertet. Daher können die von Ihrem Makro zurückgegebenen verarbeiteten Vorlageninhalte Aufrufe von intrinsischen Funktionen beinhalten, die wie gewohnt ausgewertet werden.
- StackSets unterstützt derzeit nicht das Erstellen oder Aktualisieren von Stacksets mit vom Dienst verwalteten Berechtigungen anhand von Vorlagen, die auf Makros verweisen. CloudFormation

Umfang und Berechtigungen für Makrokonten

Makros können Sie nur in dem Konto verwenden, in dem sie als Ressource angelegt wurden. Der Name des Makros muss innerhalb eines bestimmten Kontos eindeutig sein. Sie können jedoch die gleiche Funktionalität in mehreren Konten zur Verfügung stellen, indem Sie den kontenübergreifenden Zugriff auf die zugrundeliegende Lambda-Funktion aktivieren und dann Makrodefinitionen erstellen, die sich in mehreren Konten auf diese Funktion beziehen. Im folgenden Beispiel enthalten drei Konten Makrodefinitionen, die jeweils auf dieselbe Lambda-Funktion verweisen.

Um eine Makrodefinition zu erstellen, muss der Benutzer die Berechtigung haben, einen Stack oder Stack-Set innerhalb des angegebenen Kontos zu erstellen.

CloudFormation Um ein in einer Vorlage enthaltenes Makro erfolgreich ausführen zu können, muss der Benutzer über `Invoke` Berechtigungen für die zugrunde liegende Lambda-Funktion verfügen. Um eine mögliche Eskalation von Berechtigungen zu verhindern, gibt sich bei der CloudFormation Ausführung des Makros die Identität des Benutzers aus.

Weitere Informationen finden Sie unter Verwaltung von Berechtigungen AWS Lambda im AWS Lambda Entwicklerhandbuch und unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Lambda im Service Authorization Reference.

Erstellen Sie eine CloudFormation Makrodefinition

Wenn Sie eine Makrodefinition erstellen, macht die Makrodefinition die zugrunde liegende Lambda-Funktion im angegebenen Konto verfügbar, sodass sie zur Verarbeitung der Vorlagen CloudFormation aufgerufen wird.

Zuordnung von Ereignissen

Wenn die Lambda-Funktion eines Makros CloudFormation aufgerufen wird, sendet es eine Anfrage im JSON-Format mit der folgenden Struktur:

```
{
    "region" : "us-east-1",
    "accountId" : "$ACCOUNT_ID",
    "fragment" : { ... },
    "transformId" : "$TRANSFORM_ID",
    "params" : { ... },
```

```
    "requestId" : "$REQUEST_ID",
    "templateParameterValues" : { ... }
}
```

- `region`

  Die Region, in der sich das Makro befindet.

- `accountId`

  Die Konto-ID des Kontos, von dem aus das Makro die Lambda-Funktion aufruft.

- `fragment`

  Der Vorlageninhalt, der für die benutzerdefinierte Verarbeitung zur Verfügung steht, im JSON-Format.

  - Bei Makros, die im Vorlagenabschnitt `Transform` enthalten sind, ist dies die gesamte Vorlage mit Ausnahme des Abschnitts `Transform`.
  - Bei Makros, die in einem intrinsischen `Fn::Transform`-Funktionsaufruf enthalten sind, sind dies alle Geschwisterknoten (und ihre Unterknoten), basierend auf der Position der intrinsischen Funktion innerhalb der Vorlage, mit Ausnahme der `Fn::Transform` Funktion. Weitere Informationen finden Sie unter Geltungsbereich der Makrovorlage.

- `transformId`

  Der Name des Makros, das diese Funktion aufruft.

- `params`

  Bei `Fn::Transform` Funktionsaufrufen alle angegebenen Parameter für die Funktion. CloudFormation wertet diese Parameter nicht aus, bevor sie an die Funktion übergeben werden.

  Für Makros, die im Vorlagenabschnitt `Transform` enthalten sind, ist dieser Abschnitt leer.

- `requestId`

  Die ID der Anforderung, die diese Funktion aufruft.

- `templateParameterValues`

  Alle im Parameters Abschnitt der Vorlage angegebenen Parameter. CloudFormation wertet diese Parameter aus, bevor sie an die Funktion übergeben werden.

Reaktionsformat

CloudFormation erwartet, dass die Lambda-Funktion eine Antwort im folgenden JSON-Format zurückgibt:

```
{
    "requestId" : "$REQUEST_ID",
    "status" : "$STATUS",
    "fragment" : { ... },
    "errorMessage": "optional error message for failures"
}
```

- `requestId`

  Die ID der Anforderung, die diese Funktion aufruft. Diese muss mit der Anforderungs-ID übereinstimmen, die CloudFormation beim Aufrufen der Funktion angegeben wurde.

- `status`

  Der Status der Anforderung (Groß- und Kleinschreibung wird nicht berücksichtigt). Sollte auf gesetzt sein. `success` CloudFormation behandelt jede andere Antwort als Fehler.

- `fragment`

  Der Inhalt der verarbeiteten Vorlage CloudFormation , der in die verarbeitete Vorlage aufgenommen werden soll, einschließlich gleichgeordneter Elemente. CloudFormation ersetzt den Vorlageninhalt, der an die Lambda-Funktion übergeben wird, durch das Vorlagenfragment, das es in der Lambda-Antwort empfängt.

  Der Inhalt der verarbeiteten Vorlage muss gültiges JSON sein und seine Aufnahme in die verarbeitete Vorlage muss zu einer gültigen Vorlage führen.

  Wenn Ihre Funktion den Vorlageninhalt, den CloudFormation an sie übergibt, nicht verändert, Sie diesen Inhalt aber dennoch in die bearbeitete Vorlage aufnehmen müssen, muss Ihre Funktion diesen Vorlageninhalt in ihrer Antwort an CloudFormation zurückgeben.

- `errorMessage`

  Die Fehlermeldung, die erklärt, warum die Transformation fehlgeschlagen ist. CloudFormation zeigt diese Fehlermeldung im Bereich Ereignisse der Seite Stack-Details für Ihren Stack an.

  Zum Beispiel:

```
Error creating change set: Transform
                           AWS-Konto account
                           number::macro name failed with:
                           error message string.
```

Erstellen Sie eine Makrodefinition

Um eine CloudFormation Makrodefinition zu erstellen

1. Erstellen Sie eine Lambda-Funktion, die die Verarbeitung von Vorlageninhalten übernimmt. Sie kann jeden Teil einer Vorlage bis hin zur gesamten Vorlage verarbeiten.
2. Erstellen Sie eine CloudFormation Vorlage, die einen `AWS::CloudFormation::Macro` Ressourcentyp enthält, und geben Sie die `FunctionName` Eigenschaften `Name` und an. Die `FunctionName` Eigenschaft muss den ARN der Lambda-Funktion enthalten, die aufgerufen werden soll, wenn das Makro CloudFormation ausgeführt wird.
3. (Optional) Um das Debuggen zu erleichtern, können Sie auch die `LogRoleArn` Eigenschaften `LogGroupName` und angeben, wenn Sie den `AWS::CloudFormation::Macro` Ressourcentyp für Ihr Makro erstellen. Mit diesen Eigenschaften können Sie die Protokollgruppe CloudWatch Logs angeben, an die Fehlerprotokollierungsinformationen CloudFormation gesendet werden, wenn die dem Makro zugrunde liegende Lambda-Funktion aufgerufen wird, und die Rolle CloudFormation sollte beim Senden von Logeinträgen an diese Logs übernommen werden.
4. Erstellen Sie mithilfe der Vorlage einen Stack mit dem Makro in dem Konto, in dem Sie es verwenden möchten. Oder erstellen Sie ein Stack-Set mit selbstverwalteten Berechtigungen, indem Sie die Vorlage mit dem Makro im Administratorkonto verwenden, und erstellen Sie dann Stack-Instanzen in den Zielkonten.
5. Nachdem CloudFormation die Stacks, die die Makrodefinition enthalten, erfolgreich erstellt wurden, ist das Makro für die Verwendung in diesen Konten verfügbar. Sie verwenden ein Makro, indem Sie es in einer Vorlage an der Stelle referenzieren, die für die zu bearbeitenden Vorlageninhalte relevant ist.

Geltungsbereich der Makrovorlage

Makros, die im Abschnitt `Transform` einer Vorlage referenziert werden, können den gesamten Inhalt dieser Vorlage verarbeiten.

Makros, die in einer `Fn::Transform`-Funktion referenziert werden, können den Inhalt eines beliebigen der Geschwisterelemente (einschließlich untergeordneter) der `Fn::Transform`-Funktion in der Vorlage verarbeiten.

Beispielsweise kann `AWS::Include` im untenstehenden Vorlagenbeispiel die `MyBucket`-Eigenschaften verarbeiten, basierend auf der Position der `Fn::Transform`-Funktion, die sie enthält. `MyMacro` kann den Inhalt der gesamten Vorlage verarbeiten, da sie in den Abschnitt `Transform` aufgenommen wurde.

```
# Start of processable content for MyMacro
AWSTemplateFormatVersion: 2010-09-09
 Transform: [MyMacro]
 Resources:
    WaitCondition:
      Type: AWS::CloudFormation::WaitCondition
    MyBucket:
      Type: AWS::S3::Bucket
      # Start of processable content for AWS::Include
      Properties:
        BucketName: amzn-s3-demo-bucket1
        Tags: [{"key":"value"}]
        'Fn::Transform':
          - Name: 'AWS::Include'
              Parameters:
                Location: s3://amzn-s3-demo-bucket2/MyFileName.yaml
        CorsConfiguration: []
        # End of processable content for AWS::Include
    MyEc2Instance:
      Type: AWS::EC2::Instance
      Properties:
        ImageID: ami-1234567890abcdef0
# End of processable content for MyMacro
```

Reihenfolge der Makro-Evaluierung

Sie können in einer bestimmten Vorlage auf mehrere Makros verweisen, einschließlich Transformationen, die von bereitgestellt werden CloudFormation, wie AWS::Includez. B. und. AWS::Serverless

Makros werden so ausgewertet, dass sie aufgrund ihrer Position in der Vorlage von den am tiefsten verschachtelten nach außen bis zu den allgemeinsten reichen. Makros an derselben Stelle in der Vorlage werden seriell anhand der Reihenfolge, in der sie aufgelistet sind, ausgewertet.

Transformationen wie `AWS::Include` und `AWS::Transform` werden in Bezug auf die Reihenfolge und den Anwendungsbereich wie alle anderen Makros behandelt.

Im nachfolgenden Vorlagenbeispiel wird beispielsweise das `PolicyAdder` Makro zuerst CloudFormation ausgewertet, da es sich dabei um das am tiefsten verschachtelte Makro in der Vorlage handelt. CloudFormation führt dann `MyMacro` vor der Auswertung eine Auswertung durch, `AWS::Serverless` da es bereits im Abschnitt aufgeführt ist. `AWS::Serverless` `Transform`

```
AWSTemplateFormatVersion: 2010-09-09
 Transform: [MyMacro, AWS::Serverless]
 Resources:
    WaitCondition:
      Type: AWS::CloudFormation::WaitCondition
    MyBucket:
      Type: AWS::S3::Bucket
      Properties:
        BucketName: amzn-s3-demo-bucket
        Tags: [{"key":"value"}]
        'Fn::Transform':
          - Name: PolicyAdder
        CorsConfiguration: []
    MyEc2Instance:
      Type: AWS::EC2::Instance
      Properties:
        ImageID: ami-1234567890abcdef0
```

Beispiel für ein einfaches Makro zum Ersetzen von Zeichenketten

Das folgende Beispiel führt Sie durch den Prozess der Verwendung von Makros, von der Definition des Makros in einer Vorlage über die Erstellung einer Lambda-Funktion für das Makro bis hin zur Verwendung des Makros in einer Vorlage.

In diesem Beispiel erstellen wir ein einfaches Makro, das die angegebene Zeichenfolge anstelle des angegebenen Zielinhalts in die verarbeitete Vorlage einfügt. Dann fügen wir damit ein leeres `WaitHandleCondition` an der angegebenen Stelle in der bearbeiteten Vorlage ein.

Ein Makro erstellen

Bevor wir ein Makro verwenden, müssen wir zunächst zwei Dinge erledigen: die Lambda-Funktion erstellen, die die gewünschte Vorlagenverarbeitung durchführt, und dann diese Lambda-Funktion verfügbar machen, CloudFormation indem wir eine Makrodefinition erstellen.

Die folgende Beispielvorlage enthält die Definition für unser Beispielmakro. Um das Makro in einem bestimmten Format verfügbar zu machen AWS-Konto, erstellen Sie einen Stack aus der Vorlage. Die Makrodefinition gibt den Makronamen und eine kurze Beschreibung an und verweist auf den ARN der Lambda-Funktion, die CloudFormation aufgerufen wird, wenn dieses Makro in einer Vorlage verwendet wird. (Wir haben keine `LogRoleARN` Eigenschaft `LogGroupName` oder für die Fehlerprotokollierung hinzugefügt.)

Gehen Sie in diesem Beispiel davon aus, dass der aus dieser Vorlage erstellte Stack benannt ist`JavaMacroFunc`. Da die `Name` Makroeigenschaft auf den Stacknamen gesetzt ist, wird `JavaMacroFunc` auch das resultierende Makro benannt.

```
AWSTemplateFormatVersion: 2010-09-09
  Resources:
    Macro:
      Type: AWS::CloudFormation::Macro
      Properties:
        Name: !Sub '${AWS::StackName}'
        Description: Adds a blank WaitConditionHandle named WaitHandle
        FunctionName: 'arn:aws:lambda:us-east-1:012345678910:function:JavaMacroFunc'
```

Verwenden des Makros

Um unser Makro zu verwenden, fügen wir es mithilfe der `Fn::Transform` systemeigenen Funktion in eine Vorlage ein.

CloudFormation Ruft unser Beispiel-Makro auf, wenn wir mit der folgenden Vorlage einen Stack erstellen. Die zugrundeliegende Lambda-Funktion ersetzt eine angegebene Zeichenfolge durch eine andere angegebene Zeichenfolge. In diesem Fall ergibt sich daraus eine leere `AWS::CloudFormation::WaitConditionHandle`-Eigenschaft, die in die verarbeitete Vorlage eingefügt wird.

```
Parameters:
  ExampleParameter:
    Type: String
    Default: 'SampleMacro'

Resources:
  2a:
    Fn::Transform:
      Name: "JavaMacroFunc"
      Parameters:
```

```
        replacement: 'AWS::CloudFormation::WaitConditionHandle'
        target: '$$REPLACEMENT$$'
    Type: '$$REPLACEMENT$$'
```

- Das aufzurufende Makro wird als angegeben`JavaMacroFunc`, was aus dem vorherigen Beispiel für eine Makrodefinition stammt.
- Dem Makro werden zwei Parameter, `target` und `replacement`, übergeben, die die Zielzeichenfolge und ihren gewünschten Ersatzwert darstellen.
- Das Makro kann auf den Inhalt des `Type`-Knotens zugreifen, da `Type` ein Geschwisterelement der `Fn::Transform`-Funktion ist, das sich auf das Makro bezieht.
- Die resultierende `AWS::CloudFormation::WaitConditionHandle`-Eigenschaft wird als `2a` benannt.
- Die Vorlage enthält auch einen Vorlagenparameter (`ExampleParameter`), auf den das Makro ebenfalls Zugriff hat (in diesem Fall aber nicht verwendet wird).

Lambda-Eingabedaten

Wenn unsere Beispielvorlage während der Stack-Erstellung CloudFormation verarbeitet wird, übergibt sie die folgende Event-Mapping an die Lambda-Funktion, auf die in der `JavaMacroFunc` Makrodefinition verwiesen wird.

- `region` : `us-east-1`
- `accountId` : `012345678910`
- `fragment` :

  ```
  {
    "Type": "$$REPLACEMENT$$"
  }
  ```

- `transformId` : `012345678910::JavaMacroFunc`
- `params` :

  ```
  {
      "replacement": "AWS::CloudFormation::WaitConditionHandle",
      "target": "$$REPLACEMENT$$"
  }
  ```

- `requestId` : `5dba79b5-f117-4de0-9ce4-d40363bfb6ab`

- `templateParameterValues` :

```
{
    "ExampleParameter": "SampleMacro"
}
```

`fragment` enthält JSON, das das Vorlagenfragment darstellt, das das Makro verarbeiten kann. Dieses Fragment besteht aus den Geschwistern des `Fn::Transform`-Funktionsaufrufs, nicht aber aus dem Funktionsaufruf selbst. Außerdem enthält `params` JSON, das die Makroparameter repräsentiert. In diesem Fall Ersatz und Ziel. Ebenso enthält `templateParameterValues` JSON, das die für die gesamte Vorlage angegebenen Parameter repräsentiert.

Lambda-Funktionscode

Im Folgenden finden Sie den eigentlichen Code für die Lambda-Funktion, die dem `JavaMacroFunc` Beispielmakro zugrunde liegt. Es durchläuft das in der Antwort enthaltene Vorlagenfragment (als Zeichenfolge, Liste oder Zuordnung) und sucht nach der angegebenen Zielzeichenfolge. Wenn die angegebene Zeichenfolge gefunden wird, ersetzt die Lambda-Funktion die Zielzeichenfolge durch die angegebene Ersatzzeichenfolge. Wenn nicht, lässt die Funktion das Vorlagenfragment unverändert. Anschließend gibt die Funktion eine Übersicht der erwarteten Eigenschaften zurück, auf die weiter unten näher eingegangen wird. CloudFormation

```
package com.macroexample.lambda.demo;

import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

public class LambdaFunctionHandler implements RequestHandler<Map<String, Object>,
 Map<String, Object>> {

 private static final String REPLACEMENT = "replacement";
 private static final String TARGET = "target";
 private static final String PARAMS = "params";
 private static final String FRAGMENT = "fragment";
 private static final String REQUESTID = "requestId";
```

```
private static final String STATUS = "status";
private static final String SUCCESS = "SUCCESS";
private static final String FAILURE = "FAILURE";
   @Override
   public Map<String, Object> handleRequest(Map<String, Object> event, Context
context) {
       // TODO: implement your handler
    final Map<String, Object> responseMap = new HashMap<String, Object>();
       responseMap.put(REQUESTID, event.get(REQUESTID));
       responseMap.put(STATUS, FAILURE);
    try {
        if (!event.containsKey(PARAMS)) {
         throw new RuntimeException("Params are required");
        }

        final Map<String, Object> params = (Map<String, Object>) event.get(PARAMS);
        if (!params.containsKey(REPLACEMENT) || !params.containsKey(TARGET)) {
         throw new RuntimeException("replacement or target under Params are
required");
        }

     final String replacement = (String) params.get(REPLACEMENT);
     final String target = (String) params.get(TARGET);
     final Object fragment = event.getOrDefault(FRAGMENT, new HashMap<String,
Object>());
     final Object retFragment;
     if (fragment instanceof String) {
      retFragment = iterateAndReplace(replacement, target, (String) fragment);
     } else if (fragment instanceof List) {
      retFragment = iterateAndReplace(replacement, target, (List<Object>) fragment);
     } else if (fragment instanceof Map) {
      retFragment = iterateAndReplace(replacement, target, (Map<String, Object>)
fragment);
     } else {
      retFragment = fragment;
     }
        responseMap.put(STATUS, SUCCESS);
        responseMap.put(FRAGMENT, retFragment);
        return responseMap;
    } catch (Exception e) {
     e.printStackTrace();
     context.getLogger().log(e.getMessage());
     return responseMap;
    }
```

```
    }

    private Map<String, Object> iterateAndReplace(final String replacement, final
String target, final Map<String, Object> fragment) {
     final Map<String, Object> retFragment = new HashMap<String, Object>();
     final List<String> replacementKeys = new ArrayList<>();
     fragment.forEach((k, v) -> {
      if (v instanceof String) {
       retFragment.put(k, iterateAndReplace(replacement, target, (String)v));
      } else if (v instanceof List) {
       retFragment.put(k, iterateAndReplace(replacement, target, (List<Object>)v));
      } else if (v instanceof Map ) {
       retFragment.put(k, iterateAndReplace(replacement, target, (Map<String, Object>)
v));
      } else {
       retFragment.put(k, v);
      }
     });
     return retFragment;
    }

    private List<Object> iterateAndReplace(final String replacement, final String
target, final List<Object> fragment) {
     final List<Object> retFragment = new ArrayList<>();
     fragment.forEach(o -> {
      if (o instanceof String) {
       retFragment.add(iterateAndReplace(replacement, target, (String) o));
      } else if (o instanceof List) {
       retFragment.add(iterateAndReplace(replacement, target, (List<Object>) o));
      } else if (o instanceof Map) {
       retFragment.add(iterateAndReplace(replacement, target, (Map<String, Object>)
o));
      } else {
       retFragment.add(o);
      }
     });
     return retFragment;
    }

    private String iterateAndReplace(final String replacement, final String target,
final String fragment) {
     System.out.println(replacement + " == " + target + " == " + fragment );
     if (fragment != null AND_AND fragment.equals(target))
      return replacement;
```

```
     return fragment;
    }
}
```

Antwort der Lambda-Funktion

Im Folgenden finden Sie das Mapping, zu dem die Lambda-Funktion CloudFormation zur Verarbeitung zurückkehrt.

- `requestId` : `5dba79b5-f117-4de0-9ce4-d40363bfb6ab`
- `status` : `SUCCESS`
- `fragment` :

```
{
  "Type": "AWS::CloudFormation::WaitConditionHandle"
}
```

Die `requestId` Übereinstimmungen, die von gesendet wurden CloudFormation, und ein `status` Wert von `SUCCESS` bedeuten, dass die Lambda-Funktion das in der Anfrage enthaltene Vorlagenfragment erfolgreich verarbeitet hat. In dieser Antwort enthält `fragment` JSON, das den Inhalt darstellt, der anstelle des ursprünglichen Vorlagenausschnitts in die verarbeitete Vorlage eingefügt werden soll.

Das Ergebnis: verarbeitete Vorlage

Nachdem sie CloudFormation eine erfolgreiche Antwort von der Lambda-Funktion erhalten hat, fügt sie das zurückgegebene Vorlagenfragment in die verarbeitete Vorlage ein.

Nachfolgend sehen Sie die resultierende verarbeitete Vorlage für unser Beispiel. Der `Fn::Transform` systeminterne Funktionsaufruf, der auf das `JavaMacroFunc` Makro verwiesen hat, ist nicht mehr enthalten. Das von der Lambda-Funktion zurückgegebene Vorlagenfragment wird an der entsprechenden Stelle eingefügt, sodass der Inhalt `"Type": "$$REPLACEMENT$$"` durch `"Type": "AWS::CloudFormation::WaitConditionHandle"` ersetzt wurde.

```
{
    "Parameters": {
        "ExampleParameter": {
            "Default": "SampleMacro",
            "Type": "String"
```

```
        }
    },
    "Resources": {
        "2a": {
            "Type": "AWS::CloudFormation::WaitConditionHandle"
        }
    }
}
```

Fehler bei der verarbeiteten Vorlage beheben

Wenn Sie ein Makro verwenden, finden Sie die verarbeitete Vorlage in der CloudFormation Konsole.

Die Phase einer Vorlage gibt ihren Verarbeitungsstatus an:

- `Original`: Die Vorlage, die der Benutzer ursprünglich zur Erstellung oder Aktualisierung des Stacks oder Stack-Sets übermittelt hat.
- `Processed`: Die Vorlage, die CloudFormation verwendet wird, um den Stack oder das Stack-Set nach der Verarbeitung aller referenzierten Makros zu erstellen oder zu aktualisieren. Die verarbeitete Vorlage wird als JSON formatiert, auch wenn die ursprüngliche Vorlage als YAML formatiert wurde.

Verwenden Sie zur Fehlerbehebung die verarbeitete Vorlage. Wenn eine Vorlage keine Makros referenziert, sind die ursprünglichen und verarbeiteten Vorlagen identisch.

Weitere Informationen finden Sie unter Stack-Informationen von der CloudFormation Konsole aus anzeigen.

Verwenden Sie den get-templateBefehl, AWS CLI um die verarbeitete Vorlage abzurufen.

Größenbeschränkung

Die maximale Größe für eine verarbeitete Stack-Vorlage beträgt 51 200 Byte, falls sie direkt an eine `CreateStack`-, `UpdateStack`- oder `ValidateTemplate`-Anforderung übergeben wird, oder 1 MB, wenn sie unter Verwendung einer Amazon-S3-Vorlagen-URL als S3-Objekt übergeben wird. Während der Verarbeitung wird jedoch der temporäre Status der Vorlage CloudFormation aktualisiert, da die in der Vorlage enthaltenen Makros seriell verarbeitet werden. Aus diesem Grund kann die Größe der Vorlage während der Verarbeitung vorübergehend die zulässige Größe einer vollständig verarbeiteten Vorlage überschreiten. CloudFormation ermöglicht einen gewissen Puffer für diese in

Bearbeitung befindlichen Vorlagen. Allerdings sollten Sie beim Entwurf Ihrer Vorlagen und Makros immer die maximal zulässige Größe für eine verarbeitete Stack-Vorlage im Blick behalten.

Wenn bei der Verarbeitung Ihrer Vorlage ein `Transformation data limit exceeded` Fehler CloudFormation zurückgegeben wird, hat Ihre Vorlage die maximale Vorlagengröße überschritten, die während der Verarbeitung CloudFormation zulässig ist.

Sie können das Problem u. U. wie folgt beheben:

- Teilen Sie Ihre Vorlage in mehrere Vorlagen auf, um zu vermeiden, dass die maximale Vorlagengröße während der Verarbeitung überschritten wird. Beispiel:
  - Verwenden Sie verschachtelte Stack-Vorlagen, um Teile der Vorlage einzukapseln. Weitere Informationen finden Sie unter Teilen Sie eine Vorlage mithilfe verschachtelter Stapel in wiederverwendbare Teile auf.
  - Erstellen Sie mehrere Stacks und verwenden Sie Stack-übergreifende Referenzen, damit Informationen zwischen den Stacks ausgetauscht werden können. Weitere Informationen finden Sie unter Verweisen Sie auf Ressourcenausgaben in einem anderen CloudFormation Stack.
- Reduzieren Sie die Größe des von einem bestimmten Makro zurückgegebenen Vorlagenfragments. CloudFormation manipuliert den Inhalt von Fragmenten, die von Makros zurückgegeben werden, nicht.

Teilen Sie eine Vorlage mithilfe verschachtelter Stapel in wiederverwendbare Teile auf

Wenn Ihre Infrastruktur wächst, müssen Sie möglicherweise wiederholt identische Ressourcenkonfigurationen für mehrere Vorlagen erstellen. Um diese Redundanz zu vermeiden, können Sie diese allgemeinen Konfigurationen in spezielle Vorlagen unterteilen. Anschließend können Sie die AWS::CloudFormation::StackRessource in anderen Vorlagen verwenden, um auf diese speziellen Vorlagen zu verweisen und verschachtelte Stapel zu erstellen.

Nehmen wir beispielsweise an, Sie haben eine Load Balancer-Konfiguration, die Sie für die meisten Ihrer Stacks verwenden. Anstatt dieselbe Konfigurationen zu kopieren und in Ihre Vorlagen einzufügen, können Sie eine dedizierte Vorlage für den Load Balancer erstellen. Dann können Sie in anderen Vorlagen, die dieselbe Load Balancer-Konfiguration erfordern, auf diese Vorlage verweisen.

Verschachtelte Stapel können selbst andere verschachtelte Stapel enthalten, was zu einer Hierarchie von Stapeln führt, wie in der folgenden Abbildung dargestellt. Der Root-Stack ist der Stack der obersten Ebene, zu dem letztendlich alle verschachtelten Stacks gehören. Jeder verschachtelte

Stapel hat einen unmittelbaren übergeordneten Stapel. Bei der ersten Ebene verschachtelter Stacks ist der Root-Stack auch der übergeordnete Stack.

- Stack A ist der Root-Stack für alle anderen – verschachtelten – Stacks in der Hierarchie.
- Für Stack B ist Stack A sowohl der übergeordnete Stack als auch der Root-Stack.
- Für Stack D ist Stack C der übergeordnete Stack, während Stack B der übergeordnete Stack für Stack C ist.

Themen

- Beispiel für das Teilen einer Vorlage vor und nach dem Aufteilen
- Beispiel für eine verschachtelte Stack-Architektur
- Ausführen von Stack-Operationen auf verschachtelten Stacks
- Ähnliche Informationen

Beispiel für das Teilen einer Vorlage vor und nach dem Aufteilen

Dieses Beispiel zeigt, wie Sie eine einzelne, große CloudFormation Vorlage mithilfe verschachtelter Vorlagen in ein strukturierteres und wiederverwendbares Design umorganisieren können. Anfänglich zeigt die Vorlage „Before Nesting Stacks“ alle in einer Datei definierten Ressourcen. Dies kann unübersichtlich und schwer zu verwalten werden, wenn die Anzahl der Ressourcen zunimmt.
Die Vorlage „After Nesting Stacks“ teilt die Ressourcen in kleinere, separate Vorlagen auf. Jeder verschachtelte Stapel verwaltet einen bestimmten Satz verwandter Ressourcen, wodurch die Gesamtstruktur übersichtlicher und einfacher zu verwalten ist.

Vor dem Verschachteln von Stapeln	Nach dem Verschachteln von Stapeln

`AWSTemplateFormatVersion: 2010-09-09`
`Parameters:`
`  InstanceType:`
`  Type: String`
`  Default: t2.micro`
`  Description: The EC2 instance type`
 	
`  Environment:`
`  Type: String`
	`AWSTemplateFormatVersion: 2010-09-09`
	`Resources:`
	`  MyFirstNestedStack:`
	`  Type: AWS::CloudFormation::Stack`
	`  Properties:`
	`  TemplateURL: https://s3.amazona`
	`ws.com/ amzn-s3-demo-bucket  /first-ne`
	`sted-stack.yaml`
	`  Parameters:`

Vor dem Verschachteln von Stapeln	Nach dem Verschachteln von Stapeln

`Default: Production`
`Description: The deployment environment`
 	
`Resources:`
`MyEC2Instance:`
`Type: AWS::EC2::Instance`
`Properties:`
`ImageId: ami-1234567890abcdef0`
`InstanceType: !Ref InstanceType`
 	
`MyS3Bucket:`
`Type: AWS::S3::Bucket`
	`# Pass parameters to the nested stack if needed`
	`InstanceType: t3.micro`
	
	`MySecondNestedStack:`
	`Type: AWS::CloudFormation::Stack`
	`Properties:`
	`TemplateURL: https://s3.amazonaws.com/ amzn-s3-demo-bucket /second-nested-stack.yaml`
	`Parameters:`
	`# Pass parameters to the nested stack if needed`
	`Environment: Testing`
	`DependsOn: MyFirstNestedStack`

Beispiel für eine verschachtelte Stack-Architektur

In diesem Abschnitt wird eine verschachtelte Stack-Architektur demonstriert, die aus einem Stack der obersten Ebene besteht, der auf einen verschachtelten Stack verweist. Der verschachtelte Stack stellt eine Lambda-Funktion von Node.js bereit, empfängt einen Parameterwert vom Stack der obersten Ebene und gibt eine Ausgabe zurück, die über den Stack der obersten Ebene verfügbar gemacht wird.

Themen

- Schritt 1: Erstellen Sie eine Vorlage für den verschachtelten Stack auf Ihrem lokalen System
- Schritt 2: Erstellen Sie eine Vorlage für den Top-Level-Stack auf Ihrem lokalen System
- Schritt 3: Verpacken und Bereitstellen der Vorlagen

Schritt 1: Erstellen Sie eine Vorlage für den verschachtelten Stack auf Ihrem lokalen System

Das folgende Beispiel zeigt das Format der Vorlage für verschachtelte Stacks.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Nested stack template for Lambda function deployment
```

```yaml
Parameters:
  MemorySize:
    Type: Number
    Default: 128
    MinValue: 128
    MaxValue: 10240
    Description: Lambda function memory allocation (128-10240 MB)
Resources:
  LambdaFunction:
    Type: AWS::Lambda::Function
    Properties:
      FunctionName: !Sub "${AWS::StackName}-Function"
      Runtime: nodejs18.x
      Handler: index.handler
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile: |
          exports.handler = async (event) => {
            return {
              statusCode: 200,
              body: JSON.stringify('Hello from Lambda!')
            };
          };
      MemorySize: !Ref MemorySize
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
            Action: sts:AssumeRole
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole'
Outputs:
  LambdaArn:
    Description: ARN of the created Lambda function
    Value: !GetAtt LambdaFunction.Arn
```

Schritt 2: Erstellen Sie eine Vorlage für den Top-Level-Stack auf Ihrem lokalen System

Das folgende Beispiel zeigt das Format der Vorlage für den obersten Stack und die AWS::CloudFormation::StackRessource, die auf den Stack verweist, den Sie im vorherigen Schritt erstellt haben.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Top-level stack template that deploys a nested stack
Resources:
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: /path_to_template/nested-template.yaml
      Parameters:
        MemorySize: 256
Outputs:
  NestedStackLambdaArn:
    Description: ARN of the Lambda function from nested stack
    Value: !GetAtt NestedStack.Outputs.LambdaArn
```

Schritt 3: Verpacken und Bereitstellen der Vorlagen

> **Note**
>
> Wenn Sie lokal mit Vorlagen arbeiten, kann Ihnen der AWS CLI package Befehl dabei helfen, Vorlagen für die Bereitstellung vorzubereiten. Es verarbeitet automatisch den Upload lokaler Artefakte auf Amazon S3 (einschließlich`TemplateURL`) und generiert eine neue Vorlagendatei mit aktualisierten Verweisen auf diese S3-Speicherorte. Weitere Informationen finden Sie unter Laden Sie lokale Artefakte in einen S3-Bucket hoch mit dem AWS CLI.

Als Nächstes können Sie den packageBefehl verwenden, um die verschachtelte Vorlage in einen Amazon S3 S3-Bucket hochzuladen.

```
aws cloudformation package \
  --s3-bucket amzn-s3-demo-bucket \
  --template /path_to_template/top-level-template.yaml \
  --output-template-file packaged-template.yaml \
  --output json
```

Der Befehl generiert eine neue Vorlage in dem von `--output-template-file` angegebenen Pfad. Es ersetzt die `TemplateURL` Referenz durch den Amazon S3 S3-Standort, wie unten gezeigt.

Resultierende Vorlage

```
AWSTemplateFormatVersion: 2010-09-09
Description: Top-level stack template that deploys a nested stack
Resources:
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: https://s3.us-west-2.amazonaws.com/amzn-s3-demo-
bucket/8b3bb7aa7abfc6e37e2d06b869484bed.template
      Parameters:
        MemorySize: 256
Outputs:
  NestedStackLambdaArn:
    Description: ARN of the Lambda function from nested stack
    Value:
      Fn::GetAtt:
      - NestedStack
      - Outputs.LambdaArn
```

Nachdem Sie den package Befehl ausgeführt haben, können Sie die verarbeitete Vorlage mithilfe des deployBefehls bereitstellen. Bei verschachtelten Stacks, die IAM-Ressourcen enthalten, müssen Sie die IAM-Funktionen bestätigen, indem Sie die Option angeben. `--capabilities`

```
aws cloudformation deploy \
  --template-file packaged-template.yaml \
  --stack-name stack-name \
  --capabilities CAPABILITY_NAMED_IAM
```

Ausführen von Stack-Operationen auf verschachtelten Stacks

Wenn Sie mit verschachtelten Stacks arbeiten, müssen Sie diese während des Betriebs sorgfältig behandeln. Bestimmte Stack-Operationen, wie z. B. Stack-Updates, sollten vom Root-Stack aus initiiert und nicht direkt auf den verschachtelten Stacks ausgeführt werden. Wenn Sie einen Root-Stack aktualisieren, werden nur verschachtelte Stacks mit Vorlagenänderungen aktualisiert.

Darüber hinaus kann das Vorhandensein der verschachtelten Stacks die Operationen auf dem Root-Stack beeinflussen. Wenn beispielsweise ein verschachtelter Stack im

`UPDATE_ROLLBACK_IN_PROGRESS` Status stecken bleibt, wartet der Root-Stack, bis dieser verschachtelte Stack sein Rollback abgeschlossen hat, bevor er fortfährt. Bevor Sie mit den Aktualisierungsvorgängen fortfahren, stellen Sie sicher, dass Sie über IAM-Berechtigungen verfügen, um ein Stack-Update abzubrechen, falls es zurückgesetzt wird. Weitere Informationen finden Sie unter Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management.

Gehen Sie wie folgt vor, um den Root-Stack und die verschachtelten Stacks zu finden.

So zeigen Sie den Root-Stack eines verschachtelten Stacks an

1. Melden Sie sich bei /cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole. https://console.aws.amazon.com
2. Wählen Sie auf der Seite Stacks den Namen des verschachtelten Stacks aus, dessen Root-Stack Sie anzeigen möchten.

   Bei verschachtelten Stacks wird NESTED über ihrem Stacknamen angezeigt.
3. Wählen Sie auf der Registerkarte Stack-Info im Abschnitt Übersicht den Stack-Namen aus, der als Root-Stack aufgeführt ist.

So zeigen Sie die verschachtelten Stacks eines Root-Stacks an

1. Wählen Sie aus dem Root-Stack, dessen verschachtelte Stapel Sie anzeigen möchten, den Tab Ressourcen aus.
2. Suchen Sie in der Spalte Typ nach Ressourcen des Typs. AWS::CloudFormation::Stack

Ähnliche Informationen

- Verschachteln eines vorhandenen Stacks
- Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen
- Setzt das Rollback nach fehlgeschlagenen Aktualisierungen des verschachtelten Stacks fort
- Rollback-Fehler bei verschachtelten Stacks

Wartebedingungen in einer CloudFormation Vorlage erstellen

In diesem Thema wird erklärt, wie Sie eine Wartebedingung in einer Vorlage erstellen, um die Erstellung von Stack-Ressourcen zu koordinieren oder den Fortschritt eines Konfigurationsprozesses

zu verfolgen. Sie können z. B. die Erstellung einer anderen Ressource beginnen, nachdem eine Anwendungskonfiguration teilweise abgeschlossen ist, oder während eines Installations- und Konfigurationsvorgangs Signale senden, um dessen Fortschritt zu überwachen.

Wenn ein Stack CloudFormation erstellt wird, der eine Wartebedingung enthält:

- Es erstellt wie jede andere Ressource eine Wartebedingung und setzt den Status der Wartebedingung auf`CREATE_IN_PROGRESS`.
- CloudFormation wartet, bis die erforderliche Anzahl von Erfolgssignalen empfangen wurde oder die Zeitüberschreitung der Wartebedingung abgelaufen ist.
- Wenn es die erforderliche Anzahl von Erfolgssignalen empfängt, bevor der Timeout-Zeitraum abläuft:
  - Der Status des Wartezustands ändert sich zu `CREATE_COMPLETE`
  - Die Stack-Erstellung wird fortgesetzt
- Wenn das Timeout abläuft oder ein Fehlersignal empfangen wird:
  - Der Status des Wartezustands ändert sich zu `CREATE_FAILED`
  - Der Stapel wird zurückgesetzt

> **Important**
>
> Für Amazon EC2 - und Auto Scaling Scaling-Ressourcen empfehlen wir, anstelle von Wartebedingungen ein CreationPolicy Attribut zu verwenden. Fügen Sie diesen Ressourcen ein CreationPolicy Attribut hinzu und verwenden Sie das cfn-signal-Hilfsskript, um zu signalisieren, dass ein Instance-Erstellungsprozess erfolgreich abgeschlossen wurde. Weitere Informationen finden Sie unter CreationPolicy-Attribut.

> **Note**
>
> Wenn Sie diese Option verwenden AWS PrivateLink, müssen Ressourcen in der VPC, die auf Wartebedingungen reagieren, Zugriff auf CloudFormation spezifische Amazon Simple Storage Service (Amazon S3) -Buckets haben. Ressourcen müssen Wartebedingungs-Antworten an eine vorsignierte Amazon-S3-URL senden. Wenn sie keine Antworten an Amazon S3 senden können, empfängt CloudFormation keine Antwort und der Stack-Vorgang schlägt fehl. Weitere Informationen finden Sie unter Zugriff CloudFormation über einen

Schnittstellenendpunkt (AWS PrivateLink) und Steuern des Zugriffs von VPC-Endpunkten mit Bucket-Richtlinien.

Themen

- Eine Wartebedingung in Ihrer Vorlage erstellen
- Syntax des Signals für Wartebedingungen
- Zugreifen auf Signaldaten

Eine Wartebedingung in Ihrer Vorlage erstellen

1. Handle für Wartebedingungen

Sie beginnen damit, eine AWS::CloudFormation::WaitConditionHandleRessource in der Vorlage des Stacks zu definieren. Diese Ressource generiert die vorsignierte URL, die zum Senden von Signalen benötigt wird. Auf diese Weise können Sie ein Signal senden, ohne Ihre AWS Anmeldeinformationen angeben zu müssen. Zum Beispiel:

```
Resources:
  MyWaitHandle:
    Type: AWS::CloudFormation::WaitConditionHandle
```

2. Wartebedingung

Als Nächstes definieren Sie eine AWS::CloudFormation::WaitConditionRessource in der Vorlage des Stacks. Die Grundstruktur von a `AWS::CloudFormation::WaitCondition` sieht wie folgt aus:

```
  MyWaitCondition:
    Type: AWS::CloudFormation::WaitCondition
    Properties:
      Handle: String
      Timeout: String
      Count: Integer
```

Die `AWS::CloudFormation::WaitCondition` Ressource hat zwei erforderliche Eigenschaften und eine optionale Eigenschaft.

- `Handle`(erforderlich) — Ein Verweis auf eine in der Vorlage `WaitConditionHandle` deklarierte.

- `Timeout`(erforderlich) — Die Anzahl der Sekunden, die benötigt werden CloudFormation, um auf den Empfang der erforderlichen Anzahl von Signalen zu warten. `Timeout`ist eine Eigenschaft, die an das Minimum gebunden ist. Das bedeutet, dass der Timeout frühestens zu dem von Ihnen angegebenen Zeitpunkt eintritt, aber auch kurz danach eintreten kann. Die maximale Zeit, die Sie angeben können, ist 43200 Sekunden (12 Stunden).
- `Count`(optional) — Die Anzahl der Erfolgssignale, die empfangen CloudFormation werden müssen, bevor der Status dieser Wartebedingung auf gesetzt `CREATE_COMPLETE` und die Erstellung des Stacks fortgesetzt wird. Wenn nicht angegeben, ist der Standardwert 1.

In der Regel möchten Sie, dass eine Wartebedingung unmittelbar nach der Erstellung einer bestimmten Ressource beginnt. Dazu fügen Sie das `DependsOn` Attribut einer Wartebedingung hinzu. Wenn Sie einer Wartebedingung ein `DependsOn` Attribut hinzufügen, CloudFormation erstellt zuerst die Ressource im `DependsOn` Attribut und dann die Wartebedingung. Weitere Informationen finden Sie unter DependsOn-Attribut.

Das folgende Beispiel zeigt eine Wartebedingung, die:

- Beginnt nach der erfolgreichen Erstellung der `MyEC2Instance` Ressource
- Verwendet die `MyWaitHandle` Ressource als `WaitConditionHandle`
- Hat ein Timeout von 4500 Sekunden
- Hat den Standardwert `Count` 1 (da keine `Count` Eigenschaft angegeben ist)

```
MyWaitCondition:
  Type: AWS::CloudFormation::WaitCondition
  DependsOn: MyEC2Instance
  Properties:
    Handle: !Ref MyWaitHandle
    Timeout: '4500'
```

3. Ein Signal senden

Um Erfolg oder Misserfolg zu signalisieren CloudFormation, führen Sie normalerweise Code oder ein Skript aus. Beispielsweise kann eine Anwendung, die auf einer EC2 Instanz ausgeführt wird, einige zusätzliche Konfigurationsaufgaben ausführen und dann ein Signal an senden, CloudFormation um den Abschluss anzuzeigen.

Das Signal muss an die vorsignierte URL gesendet werden, die durch das Wartebedingungs-Handle generiert wurde. Sie verwenden diese vorsignierte URL, um Erfolg oder Misserfolg zu signalisieren.

Um ein Signal zu senden

1. Um die vorsignierte URL innerhalb der Vorlage abzurufen, verwenden Sie die `Ref` systeminterne Funktion mit dem logischen Namen des Wartebedingungs-Handles.

   Wie im folgenden Beispiel gezeigt, kann Ihre Vorlage eine EC2 Amazon-Instance deklarieren und die vorsignierte URL mithilfe der EC2 `UserData` Amazon-Eigenschaft an EC2 Instances übergeben. Auf diese Weise können Skripte oder Anwendungen, die auf diesen Instances ausgeführt werden, den Erfolg oder Misserfolg signalisieren. CloudFormation

   ```
    MyEC2Instance:
      Type: AWS::EC2::Instance
      Properties:
      InstanceType: t2.micro  # Example instance type
      ImageId: ami-055e3d4f0bbeb5878  # Change this as needed (Amazon Linux 2023 in
   us-west-2)
      UserData:
        Fn::Base64:
          Fn::Join:
            - ""
            - - "SignalURL="
              - { "Ref": "MyWaitHandle" }
   ```

   Dies führt zu einer `UserData` Ausgabe, die der folgenden ähnelt:

   ```
   SignalURL=https://amzn-s3-demo-bucket.s3.amazonaws.com/....
   ```

   Hinweis: In den Befehlszeilentools AWS Management Console und den Befehlszeilentools wird die vorsignierte URL als physische ID der Ressource „Wait Condition Handle“ angezeigt.

2. (Optional) Um zu erkennen, wann der Stack in die Wartebedingung übergeht, können Sie eine der folgenden Methoden verwenden:

   - Wenn Sie den Stack mit aktivierten Benachrichtigungen erstellen, veröffentlicht CloudFormation eine Benachrichtigung für jedes Stack-Ereignis für das angegebene Thema. Wenn Sie oder Ihre Anwendung dieses Thema abonnieren, können Sie die

Benachrichtigungen für das Wartebedingungs-Handle-Erstellungsereignis überwachen und die vorsignierte URL aus der Benachrichtigungsnachricht abrufen.

- Sie können die Ereignisse des Stacks auch mit dem AWS Management Console AWS CLI, dem oder einem SDK überwachen.

3. Um ein Signal zu senden, senden Sie eine HTTP-Anforderungsnachricht mithilfe der vorsignierten URL. Die Anforderungsmethode muss `PUT` und der Header `Content-Type` muss eine leere Zeichenfolge oder nicht angegeben sein. Die Anforderungsnachricht muss eine JSON-Struktur der Form aufweisen, die in Syntax des Signals für Wartebedingungen angegeben ist.

   Sie müssen die in der `Count` Eigenschaft angegebene Anzahl von Erfolgssignalen senden, um mit der Stack-Erstellung fortzufahren. CloudFormation Wenn `Count` größer als 1 ist, muss der Wert `UniqueId` für alle Signale, die an eine bestimmte Wartebedingung gesendet werden, jeweils eindeutig sein. Die `UniqueId` ist eine beliebige alphanumerische Zeichenfolge.

   Ein `curl`-Befehl ist eine Möglichkeit zum Senden eines Signals. Das folgende Beispiel zeigt eine `curl`-Befehlszeile, die einer Wartebedingung den Erfolg signalisiert.

```
$ curl -T /tmp/a \
  "https://amzn-s3-demo-bucket.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-
west-2%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-
WaitCondition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wyOoRejVo
%3D"
```

   wobei die Datei die folgende JSON-Struktur `/tmp/a` enthält:

```
{
   "Status" : "SUCCESS",
   "Reason" : "Configuration Complete",
   "UniqueId" : "ID1234",
   "Data" : "Application has completed configuration."
}
```

   Das folgende Beispiel zeigt eine `curl`-Befehlszeile, die dasselbe Erfolgssignal sendet, wobei aber die JSON-Struktur als Parameter in der Befehlszeile gesendet wird.

```
$ curl -X PUT \
```

```
  -H 'Content-Type:' --data-binary '{"Status" : "SUCCESS","Reason" : "Configuration
 Complete","UniqueId" : "ID1234","Data" : "Application has completed
 configuration."}' \
  "https://amzn-s3-demo-bucket.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-
west-2%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-
WaitCondition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wyOoRejVo
%3D"
```

Syntax des Signals für Wartebedingungen

Wenn Sie Signale an die URL senden, die durch das Wartebedingungs-Handle generiert wurde, müssen Sie das folgende JSON-Format verwenden:

```
{
  "Status" : "StatusValue",
  "UniqueId" : "Some UniqueId",
  "Data" : "Some Data",
  "Reason" : "Some Reason"
}
```

Eigenschaften

Das `Status` Feld muss einen der folgenden Werte haben:

- `SUCCESS`
- `FAILURE`

Das `UniqueId` Feld identifiziert das Signal an CloudFormation. Wenn die `Count` Eigenschaft der Wartebedingung größer als 1 ist, muss der `UniqueId` Wert für alle Signale, die für eine bestimmte Wartebedingung gesendet wurden, eindeutig sein. Andernfalls CloudFormation wird das Signal als erneute Übertragung des zuvor gesendeten Signals mit derselben betrachtet `UniqueId` und ignoriert.

Das `Data` Feld kann alle Informationen enthalten, die Sie zusammen mit dem Signal zurücksenden möchten. Sie können auf den `Data` Wert zugreifen, indem Sie die GetAtt Funktion Fn:: innerhalb der Vorlage verwenden.

Das `Reason` Feld ist eine Zeichenfolge, deren Inhalt außer der JSON-Konformität keine weiteren Einschränkungen aufweist.

Zugreifen auf Signaldaten

Um auf die von gültigen Signalen gesendeten Daten zuzugreifen, können Sie in Ihrer CloudFormation Vorlage einen Ausgabewert für die Wartebedingung erstellen. Zum Beispiel:

```
Outputs:
  WaitConditionData:
    Description: The data passed back as part of signalling the WaitCondition
    Value: !GetAtt MyWaitCondition.Data
```

Sie können diese Daten dann mit dem describe-stacksBefehl oder auf der Registerkarte Ausgaben der CloudFormation Konsole anzeigen.

Die `Fn::GetAtt` Funktion gibt das `UniqueId` und `Data` als name/value Paar innerhalb einer JSON-Struktur zurück. Zum Beispiel:

```
{"Signal1":"Application has completed configuration."}
```

Erstellen Sie wiederverwendbare Ressourcenkonfigurationen, die in allen Vorlagen mit CloudFormation Modulen enthalten sein können

Module sind eine Möglichkeit für Sie, Ressourcenkonfigurationen für die Aufnahme in Stack-Vorlagen auf transparente, verwaltbare und wiederholbare Weise zu erstellen. Module können gängige Servicekonfigurationen und Best Practices als modulare, anpassbare Bausteine zusammenfassen, die Sie in Ihre Stack-Vorlagen aufnehmen können. Module ermöglichen es Ihnen, Ressourcenkonfigurationen, die Best Practices, Fachwissen und akzeptierte Richtlinien (für Bereiche wie Sicherheit, Compliance, Governance und Branchenvorschriften) in Ihre Vorlagen einbeziehen, ohne fundierte Kenntnisse über die Feinheiten der Ressourcenimplementierung erwerben zu müssen.

Ein Netzwerkexperte könnte beispielsweise ein Modul erstellen, das integrierte Sicherheitsgruppen und ingress/egress Regeln enthält, die den Sicherheitsrichtlinien entsprechen. Sie könnten dieses Modul dann in Ihre Vorlage aufnehmen, um eine sichere Netzwerkinfrastruktur in Ihrem Stack bereitzustellen, ohne Zeit damit verbringen zu müssen, herauszufinden VPCs, wie Subnetze, Sicherheitsgruppen und Gateways funktionieren. Und da Module versioniert sind, kann der

Modulautor eine neue Version des Moduls erstellen, wenn sich die Sicherheitsrichtlinien im Laufe der Zeit ändern, die diese Änderungen enthält.

Zu den Merkmalen der Verwendung von Modulen in Ihren Vorlagen gehören:

- Vorhersehbarkeit — Ein Modul muss sich an das Schema halten, das es in der CloudFormation Registrierung registriert, sodass Sie wissen, in welche Ressourcen es aufgelöst werden kann, wenn Sie es in Ihre Vorlage aufnehmen.
- Wiederverwendbarkeit — Sie können dasselbe Modul für mehrere Vorlagen und Konten verwenden.
- Rückverfolgbarkeit — CloudFormation Behält das Wissen darüber bei, welche Ressourcen in einem Stack über ein Modul bereitgestellt wurden, sodass Sie die Quelle der Ressourcenänderungen leicht nachvollziehen können.
- Verwaltbarkeit — Sobald Sie ein Modul registriert haben, können Sie es über die CloudFormation Registrierung verwalten, einschließlich Versionierung, Konto- und regionaler Verfügbarkeit.

Ein Modul kann enthalten:

- Eine oder mehrere Ressourcen, die aus dem Modul bereitgestellt werden sollen, zusammen mit allen zugehörigen Daten wie Ausgaben oder Bedingungen.
- Alle Modulparameter, mit denen Sie bei jeder Verwendung des Moduls benutzerdefinierte Werte angeben können.

Informationen zur Entwicklung von Modulen finden Sie unter Entwickeln von Modulen im CloudFormation CLI-Benutzerhandbuch.

Themen

- Überlegungen zur Verwendung von Modulen
- Verstehen Sie die Versionierung von Modulen
- Verwenden Sie Module aus der CloudFormation privaten Registry
- Verwenden Sie Parameter, um Modulwerte anzugeben
- Referenzmodulressourcen in CloudFormation Vorlagen

Überlegungen zur Verwendung von Modulen

- Für die Nutzung von Modulen fallen keine zusätzlichen Kosten an. Sie zahlen nur für die Ressourcen, die diese Module in Ihren Stacks auflösen.
- CloudFormation Kontingente, wie z. B. die maximal zulässige Anzahl von Ressourcen in einem Stapel oder die maximale Größe des Vorlagentexts, gelten für die verarbeitete Vorlage, unabhängig davon, ob die in dieser Vorlage enthaltenen Ressourcen aus Modulen stammen oder nicht. Weitere Informationen finden Sie unter CloudFormation Kontingente verstehen.
- Tags, die Sie auf Stack-Ebene angeben, werden den einzelnen Ressourcen zugewiesen, die vom Modul abgeleitet werden.
- Auf Modulebene angegebene Hilfsskripten werden bei der Verarbeitung der Vorlage nicht auf die einzelnen Ressourcen übertragen, die im Modul enthalten sind. CloudFormation
- Die im Modul angegebenen Ausgaben werden an Ausgaben auf Vorlagenebene weitergegeben.

  Jeder Ausgabe wird eine logische ID zugewiesen, die eine Verkettung des logischen Namens des Moduls und des Ausgabenamens ist, wie im Modul definiert. Weitere Informationen finden Sie unter Holen Sie sich exportierte Ausgaben aus einem bereitgestellten CloudFormation Stack.
- Die im Modul angegebenen Parameter werden nicht an Parameter auf Vorlagenebene weitergegeben.

  Sie können jedoch Parameter auf Vorlagenebene erstellen, die auf Parameter auf Modulebene verweisen. Weitere Informationen finden Sie unter Verwenden Sie Parameter, um Modulwerte anzugeben.

Verstehen Sie die Versionierung von Modulen

Die CloudFormation Registrierung dient als Repository, in dem Sie Module für die Verwendung in Ihrer Region registrieren AWS-Konto und verwalten können. Sie können Module aus verschiedenen Quellen, einschließlich Drittanbietern AWS, und Ihre eigenen benutzerdefinierten Erweiterungen innerhalb Ihres Kontos und Ihrer Region registrieren. Weitere Informationen finden Sie unter Erweiterungen mit der CloudFormation Registry verwalten.

Module können unterschiedliche Versionen haben, sodass Sie angeben können, welche Version eines Moduls Sie verwenden möchten. Diese Versionierungsfunktion ist besonders nützlich, wenn Sie ein Modul aktualisieren oder ändern müssen, ohne bestehende Stacks zu beschädigen, die davon abhängen.

Beachten Sie die folgenden Überlegungen, wenn Sie mehrere Versionen eines Moduls verwenden:

- CloudFormation Verwendet bei Stack-Vorgängen die Version des Moduls, die derzeit als Standardversion in der Region AWS-Konto und registriert ist, in der der Stack-Vorgang ausgeführt wird. Dies schließt Module ein, die in anderen Modulen verschachtelt sind.

  Beachten Sie daher, dass die Verwendung derselben Vorlage zu unterschiedlichen Ergebnissen führen kann, wenn Sie unterschiedliche Versionen desselben Moduls als Standardversion in verschiedenen Konten oder Regionen registriert haben.
- CloudFormation Verwendet bei Stack-Vorgängen die Version der Ressource, die derzeit als Standardversion in der Region AWS-Konto und registriert ist, in der der Stack-Vorgang ausgeführt wird. Dies schließt die Ressourcen ein, die durch das Einschließen von Modulen generiert wurden.
- Durch das Ändern der Standardversion eines Moduls wird kein Stack-Aktualisierungsvorgang ausgelöst. Wenn Sie jedoch das nächste Mal eine Stack-Operation mit einer beliebigen Vorlage ausführen, die dieses Modul enthält, z. B. ein Stack-Update, CloudFormation wird die neue Standardversion für den Vorgang verwendet.

  Die einzige Ausnahme besteht darin, ein Stack-Update mit der Option Vorherige Vorlagenoption verwenden durchzuführen, wie unten beschrieben.
- Wenn Sie bei Stack-Aktualisierungsvorgängen die Option „Vorherige Vorlage verwenden“ angeben, wird die zuvor verarbeitete Vorlage für das Stack-Update CloudFormation verwendet und das Modul nicht erneut verarbeitet, wenn Sie Änderungen daran vorgenommen haben.
- Um einheitliche Ergebnisse zu gewährleisten, sollten Sie, wenn Sie Module in eine Stack-Vorlage zur Verwendung mit Stack-Sets aufnehmen, sicherstellen, dass dieselbe Version des Moduls in allen Konten und Regionen, in denen Sie Ihre Stack-Instances bereitstellen möchten, als Standardversion festgelegt ist. Dies gilt auch für Module, die in anderen Modulen verschachtelt sind. Weitere Informationen finden Sie unter Verwaltung von Stacks über Konten und Regionen hinweg mit StackSets.

Anforderungen für die Aktivierung öffentlicher Module von Drittanbietern

Um ein öffentliches Modul eines Drittanbieters in Ihrem Konto und Ihrer Region erfolgreich zu aktivieren, muss Folgendes für jede öffentliche Erweiterung (Ressource oder Modul) eines Drittanbieters gelten, die im Modul enthalten ist:

- Aktivierung der Erweiterung — Die Erweiterung muss in dem Konto und der Region aktiviert sein, in der Sie sie verwenden möchten. Weitere Informationen finden Sie unter Verwenden Sie öffentliche Erweiterungen von Drittanbietern aus der CloudFormation Registrierung.
- Alias-Registrierung — Wenn die Erweiterung im Modul einen Typnamen-Alias verwendet, muss die Erweiterung in Ihrem Konto und Ihrer Region unter Verwendung desselben Typnamens als Alias registriert sein. Weitere Informationen finden Sie unter Verwenden Sie Aliase, um auf Erweiterungen zu verweisen.
- Versionskompatibilität — Bei der aktuell aktivierten Erweiterungsversion muss es sich um eine der unterstützten Hauptversionen der im Modul angegebenen Erweiterung handeln.

Wenn Sie nicht die richtigen öffentlichen Erweiterungen und Erweiterungsversionen von Drittanbietern aktiviert haben, schlägt der Vorgang fehl und es CloudFormation wird eine Fehlermeldung angezeigt, in der die Erweiterungen und Versionen aufgeführt werden, die aktiviert werden müssen, bevor das Modul erfolgreich aktiviert werden kann.

Verwenden Sie Module aus der CloudFormation privaten Registry

In diesem Thema wird erklärt, wie Module in CloudFormation Vorlagen verwendet werden. Stellen Sie sich Module als vorgefertigte Bündel von Ressourcen vor, die Sie zu Ihren Vorlagen hinzufügen können.

Um ein Modul zu verwenden, gehen Sie wie folgt vor:

- Modul registrieren — Sie registrieren Module in der CloudFormation Registrierung als private Erweiterungen. Stellen Sie sicher, dass es in der Region AWS-Konto und der Region registriert ist, in der Sie arbeiten. Weitere Informationen finden Sie unter CloudFormation Konzepte der Registrierung.
- Fügen Sie es in Ihre Vorlage ein — Fügen Sie das Modul dem Resources Abschnitt Ihrer CloudFormation Vorlage hinzu, genau wie Sie es mit anderen Ressourcen tun würden. Sie müssen auch alle erforderlichen Eigenschaften für das Modul angeben.
- Den Stack erstellen oder aktualisieren — Wenn Sie einen Stack-Vorgang starten, CloudFormation wird eine verarbeitete Vorlage generiert, die alle enthaltenen Module in die entsprechenden Ressourcen auflöst.
- Änderungen in der Vorschau anzeigen — Bevor Sie Änderungen vornehmen, können Sie anhand eines Änderungssatzes sehen, welche Ressourcen hinzugefügt oder geändert werden. Weitere Informationen finden Sie unter CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren.

Betrachten Sie das folgende Beispiel: Sie haben eine Vorlage, die sowohl Ressourcen als auch Module enthält. Die Vorlage enthält eine einzelne Ressource sowie ein Modul`ModuleParent`. `ResourceA` Dieses Modul enthält zwei Ressourcen sowie ein verschachteltes Modul. `ResourceB` `ResourceC` `ModuleChild` `ModuleChild`enthält eine einzelne Ressource,`ResourceD`. Wenn Sie aus dieser Vorlage einen Stapel erstellen, CloudFormation verarbeitet die Vorlage und löst die Module in die entsprechenden Ressourcen auf. Der resultierende Stapel hat vier Ressourcen:`ResourceA`, `ResourceB``ResourceC`, und`ResourceD`.

CloudFormation verfolgt, welche Ressourcen in einem Stapel aus Modulen erstellt wurden. Sie können diese Informationen auf den Registerkarten Ereignisse, Ressourcen und Abweichungen für einen bestimmten Stack anzeigen und sie sind auch in der Änderungssatzvorschau enthalten.

Module sind von Ressourcen in einer Vorlage unterscheidbar, da sie der folgenden vierteiligen Namenskonvention entsprechen, im Gegensatz zu der typischen dreiteiligen Konvention, die von Ressourcen verwendet wird:

```
organization::service::use-case::MODULE
```

Verwenden Sie Parameter, um Modulwerte anzugeben

In können Sie Vorlagenparameter verwenden CloudFormation, um Ihre Stacks anzupassen, indem Sie bei der Erstellung oder Aktualisierung des Stacks Eingabewerte angeben. Mit diesen Parametern können Sie bestimmte Aspekte des Stacks an Ihre Bedürfnisse anpassen. Weitere Informationen zur Definition von Vorlagenparametern finden Sie unterCloudFormation ParametersVorlagensyntax.

In ähnlicher Weise können Module auch Parameter haben. Mit diesen Modulparametern können Sie benutzerdefinierte Werte aus der Vorlage (oder einem anderen Modul), das sie verwendet, in das Modul eingeben. Das Modul kann diese benutzerdefinierten Werte dann verwenden, um Eigenschaftswerte für die darin enthaltenen Ressourcen festzulegen.

Sie können auch Vorlagenparameter definieren, die Moduleigenschaften festlegen, sodass Sie Werte eingeben können, die zum Zeitpunkt des Stack-Vorgangs an das Modul übergeben werden.

Wenn ein Modul ein verschachteltes Modul mit eigenen Modulparametern enthält, können Sie entweder:

- Werte für die Parameter des verschachtelten Moduls direkt im übergeordneten Modul angeben.

- Entsprechende Modulparameter im übergeordneten Modul definieren, die es ermöglichen, die Parameter des verschachtelten Moduls durch die Vorlage (oder das Modul) festzulegen, in der das übergeordnete Modul enthalten ist.

Verwenden von Vorlagenparametern zur Angabe von Modulparameterwerten

Das folgende Beispiel zeigt, wie Vorlagenparameter definiert werden, die Werte an ein Modul übergeben.

Diese Vorlage, die `My::S3::SampleBucket::MODULE` einen Vorlagenparameter enthält`BucketName`, ermöglicht es dem Benutzer, während des Stack-Vorgangs einen S3-Bucket-Namen anzugeben.

```
# Template containing My::S3::SampleBucket::MODULE
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: 'My::S3::SampleBucket::MODULE'
    Properties:
      BucketName: !Ref BucketName
```

Festlegen von Eigenschaften für Ressourcen in einem untergeordneten Modul aus dem übergeordneten Modul

Das folgende Beispiel veranschaulicht, wie Parameterwerte in einem Modul angegeben werden, das in einem anderen Modul verschachtelt ist.

Dieses erste Modul, `My::S3::SampleBucketPrivate::MODULE`, ist das untergeordnete Modul. Es definiert zwei Parameter: `BucketName` und `AccessControl`. Die für diese Parameter angegebenen Werte werden verwendet, um die Eigenschaften `BucketName` und `AccessControl` der Ressource `AWS::S3::Bucket`, die das Modul enthält, anzugeben. Unten sehen Sie das Vorlagenfragment für `My::S3::SampleBucketPrivate::MODULE`.

```
# My::S3::SampleBucketPrivate::MODULE
AWSTemplateFormatVersion: 2010-09-09
Description: A sample S3 Bucket with Versioning and DeletionPolicy.
```

```
Parameters:
  BucketName:
    Description: Name for the bucket
    Type: String
  AccessControl:
    Description: AccessControl for the bucket
    Type: String
Resources:
  S3Bucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: !Ref BucketName
      AccessControl: !Ref AccessControl
      DeletionPolicy: Retain
      VersioningConfiguration:
        Status: Enabled
```

Als Nächstes ist das vorherige Modul in ein übergeordnetes Modul, `My::S3::SampleBucket::MODULE`, verschachtelt. Das übergeordnete Modul, `My::S3::SampleBucket::MODULE`, legt die Parameter des untergeordneten Moduls auf folgende Weise fest:

- Sie legt den `AccessControl` Parameter `My::S3::SampleBucketPrivate::MODULE` auf fest`Private`.
- Für `BucketName` definiert es einen Modulparameter, der es ermöglicht, den Bucket-Namen in der Vorlage (oder dem Modul) anzugeben, die/das `My::S3::SampleBucket::MODULE` enthält.

```
# My::S3::SampleBucket::MODULE
AWSTemplateFormatVersion: 2010-09-09
Description: A sample S3 Bucket. With Private AccessControl.
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: 'My::S3::SampleBucketPrivate::MODULE'
    Properties:
      BucketName: !Ref BucketName
      AccessControl: Private
```

Angeben von Beschränkungen für Modulparameter

Modulparameter unterstützen die Durchsetzung von Beschränkungen nicht. Um die Einschränkungsprüfung für einen Modulparameter durchzuführen, erstellen Sie einen Vorlagenparameter mit den gewünschten Einschränkungen. Verweisen Sie dann in Ihrem Modulparameter auf diesen Vorlagenparameter. Weitere Hinweise zur Definition von Vorlagenparametern finden Sie unterCloudFormation ParametersVorlagensyntax.

Referenzmodulressourcen in CloudFormation Vorlagen

In CloudFormation Vorlagen müssen Sie häufig Eigenschaften für eine Ressource festlegen, die auf dem Namen oder der Eigenschaft einer anderen Ressource basieren. Weitere Informationen finden Sie unter Ressourcen referenzieren.

Um auf eine Ressource zu verweisen, die in einem Modul in Ihrer CloudFormation Vorlage enthalten ist, müssen Sie zwei logische Namen kombinieren:

- Der logische Name, den Sie dem Modul selbst gegeben haben, als Sie es in Ihre Vorlage aufgenommen haben.
- Der logische Name der spezifischen Ressource innerhalb dieses Moduls.

Sie können diese beiden logischen Namen mit oder ohne einen Punkt (.) kombinieren. Lautet der logische Name des Moduls beispielsweise `MyModule` und der logische Name der Ressource lautet`MyBucket`, können Sie diese Ressource entweder als `MyModule.MyBucket` oder bezeichnen`MyModuleMyBucket`.

Um die logischen Namen von Ressourcen innerhalb eines Moduls zu finden, können Sie das Schema des Moduls konsultieren, das in der CloudFormation Registrierung verfügbar ist, oder indem Sie den DescribeTypeVorgang verwenden. Das Schema listet alle Ressourcen und ihre logischen Namen auf, die Teil des Moduls sind.

Sobald Sie den vollständigen logischen Namen haben, können Sie CloudFormation Funktionen wie `GetAtt` und verwenden, `Ref` um auf Eigenschaftswerte von Modulressourcen zuzugreifen.

Sie haben beispielsweise ein `My::S3::SampleBucket::MODULE` Modul, das eine `AWS::S3::Bucket` Ressource mit dem logischen Namen enthält`S3Bucket`. Um mit der `Ref` Funktion auf den Namen dieses Buckets zu verweisen, kombinieren Sie den Namen des Moduls in Ihrer Vorlage (`MyBucket`) mit dem logischen Namen der Ressource im Modul (`S3Bucket`). Der vollständige logische Name ist entweder `MyBucket.S3Bucket` oder`MyBucketS3Bucket`.

-Beispielvorlage

Die folgende Beispielvorlage erstellt mithilfe des `My::S3::SampleBucket::MODULE` Moduls einen S3-Bucket. Es erstellt auch eine Amazon SQS SQS-Warteschlange und legt ihren Namen so fest, dass er mit dem Bucket-Namen aus dem Modul übereinstimmt. Darüber hinaus gibt die Vorlage den Amazon-Ressourcennamen (ARN) des erstellten S3-Buckets aus.

```
# Template that uses My::S3::SampleBucket::MODULE
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: My::S3::SampleBucket::MODULE
    Properties:
      BucketName: !Ref BucketName
  exampleQueue:
    Type: AWS::SQS::Queue
    Properties:
      QueueName: !Ref MyBucket.S3Bucket
Outputs:
  BucketArn:
    Value: !GetAtt MyBucket.S3Bucket.Arn
```

Verwaltung von AWS Ressourcen als eine Einheit mit AWS CloudFormation Stacks

Ein Stapel ist eine Sammlung von AWS Ressourcen, die Sie als eine Einheit verwalten können. Mit anderen Worten, Sie können eine Sammlung von Ressourcen erstellen, aktualisieren und löschen, indem Sie Stapel erstellen, aktualisieren und löschen.

Das Erstellen eines Stacks beinhaltet die Bereitstellung einer CloudFormation Vorlage, die die Ressourcen und ihre Konfigurationen spezifiziert, die CloudFormation dann bereitgestellt und konfiguriert wird.

Beim Aktualisieren eines Stacks müssen Änderungen an der Vorlage oder den Parametern vorgenommen werden. CloudFormation vergleicht die von Ihnen eingereichten Änderungen mit dem aktuellen Status Ihres Stacks und aktualisiert nur die geänderten Ressourcen. CloudFormation kann Ressourcen unterbrechen oder aktualisierte Ressourcen ersetzen, je nachdem, welche Eigenschaften Sie aktualisieren. Weitere Informationen zum Verhalten von Ressourcenaktualisierungen finden Sie unter Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen.

CloudFormation bietet zwei Methoden zum Aktualisieren von Stacks:

- Änderungssätze — Mit Änderungssätzen können Sie eine Vorschau der Änderungen, die Sie an Ihrem Stack vornehmen CloudFormation werden, anzeigen und dann entscheiden, ob Sie diese Änderungen übernehmen möchten. Änderungssätze sind Dokumente im JSON-Format, die die Änderungen, die an einem Stapel vorgenommen CloudFormation werden, zusammenfassen. Verwenden Sie Änderungssätze, wenn Sie sicherstellen möchten, dass CloudFormation keine unbeabsichtigten Änderungen vorgenommen werden, oder wenn Sie mehrere Optionen in Betracht ziehen möchten. Sie können beispielsweise einen Änderungssatz verwenden, um sicherzustellen, dass bei einem Update die Datenbankinstanzen Ihres Stacks CloudFormation nicht ersetzt werden.
- Direktes Update — Wenn Sie einen Stack direkt aktualisieren, reichen Sie Änderungen ein und stellen sie CloudFormation sofort bereit. Verwenden Sie direkte Updates, wenn Ihre Aktualisierungen schnell bereitgestellt werden sollen.

Durch das Löschen eines Stacks werden die damit verknüpften Ressourcen gelöscht. Ein Stack kann z. B. alle erforderlichen Ressourcen für die Ausführung einer Webanwendung enthalten, z. B.

einen Webserver, eine Datenbank und Netzwerkregeln. Wenn Sie diese Webanwendung nicht mehr benötigen, können Sie den Stack einfach löschen und alle verwandten Ressourcen werden gelöscht.

> **Note**
>
> Sie zahlen für die Dauer des Betriebs der Stack-Ressourcen (auch wenn Sie den Stack sofort gelöscht haben).

CloudFormation stellt sicher, dass alle Stack-Ressourcen entsprechend erstellt oder gelöscht werden. Da CloudFormation die Stack-Ressourcen als einzelne Einheit behandelt, müssen sie alle erfolgreich erstellt oder gelöscht werden, damit der Stack erstellt oder gelöscht wird. Wenn eine Ressource nicht erstellt werden kann, setzt CloudFormation den Stack zurück und löscht automatisch alle erstellten Ressourcen. Wenn eine Ressource nicht gelöscht werden kann, werden alle verbleibenden Ressourcen beibehalten, bis der Stack erfolgreich gelöscht werden kann.

Themen

- Schnittstellen für die Verwaltung Ihrer Stacks
- Einen Stack von der CloudFormation Konsole aus erstellen
- Stack-Informationen von der CloudFormation Konsole aus anzeigen
- Aktualisiere deine Stack-Vorlage
- Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen
- CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren
- Stacks direkt aktualisieren
- Ein Stack-Update abbrechen
- Löschen Sie einen Stack aus der CloudFormation Konsole
- Stack-Fortschritt überwachen
- Setzen Sie Ihren CloudFormation Stack bei einem Alarmverstoß mit Rollback-Triggern zurück
- Erkennen Sie mit Drift-Erkennung nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen
- AWS Ressourcen in einen CloudFormation Stack importieren
- Stack-Refactoring
- Unterstützung für Ressourcentypen

- Verwenden Sie Schnellerstellungslinks, um Stapel zu erstellen CloudFormation
- Beispiele für CloudFormation Stack-Operationsbefehle für AWS CLI und PowerShell

Schnittstellen für die Verwaltung Ihrer Stacks

Sie können Ihre CloudFormation Stacks über die folgenden Schnittstellen verwalten:

- CloudFormation Konsole — Stellt eine Weboberfläche bereit, über die Sie auf Ihre Stacks zugreifen können. Sie können auf die CloudFormation Konsole zugreifen, indem Sie sich bei der anmelden AWS Management Console, das Suchfeld in der Navigationsleiste verwenden CloudFormation, um nach den Suchergebnissen zu suchen und dann CloudFormationaus den Suchergebnissen eine Auswahl treffen.
- AWS Command Line Interface— Stellt Befehle für eine Vielzahl von AWS Diensten bereit, darunter Windows CloudFormation, Mac und Linux, und wird unter Windows, Mac und Linux unterstützt. Informationen zu den CloudFormation Befehlen finden Sie unter cloudformation in der AWS CLI Befehlsreferenz.
- AWS -Tools für PowerShell— Eine Reihe von PowerShell Modulen, die auf der Funktionalität von aufbauen. SDK für .NET Mit den Tools für PowerShell können Sie über die PowerShell Befehlszeile Skripts für Operationen auf Ihren AWS Ressourcen erstellen. Die Cmdlets für finden Sie CloudFormation in der Cmdlet-Referenz AWS -Tools für PowerShell .
- Abfrage-API – Bietet API-Aktionen auf niedriger Ebene, die Sie mithilfe von HTTPS-Anforderungen aufrufen. Wenn Sie in Ihrer Anwendung API-Aufrufe tätigen, müssen Sie den Code für Details auf niedriger Ebene schreiben, z. B. für die Generierung des Hashs zum Signieren der Anforderung. Weitere Informationen zu den API-Aktionen für CloudFormation finden Sie unter Aktionen in der AWS CloudFormation API-Referenz.
- AWS SDKs— Bietet sprachspezifische Funktionen APIs und kümmert sich um viele Verbindungsdetails, wie z. B. die Berechnung von Signaturen, die Bearbeitung von Wiederholungsversuchen von Anfragen und die Fehlerbehandlung. Weitere Informationen finden Sie unter Tools to Build on AWS .
- AWS Cloud Development Kit (AWS CDK)— Das AWS CDK ist ein Open-Source-Framework für die Softwareentwicklung, mit dem Sie die AWS Infrastruktur mithilfe vertrauter Programmiersprachen wie Python TypeScript, Java und .NET definieren können. Mit dem CDK können Sie Ihre Anwendungsressourcen modellieren und sie dann CloudFormation direkt von Ihrer integrierten Entwicklungsumgebung (IDE) aus bereitstellen. Weitere Informationen finden Sie unter AWS Cloud Development Kit (AWS CDK).

Einen Stack von der CloudFormation Konsole aus erstellen

Sie können eine Stack-Vorlage erstellen und diese dann verwenden, um mithilfe der CloudFormation Konsole oder eines Befehlszeilentools einen Stack zu erstellen. Die Konsole bietet eine assistentengesteuerte Oberfläche mit vordefinierten Optionen, die den Prozess der Stack-Erstellung rationalisiert.

Themen

- Erstellen eines Stacks
- Konfigurieren Sie die Stack-Optionen
- Sehen Sie sich eine Vorschau der Konfiguration Ihres Stacks an

Erstellen eines Stacks

Folgen Sie den Schritten in diesem Abschnitt, um Ihre Vorlage bereitzustellen und einen Stack zu erstellen.

Voraussetzungen

- Sie müssen eine Stack-Vorlage erstellt haben. Weitere Informationen finden Sie unter Mit CloudFormation Vorlagen arbeiten.

Um einen Stack (Konsole) zu erstellen

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, in der der Stapel erstellt werden AWS-Region soll.
3. Wählen Sie auf der Seite „Stapel“ oben rechts die Option „Stapel erstellen“ und dann „Mit neuen Ressourcen (Standard)“ aus.

   Alternativ können Sie die Option Mit vorhandenen Ressourcen (Importressourcen) wählen, um vorhandene AWS Ressourcen zu importieren, die in Ihrer Vorlage beschrieben sind. Weitere Informationen zu dieser Option finden Sie unterAWS Ressourcen in einen CloudFormation Stack importieren.
4. Führen Sie auf der Seite „Stack erstellen“ einen der folgenden Schritte aus:

- Um eine vorhandene Vorlage zu verwenden, wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorhandene Vorlage auswählen aus. Wählen Sie dann unter Vorlage angeben entweder Amazon S3 S3-URL oder Laden Sie eine Vorlagendatei basierend auf dem Speicherort der Vorlage hoch.
  - Wenn Sie Amazon S3 S3-URL wählen, geben Sie eine URL zur Vorlagendatei in einem S3-Bucket an.

    Wenn Ihre Vorlage verschachtelte Stapel enthält (z. B. Stapel, die in anderen Vorlagendokumenten in Unterverzeichnissen beschrieben sind), stellen Sie sicher, dass Ihr S3-Bucket die erforderlichen Dateien und Verzeichnisse enthält.

    Wenn Sie über eine Vorlage aus einem Bucket mit aktivierter Versionierung verfügen, können Sie eine bestimmte Version der Vorlage angeben, indem Sie sie an die URL anhängen. `?versionId=`*`version-id`* Weitere Informationen zu Buckets mit aktivierter Versionierung finden Sie unter Arbeiten mit Objekten in einem Bucket mit aktivierter Versionierung im Amazon Simple Storage Service-Benutzerhandbuch.

    Die URL muss auf eine Vorlage mit einer maximalen Größe von 1 MB verweisen, die in einem S3-Bucket gespeichert ist, für den Sie Leseberechtigungen haben. Die URL darf maximal 1 024 Zeichen lang sein. Für einige Ressourcen muss sich der Bucket möglicherweise in derselben Region befinden wie der Stack.
  - Wenn Sie eine Vorlagendatei hochladen wählen, wählen Sie Datei auswählen, um eine Vorlagendatei von Ihrem lokalen Computer auszuwählen. Die Größe der Vorlagendatei muss 1 MB oder weniger betragen.

    Sobald Sie Ihre Vorlage ausgewählt haben, CloudFormation lädt die Datei hoch und zeigt die S3-URL an. CloudFormation lädt es in einen Amazon S3 S3-Bucket in Ihrem AWS-Konto hoch. Wenn Sie bereits einen S3-Bucket haben, der von CloudFormation erstellt wurde AWS-Konto, CloudFormation fügt die Vorlage diesem Bucket hinzu. Wenn Sie noch nicht über einen bereits CloudFormation erstellten Bucket verfügen, wird für jede Region, in die Sie eine Vorlagendatei hochladen, ein eigener Bucket erstellt.

    Folgendes sollten Sie bei der Verwendung der von erstellten S3-Buckets beachten: CloudFormation

    - Auf die Buckets kann jeder zugreifen, der über Amazon S3 S3-Berechtigungen in Ihrem AWS-Konto Unternehmen verfügt.

- CloudFormation erstellt die Buckets mit standardmäßig aktivierter serverseitiger Verschlüsselung, wodurch alle im Bucket gespeicherten Objekte verschlüsselt werden.

  Sie können die Verschlüsselungsoptionen für Buckets, die erstellt CloudFormation wurden, direkt verwalten, z. B. mit der Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/, oder dem AWS CLI. Weitere Informationen finden Sie unter Einstellung des standardmäßigen serverseitigen Verschlüsselungsverhaltens für Amazon S3 S3-Buckets im Amazon Simple Storage Service-Benutzerhandbuch.
- Sie können Ihren eigenen Bucket verwenden und dessen Berechtigungen durch manuelles Hochladen von Vorlagen in Amazon S3 verwalten. Geben Sie die Amazon S3-URL einer Vorlagendatei an, wenn Sie einen Stack erstellen oder aktualisieren.

- Wenn Sie noch keine Vorlage parat haben, können Sie Build aus Infrastructure Composer wählen, um eine Vorlage mit Infrastructure Composer zu erstellen. Weitere Informationen finden Sie unter Infrastructure Composer.

5. Wählen Sie Weiter, um fortzufahren und die Vorlage zu validieren.

   Bevor Sie fortfahren, CloudFormation validiert Ihre Vorlage, um syntaktische und einige semantische Fehler, wie z. B. zirkuläre Abhängigkeiten, zu erkennen. Prüft während der Validierung CloudFormation zunächst, ob es sich bei der Vorlage um ein gültiges JSON handelt. Ist dies nicht der Fall, wird CloudFormation geprüft, ob es sich bei der Vorlage um ein gültiges YAML handelt. Wenn beide Prüfungen fehlschlagen, wird ein Fehler bei der Vorlagenvalidierung CloudFormation zurückgegeben.
6. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) einen Stack-Namen in das Feld Stack name (Stack-Name) ein.

   Der Stack-Name ist eine Kennung, mit der Sie einen bestimmten Stack in einer Liste von Stacks finden können. 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.
7. Geben Sie im Abschnitt Parameter Werte für die Parameter an, die in der Vorlage definiert wurden.
8. Wählen Sie Weiter, um mit der Erstellung des Stacks fortzufahren.
9. (Optional) Ändern Sie auf der Seite „Stack-Optionen konfigurieren“ die Standard-Stack-Optionen. Weitere Informationen finden Sie unter Konfigurieren Sie die Stack-Optionen.
10. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie unter Capabilities die Option I acknowledge that this template may create IAM-resources aus, um anzugeben, dass Sie

IAM-Ressourcen in der Vorlage verwenden möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.

11. Wählen Sie Next (Weiter), um fortzufahren.
12. Überprüfen Sie auf der Seite Überprüfen und erstellen die Details Ihres Stacks.

    Um einen der Werte zu ändern, bevor Sie den Stack starten, wählen Sie Bearbeiten in dem Abschnitt, der die Einstellung enthält, die Sie ändern möchten.
13. (Optional) Sie können einen Änderungssatz erstellen, um eine Vorschau der Konfiguration des Stacks anzuzeigen, bevor Sie ihn erstellen. Wählen Sie auf der Seite Überprüfen und erstellen die Option Änderungssatz erstellen aus und folgen Sie den Anweisungen. Weitere Informationen finden Sie unter Sehen Sie sich eine Vorschau der Konfiguration Ihres Stacks an.
14. Wählen Sie Senden, um Ihren Stack zu starten.

CloudFormation fährt dann mit der Erstellung aller in der Vorlage definierten Ressourcen fort.

Sie können den Fortschritt und den Status der Stack-Erstellung auf der Registerkarte Ereignisse für Ihren neuen Stack überwachen. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

Um einen Stack über die Befehlszeile zu erstellen

Verwenden Sie einen der folgenden Befehle:

- create-stack (AWS CLI)
- New-CFNStack (AWS Tools for Windows PowerShell)

Beispiele für die Verwendung der Befehlszeile zum Erstellen eines Stacks finden Sie unterBeispiele für CloudFormation Stack-Operationsbefehle für AWS CLI und PowerShell.

Konfigurieren Sie die Stack-Optionen

Auf der Seite Stack-Optionen konfigurieren können Sie Optionen für Ihre CloudFormation Stacks wie Tags, Benachrichtigungen über Stack-Ereignisse oder eine Stack-Richtlinie konfigurieren.

Sie können die folgenden Stack-Optionen einrichten:

Tags

Sie können Ihrem Stack und allen Ressourcen, für die Tagging CloudFormation unterstützt wird, bis zu 50 Tag-Schlüsselpaare hinzufügen. Ein Tag ist ein vom Kunden definierter Schlüssel und Wert, der AWS Ressourcen beispielsweise zur Kostenverfolgung zugewiesen werden kann.

Ein Schlüssel besteht aus alphanumerischen Zeichen oder Leerzeichen. Tag-Schlüssel können aus bis zu 127 Zeichen bestehen.

Ein Wert besteht aus alphanumerischen Zeichen oder Leerzeichen. Tag-Werte können aus bis zu 255 Zeichen bestehen.

Nach der Stack-Erstellung wird durch Hinzufügen, Aktualisieren oder Entfernen von Tags auf Stack-Ebene ein Stack-Update ausgelöst. Alle Ressourcen, die die Tag-Propagierung auf Stack-Ebene unterstützen, werden entsprechend aktualisiert.

Berechtigungen

Eine bestehende IAM-Servicerolle, die übernommen werden kann. CloudFormation Anstatt Ihre Kontoanmeldedaten zu verwenden, werden die Anmeldeinformationen der Rolle CloudFormation verwendet, um Ihren Stack zu erstellen. Weitere Informationen finden Sie unter AWS CloudFormation Servicerolle.

Stack-Fehleroptionen

Gibt Fehleroptionen für die Bereitstellung für alle Stack-Bereitstellungen und Änderungssatz-Vorgänge an. Weitere Informationen finden Sie unter Wählen Sie aus, wie mit Ausfällen bei der Bereitstellung von Ressourcen umgegangen werden soll.

Die Rollback aller Stack-Ressourcen Option setzt alle in der Vorlage angegebenen Ressourcen zurück, wenn der Stack-Status `CREATE_FAILED` oder `UPDATE_FAILED` ist.

Bei Erstellungs-Vorgängen behält die Option Erfolgreich bereitgestellte Ressourcen beibehalten den Status erfolgreicher Ressourcen bei, während fehlerhafte Ressourcen in einem fehlerhaften Status bleiben, bis der nächste Aktualisierungs-Vorgang durchgeführt wird.

Bei Aktualisierungs- und Änderungssatz-Vorgängen behält die Option Erfolgreich bereitgestellte Ressourcen beibehalten den Status erfolgreicher Ressourcen bei, während fehlerhafte Ressourcen auf den letzten bekannten stabilen Status zurückgesetzt werden. Fehlgeschlagene Ressourcen befinden sich in einem `UPDATE_FAILED`-Status. Ressourcen ohne letzten bekannten stabilen Zustand werden beim nächsten Stack-Vorgang gelöscht.

Sie können auch die folgenden erweiterten Optionen für die Stack-Erstellung festlegen:

Stack-Richtlinie

Legt die Ressourcen, fest die Sie vor unbeabsichtigten Aktualisierungen während einer Stack-Aktualisierung schützen möchten. Standardmäßig können alle Ressourcen bei einer Stack-Aktualisierung aktualisiert werden.

Sie können die Stack-Richtlinie direkt im JSON-Format eingeben oder eine JSON-Datei, die die Stack-Richtlinie enthält, hochladen. Weitere Informationen finden Sie unter Verhindern von Aktualisierungen der Stack-Ressourcen.

Rollback-Konfiguration

Sie können den Status Ihres Stacks während der Erstellung und Aktualisierung des Stacks CloudFormation überwachen und diesen Vorgang rückgängig machen, wenn der Stack den Schwellenwert eines der von Ihnen angegebenen Alarme überschreitet. Geben Sie die CloudWatch Alarme an, die überwacht CloudFormation werden sollen. Wenn einer der Alarme während des Stack-Vorgangs oder des Überwachungszeitraums in den `ALARM` Status wechselt, CloudFormation wird der gesamte Stack-Vorgang zurückgesetzt. Weitere Informationen finden Sie unter Setzen Sie Ihren CloudFormation Stack bei einem Alarmverstoß mit Rollback-Triggern zurück.

Benachrichtigungs-Optionen

Sie können ein neues oder bestehendes Amazon Simple Notification Service-Thema angeben, an das Benachrichtigungen zu Stack-Ereignissen gesendet werden.

Wenn Sie ein Amazon SNS-Thema erstellen, müssen Sie einen Namen und eine E-Mail-Adresse angeben, an die Stack-Ereignisbenachrichtigungen gesendet werden sollen.

Stack-Erstellungsoptionen

Die folgenden Optionen sind für die Stack-Erstellung enthalten, sind jedoch nicht als Teil von Stack-Aktualisierungen verfügbar.

Timeout (Zeitüberschreitung)

Gibt die Zeitspanne in Minuten an, die bis zum Timeout bei der Stapelerstellung zur Verfügung stehen CloudFormation soll. Wenn der gesamte Stapel in der zugewiesenen Zeit nicht erstellt werden CloudFormation kann, schlägt die Stapelerstellung aufgrund eines Timeouts fehl und der Stapel wird zurückgesetzt.

Standardmäßig gibt es keinen Timeout für die Stack-Erstellung. Allerdings gibt es möglicherweise für einzelne Ressourcen einen Timeout, basierend auf der Art des von ihnen implementierten Services. Wenn beispielsweise eine einzelne Ressource in Ihrem Stack abläuft, läuft die Stack-Erstellung ebenfalls aus, auch wenn der von Ihnen angegebene Timeout für die Stack-Erstellung noch nicht erreicht wurde.

Termination protection

Verhindert das versehentliche Löschen eines Stacks. Wenn ein Benutzer versucht, einen Stack mit aktiviertem Beendigungsschutz zu löschen, schlägt das Löschen fehl und der Stack – einschließlich seines Status – bleibt unverändert. Weitere Informationen finden Sie unter CloudFormation Stapel vor dem Löschen schützen.

Der Beendigungsschutz ist standardmäßig auf Disabled (Deaktiviert) gesetzt.

Sehen Sie sich eine Vorschau der Konfiguration Ihres Stacks an

Um vor der Erstellung des CloudFormation Stacks eine Vorschau der Konfiguration eines Stacks anzuzeigen, erstellen Sie einen Änderungssatz. Durch diese Funktionalität können Sie verschiedene Konfigurationen untersuchen und Korrekturen und Änderungen an Ihrem Stack vornehmen, bevor Sie den Änderungssatz ausführen. Weitere Informationen zu Änderungssätzen finden Sie unter CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren.

Erstellen eines Änderungssatzes für einen neuen Stack

Zum Erstellen eines Änderungssatzes für einen neuen Stack wählen Sie Ihre Stack-Vorlage aus und geben Sie die Konfiguration Ihres Stacks an, wie Sie das auch beim Erstellen eines neuen Stacks tun würden. Wählen Sie dann einen neuen Änderungssatz anstatt eines neuen Stacks aus.

Um einen Änderungssatz für einen neuen Stack zu erstellen

1. Wählen Sie auf der Seite Überprüfen und erstellen die Option Änderungssatz erstellen aus.
2. Geben Sie im Dialogfeld Änderungssatz erstellen einen Namen für den Änderungssatz und, falls gewünscht, eine Beschreibung ein. Wählen Sie Create change set.

   Wenn Sie einen Änderungssatz für einen neuen Stapel erstellen, CloudFormation geht Folgendes vor:

   - Startet einen neuen Stack mit dem Status `REVIEW_IN_PROGRESS`.

- Erstellt einen Änderungssatz für den neuen Stack, der die Stack-Konfiguration darstellt, die Sie in den vorherigen Schritten angegeben haben.

CloudFormation zeigt die Seite „Änderungssätze“ für den vorgeschlagenen Stack an. Während der Änderungssatz CloudFormation erstellt wird`CREATE_IN_PROGRESS`, lautet sein Status und sein Ausführungsstatus lautet`UNAVAILABLE`. Wenn die Erstellung des Änderungssatzes erfolgreich CloudFormation abgeschlossen ist, wird der Änderungssatzstatus auf gesetzt`CREATE_COMPLETE`, und sein Ausführungsstatus lautet`AVAILABLE`. Der Stack-Status wird auf `REVIEW_IN_PROGRESS` aktualisiert. An diesem Punkt können Sie den Änderungssatz ausführen, um die Erstellung des neuen Stacks abzuschließen.

CloudFormation Zeigt im Bereich Änderungen die vorgeschlagene Konfiguration Ihres Stacks an.

Wenn CloudFormation der Änderungssatz nicht erstellt werden kann, wird der Status des Änderungssatzes auf gesetzt`CREATE_FAILED`. Korrigieren Sie den Fehler, der im Feld Status reason (Statusgrund) angezeigt wird, und erstellen Sie dann einen neuen Änderungssatz. In dieser Phase können Sie verschiedene Konfigurationen ausprobieren und Korrekturen und Änderungen an Ihrem Stack vornehmen, bevor Sie den nächsten Änderungssatz ausführen.

3. Um die Erstellung eines neuen Stacks basierend auf dem Änderungssatz abzuschließen, wählen Sie Execute (Ausführen), geben Sie Ihre Rollback-Konfiguration an und wählen Sie Execute change set (Änderungssatz ausführen).

Stack-Informationen von der CloudFormation Konsole aus anzeigen

Nachdem Sie einen CloudFormation Stack erstellt haben, können Sie den verwenden, AWS Management Console um dessen Daten und Ressourcen einzusehen.

Um Informationen über Ihren CloudFormation Stack anzuzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm aus, AWS-Region wo sich der Stapel befindet.
3. Wählen Sie auf der Seite Stacks den Stack-Namen aus. CloudFormation zeigt die Stack-Details für den ausgewählten Stack an.

> **Note**
>
> Wenn der Stapel gelöscht wurde, können Sie ihn mithilfe der Option Status filtern finden. Weitere Informationen finden Sie unter Gelöschte Stacks von der CloudFormation Konsole aus anzeigen.

Sie können die folgenden Stack-Informationen anzeigen:

Stack-Informationen

Zeigt allgemeine Informationen über den Stack und dessen Konfiguration an, einschließlich folgender Informationen:

Übersicht

Zeigt den Stack-Namen, die Stack-ID, den Root-Stack und die IAM-Rolle zusammen mit Statusinformationen wie Stack-Status, Drift-Status und Terminierungsschutz an.

Tags

Zeigt die Tags an, die dem Stack zugeordnet sind.

Stack-Richtlinie

Beschreibt die Stack-Ressourcen, die gegen Stack-Aktualisierungen geschützt sind. Um diese Ressourcen aktualisieren zu können, müssen sie explizit für eine Stack-Aktualisierung zugelassen werden. Weitere Informationen finden Sie unter Verhindern von Aktualisierungen der Stack-Ressourcen.

Rollback-Konfiguration

Zeigt alle von Ihnen angegebenen CloudWatch Alarme an, die während des Stack-Vorgangs oder des angegebenen Überwachungszeitraums überwacht werden CloudFormation sollen. Wenn einer der Alarme während des Stack-Vorgangs oder des Überwachungszeitraums in den `ALARM` Status wechselt, CloudFormation wird der gesamte Stack-Vorgang zurückgesetzt. Weitere Informationen finden Sie unter Setzen Sie Ihren CloudFormation Stack bei einem Alarmverstoß mit Rollback-Triggern zurück.

Benachrichtigungs-Optionen

Zeigt das Amazon Simple Notification Service-Thema an, in dem Benachrichtigungen über Stack-Ereignisse gesendet werden, falls angegeben.

--Ereignisse

Zeigt die Operationen an, die verfolgt werden, wenn Sie den Stack erstellen, aktualisieren oder löschen. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

Allen Ereignissen, die von einer bestimmten Stack-Operation ausgelöst werden, wird dasselbe Client-Anfrage-Token zugewiesen, das Sie verwenden können, um Operationen zu verfolgen. Stack-Operationen, die von der Konsole aus initiiert werden, verwenden das Token-Format Console- StackOperation -ID, sodass Sie den Stack-Vorgang leicht identifizieren können. Wenn Sie beispielsweise einen Stack unter Verwendung der Konsole erstellen, würde jedem resultierenden Stack-Ereignis dasselbe Token im folgenden Format zugewiesen: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Ressourcen

Zeigt die Ressourcen an, die Teil des Stacks sind.

Outputs

Zeigt die Ausgaben an, die in der Stack-Vorlage deklariert wurden. Weitere Informationen finden Sie unter Holen Sie sich exportierte Ausgaben aus einem bereitgestellten CloudFormation Stack.

Parameter

Zeigt die Stack-Parameter und deren Werte an.

Bei Stacks, die Systems Manager Manager-Parameter enthalten, werden in der Spalte Aufgelöster Wert die Werte angezeigt, die in der Stackdefinition für die Systems Manager Manager-Parameter verwendet werden. Weitere Informationen finden Sie unter Geben Sie vorhandene Ressourcen zur Laufzeit mit von CloudFormation -bereitgestellten Parametertypen an.

Vorlage

Zeigt die Stack-Vorlage an.

Wählen Sie für Stapel, die Makros enthalten, Originalvorlage anzeigen, um die vom Benutzer eingereichte Vorlage anzuzeigen, oder Verarbeitete Vorlage anzeigen, um die Vorlage anzuzeigen, nachdem CloudFormation die referenzierten Makros verarbeitet wurden. CloudFormationverwendet die verarbeitete Vorlage, um Ihren Stapel zu erstellen oder zu aktualisieren.

Änderungssätze

Zeigt die Änderungssätze des Stacks an.

Weitere Informationen finden Sie unter Einen Änderungssatz für einen CloudFormation Stack anzeigen.

Git-Synchronisierung

Zeigt das Git-Sync-Dashboard des Stacks an.

Weitere Informationen finden Sie unter Status-Dashboard für Git-Synchronisierung.

Aktualisiere deine Stack-Vorlage

Um die Ressourcen oder Eigenschaften in einem CloudFormation Stack zu ändern, müssen Sie die Vorlage des Stacks aktualisieren. Beginnen Sie mit der vorhandenen Vorlage für diesen Stapel und nehmen Sie Ihre Änderungen daran vor. Wenn Sie die Vorlage in einem Quellcodeverwaltungssystem gespeichert haben, verwenden Sie eine Kopie davon als Ausgangspunkt. Andernfalls können Sie eine Kopie der Vorlage von erhalten CloudFormation.

Wenn Sie nur die Parameter oder Einstellungen des Stacks ändern möchten (z. B. das Amazon SNS SNS-Thema eines Stacks), können Sie die vorhandene Vorlage wiederverwenden, ohne eine Kopie zu erhalten.

Sie können eine CloudFormation Stack-Vorlage aktualisieren, indem Sie einen Texteditor verwenden oderInfrastructure Composer.

Um eine vorhandene Stack-Vorlage mithilfe von Infrastructure Composer zu aktualisieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der Seite Stacks den Namen des zu aktualisierenden Stacks aus.
3. Wählen Sie die Registerkarte „Vorlage“ und dann „In Infrastructure Composer anzeigen“.

   CloudFormation öffnet die Vorlage in Infrastructure Composer.
4. Aktualisieren Sie Ihre Vorlage mit einer der folgenden Methoden:

   - Canvas-Oberfläche: Hier können Sie per Drag-and-Drop aus der Ressourcen-Palette ziehen. Konfigurieren Sie Ressourcen, indem Sie auf eine Karte doppelklicken, um den Bereich mit den Ressourceneigenschaften zu öffnen. Connect Ressourcen nach Bedarf. Ausführliche Anweisungen zur Verwendung der Canvas-Oberfläche finden Sie unter How to compose in AWS-Infrastruktur-Composer.

- Vorlagenschnittstelle: Wechseln Sie von der Canvas-Oberfläche zur Template-Oberfläche. Nehmen Sie Inline-Aktualisierungen am Vorlagencode vor. Wechseln Sie nach Bedarf zwischen JSON- und YAML-Formaten.

5. Wählen Sie Validieren, um die Vorlage auf Syntaxfehler zu überprüfen.
6. Wenn Sie bereit sind, die Änderungen zu exportieren CloudFormation, wählen Sie Vorlage aktualisieren.

Um eine bestehende Stack-Vorlage zu aktualisieren, verwenden Sie AWS CLI

1. Verwenden Sie den CLI-Befehl get-template, um die Vorlage für den Stack abzurufen, den Sie aktualisieren möchten.
2. Kopieren Sie die Vorlage, fügen Sie sie in eine Textdatei ein und ändern und speichern Sie sie. Kopieren Sie nur die Vorlage. Der Befehl umschließt die Vorlage in Anführungszeichen, kopiert aber nicht die Anführungszeichen am Anfang und Ende der Vorlage. Die Vorlage selbst beginnt mit einer offenen Klammer und endet mit der letzten engen Klammer. Geben Sie Änderungen an den Stack-Ressourcen in dieser Datei an.

Beachten Sie die folgenden Punkte, wenn Sie Änderungen an Ihrer Vorlage vornehmen:

- Sie können keinen Parameter hinzufügen, ändern oder löschen, der von einer Ressource verwendet wird, die keine Updates unterstützt.
- Für die meisten Ressourcen ist das Ändern des logischen Namens einer Ressource gleichbedeutend mit dem Löschen dieser Ressource und ersetzt sie durch eine neue. Alle anderen Ressourcen, die von der umbenannten Ressource abhängen, müssen ebenfalls aktualisiert werden und können dazu führen, dass sie ersetzt werden. Bei anderen Ressourcen müssen Sie eine Update initiieren, um eine Eigenschaft (nicht nur den logischen Namen) zu aktualisieren.
- Für einige Ressourcen gelten möglicherweise Einschränkungen hinsichtlich der Werte, die Sie für bestimmte Eigenschaften festlegen können. Beispielsweise müssen Änderungen an der `AllocatedStorage` Eigenschaft für eine RDS-Datenbankinstanz größer als der aktuelle Wert sein. Wenn Ihr Update gegen diese Regeln verstößt, schlägt dieser Teil fehl.
- Die Aktualisierung einer Ressource kann sich auch auf andere auswirken, die darauf verweisen. Wenn Sie Funktionen wie Die Funktion `Ref` oder verwendenDie Funktion `Fn::GetAtt`, um eine Eigenschaft festzulegen, die auf einer anderen Ressource basiert, CloudFormation wird auch die referenzierende Ressource aktualisiert, wenn sich die referenzierte ändert.

- Informationen zu den Auswirkungen der Aktualisierung bestimmter Ressourceneigenschaften finden Sie in der Referenz zu AWS Ressourcen- und Eigenschaftstypen. Für jede Eigenschaft werden die Auswirkungen eines Updates einer der folgenden sein:
  - Aktualisierung erfordert: Keine Unterbrechung
  - Aktualisierung erfordert: Einige Unterbrechungen
  - Aktualisierung erfordert: Austausch
- Sie können die JSON- oder YAML-Syntax Ihrer Vorlage überprüfen, indem Sie den validate-templateCLI-Befehl verwenden oder Ihre Vorlage auf der Konsole angeben. Die Konsole führt die Validierung automatisch durch. Diese Methoden überprüfen jedoch nur die Syntax Ihrer Vorlage und überprüfen nicht, ob die Eigenschaftswerte, die Sie für eine Ressource angegeben haben, für diese Ressource gültig sind. Für komplexere Validierungen oder um nach bewährten Methoden zu suchen, können Sie auch zusätzliche Tools wie CloudFormation Linter (cfn-lint) und CloudFormation Rain (rain fmt) verwenden.

> **Note**
>
> Manchmal CloudFormation werden bestimmte Änderungen, die Sie vornehmen möchten, nicht zugelassen, und es wird Ihnen mitgeteilt, dass die Änderung nicht zulässig ist. Diese Meldung kann jedoch asynchron auftreten, da Ressourcen standardmäßig CloudFormation in einer nicht deterministischen Reihenfolge erstellt und aktualisiert werden.

Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen

Wenn Sie ein Update einreichen, werden die Ressourcen auf der Grundlage von Unterschieden zwischen dem, was Sie einreichen, und der aktuellen Vorlage des Stacks AWS CloudFormation aktualisiert. Nicht geänderte Ressourcen werden während des Updates ohne Unterbrechung ausgeführt. AWS CloudFormation Verwendet für aktualisierte Ressourcen eines der folgenden Aktualisierungsverhalten:

Aktualisierung ohne Unterbrechung

AWS CloudFormation aktualisiert die Ressource, ohne den Betrieb dieser Ressource zu unterbrechen und ohne die physische ID der Ressource zu ändern. Wenn Sie beispielsweise bestimmte Eigenschaften einer AWS::CloudTrail::TrailRessource aktualisieren, wird der Pfad ohne Unterbrechung AWS CloudFormation aktualisiert.

Aktualisierungen mit einigen Unterbrechungen

AWS CloudFormation aktualisiert die Ressource mit einigen Unterbrechungen. Wenn Sie beispielsweise bestimmte Eigenschaften einer AWS::EC2::InstanceRessource aktualisieren, kann es zu einer Unterbrechung der Instance kommen, während AWS CloudFormation Amazon die Instance EC2 neu konfiguriert.

Ersatz

AWS CloudFormation erstellt die Ressource während eines Updates neu, wodurch auch eine neue physische ID generiert wird. AWS CloudFormation erstellt normalerweise zuerst die Ersatzressource, ändert Verweise von anderen abhängigen Ressourcen so, dass sie auf die Ersatzressource verweisen, und löscht dann die alte Ressource. Wenn Sie beispielsweise die `AvailabilityZone` Eigenschaft eines AWS::EC2::InstanceRessourcentyps aktualisieren, AWS CloudFormation eine neue Ressource erstellen und die aktuelle EC2 Instanzressource durch die neue ersetzen.

Wenn Sie eine Eigenschaft hinzufügen oder entfernen, die ersetzt werden muss, wird ebenfalls eine Aktualisierung ausgelöst. Die Aktualisierung passiert auch dann, wenn sich der tatsächliche Wert der Eigenschaft nicht ändert.

Die AWS CloudFormation verwendete Methode hängt davon ab, welche Eigenschaft Sie für einen bestimmten Ressourcentyp aktualisieren. Das Aktualisierungsverhalten für jede Eigenschaft wird in der Referenz zu AWS Ressourcen und Eigenschaftstypen beschrieben.

Je nach Aktualisierungsverhalten können Sie entscheiden, wann Sie Ressourcen ändern, um die Auswirkungen dieser Änderungen auf Ihre Anwendung zu verringern. Insbesondere können Sie planen, wann Ressourcen während einer Aktualisierung ersetzt werden müssen. Wenn Sie beispielsweise die `Port` Eigenschaft eines AWS: :RDS:: DBInstance -Ressourcentyps aktualisieren, AWS CloudFormation ersetzt die DB-Instance, indem eine neue DB-Instance mit der aktualisierten Porteinstellung erstellt wird, und löscht die alte DB-Instance. Bevor Sie die Aktualisierung durchführen, planen Sie möglicherweise Folgendes, um die Ersetzung der Datenbank vorzubereiten:

- Erstellen Sie einen Snapshot der aktuellen Datenbanken.
- Bereiten Sie eine Strategie dafür vor, wie Anwendungen, die diese DB-Instance verwenden, eine Unterbrechung verarbeiten, während die DB-Instance ersetzt wird.
- Stellen Sie sicher, dass die Anwendungen, die diese DB-Instance verwenden, die aktualisierte Port-Einstellung und andere Aktualisierungen, die Sie vorgenommen haben, berücksichtigen.

- Verwenden Sie den DB-Snapshot zum Wiederherstellen der Datenbanken auf der neuen DB-Instance.

Dieses Beispiel ist nicht vollständig. Es soll Ihnen jedoch eine Vorstellung davon geben, was Sie vorab berücksichtigen sollten, wenn eine Ressource während einer Aktualisierung ersetzt wird.

> **Note**
>
> Wenn die Vorlage einen oder mehrere verschachtelte Stacks enthält, initiiert sie AWS CloudFormation außerdem ein Update für jeden verschachtelten Stack. Dies ist erforderlich, um festzustellen, ob die verschachtelten Stacks geändert wurden. AWS CloudFormation aktualisiert nur die Ressourcen in den verschachtelten Stacks, deren Änderungen in den entsprechenden Vorlagen angegeben sind.

CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren

Bevor Sie einen Stack aktualisieren, ist es wichtig, zu verstehen, welche Auswirkungen die Änderungen auf die Ausführung von Ressourcen haben werden, bevor Sie diese durchführen, damit Sie die Aktualisierung sicher durchführen können. Mithilfe von Änderungssätzen können Sie in einer Vorschau anzeigen, wie sich vorgeschlagene Änderungen an einem Stack auf Ihre laufenden Ressourcen auswirken könnten, einschließlich der Auswirkungen auf die Ressourceneigenschaften und -attribute. Unabhängig davon, ob durch Ihre Änderungen wichtige Ressourcen gelöscht oder ersetzt werden, CloudFormation werden die Änderungen an Ihrem Stack erst vorgenommen, wenn Sie sich entscheiden, den Änderungssatz auszuführen. So können Sie entscheiden, ob Sie mit Ihren vorgeschlagenen Änderungen fortfahren oder andere Änderungen untersuchen möchten, indem Sie einen weiteren Änderungssatz erstellen. Sie können Änderungssätze mithilfe der CloudFormation Konsole oder der CloudFormation API erstellen und verwalten. AWS CLI

Themen

- Einen Änderungssatz für einen CloudFormation Stapel erstellen
- Einen Änderungssatz für einen CloudFormation Stack anzeigen
- Führt einen Änderungssatz für einen CloudFormation Stack aus
- Löschen Sie einen Änderungssatz für einen CloudFormation Stack
- Beispiel für Änderungssätze für CloudFormation Stapel
- Änderungssätze für verschachtelte Stacks

> **Important**
>
> Änderungssätze geben nicht an, ob ein Stack erfolgreich aktualisiert CloudFormation werden kann. Ein Änderungssatz prüft beispielsweise nicht, ob Sie ein Kontokontingent überschreiten, ob Sie eine Ressource aktualisieren, die keine Updates unterstützt, oder ob Sie nicht über ausreichende Berechtigungen zum Ändern einer Ressource verfügen. All dies kann dazu führen, dass ein Stack-Update fehlschlägt. Wenn ein Update fehlschlägt, versucht CloudFormation, ein Rollback Ihrer Ressourcen in den ursprünglichen Zustand durchzuführen.

Änderungssatz – Übersicht

Das folgende Diagramm zeigt, wie Sie mithilfe von Änderungssätzen einen Stack aktualisieren können:

1. Erstellen Sie einen Änderungssatz, indem Sie Änderungen für den Stack, den Sie aktualisieren möchten, einreichen. Sie können eine geänderte Stack-Vorlage oder geänderte Eingabeparameterwerte einreichen. CloudFormation vergleicht Ihren Stack mit den Änderungen, die Sie zur Generierung des Änderungssatzes eingereicht haben. Zu diesem Zeitpunkt werden keine Änderungen an Ihrem Stack vorgenommen.
2. Zeigen Sie den Änderungssatz an, um zu sehen, welche Stack-Einstellungen und Ressourcen geändert werden. Sie können beispielsweise sehen, welche Ressourcen hinzugefügt, geändert oder gelöscht CloudFormation werden. Darüber hinaus können Sie sich einen before-and-after Vergleich der Ressourceneigenschaften und Attribute, wie z. B. Tags, anzeigen lassen, die geändert CloudFormation werden.
3. Optional: Wenn Sie noch andere Änderungen berücksichtigen möchten, bevor Sie entscheiden, welche Änderungen vorgenommen werden sollen, erstellen Sie zusätzliche Änderungssätze. Wenn Sie mehrere Änderungssätze erstellen, können Sie besser verstehen und beurteilen, wie sich verschiedene Änderungen auf Ihre Ressourcen und Eigenschaften auswirken. Sie können so viele Änderungssätze erstellen, wie Sie benötigen.
4. Führen Sie den Änderungssatz aus, der die Änderungen enthält, die Sie auf Ihren Stack anwenden möchten. CloudFormation aktualisiert Ihren Stack mit diesen Änderungen.

> **Note**
>
> Nachdem Sie eine Änderung ausgeführt haben, werden alle Änderungssätze CloudFormation entfernt, die dem Stack zugeordnet sind, da sie für den aktualisierten Stack nicht gelten.

Sie können Änderungssätze auch löschen, um zu verhindern, dass ein Änderungssatz ausgeführt wird, der nicht angewendet werden sollte.

Einen Änderungssatz für einen CloudFormation Stapel erstellen

Um einen Änderungssatz für einen laufenden Stack zu erstellen, reichen Sie die gewünschten Änderungen ein, indem Sie eine geänderte Vorlage, neue Eingabeparameterwerte oder beides angeben. CloudFormation generiert einen Änderungssatz, indem Ihr Stack mit den von Ihnen eingereichten Änderungen verglichen wird.

Sie können eine Vorlage ändern, bevor Sie den Änderungssatz erstellen oder während Sie ihn erstellen.

Create a change set (console)

Um einen Änderungssatz zu erstellen

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie auf der Seite Stacks den laufenden Stack aus, für den Sie einen Änderungssatz erstellen möchten.
4. Wählen Sie im Bereich mit den Stack-Details die Option Stack-Aktionen und dann Change Set erstellen aus.
5. Führen Sie auf der `stack-name` Seite Änderungssatz erstellen für einen der folgenden Schritte aus, um Eingabeparameterwerte zu ändern, den Speicherort einer aktualisierten Vorlage anzugeben oder die Vorlage zu ändern:

Aufgabe	Aktion
So ändern Sie Eingabeparameterwerte	Wählen Sie „Bestehende Vorlage verwenden“ und anschließend „Weiter“, um mit der Eingabe oder Änderung von Eingabeparameterwerten fortzufahren.
So geben Sie den Speicherort einer aktualisierten Vorlage an	Wenn Sie die Vorlage geändert haben, wählen Sie Bestehende Vorlage ersetzen und führen Sie dann einen der folgenden Schritte aus: • Für eine in einem Amazon S3-Bucket gespeicherte Vorlage wählen Sie Amazon S3-URL aus. Geben Sie die URL für die Vorlage ein bzw. fügen Sie sie ein und wählen Sie dann Weiter aus. Wenn Sie eine Vorlage in einem Bucket mit aktivierter Versionierung haben, können Sie eine bestimmte Version der Vorlage angeben, indem Sie sie `?versionId=` *`version-id`* an die URL anhängen. Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem Bucket mit aktivierter Versionierung im Amazon Simple Storage Service-Benutzerhandbuch. • Für eine lokal auf Ihrem Computer gespeicherte Vorlage wählen Sie Upload a template file (Vorlagendatei hochladen) aus. Wählen Sie Choose File (Datei auswählen) aus, um zur Datei zu navigieren. Wählen Sie die Datei aus und klicken Sie auf Weiter.
So ändern Sie die Vorlage	Wenn Sie die Vorlage nicht geändert haben, wählen Sie Vorlage in Infrastructure Composer bearbeiten und dann In Infrastructure Composer bearbeiten aus. Sie werden zu AWS-Infrastruktur-Composer umgeleitet. Nachdem Sie die Vorlage geändert haben, wählen Sie Änderungssatz erstellen und dann Bestätigen und fahren fort CloudFormation, um zur *`stack-name`* Seite Änderungssatz erstellen für zurückzukehren, und wählen Sie dann Weiter.

6. Geben Sie auf der Seite „Stack-Details angeben“ einen Namen für den Änderungssatz und optional eine Beschreibung des Änderungssatzes an, um seinen Zweck im Abschnitt „Übersicht“ zu verdeutlichen. Wenn Ihre Vorlage Parameter enthält, geben Sie auf der Seite Specify stack details (Stack-Details angeben) die anwendbaren Eingabeparameterwerte ein oder ändern Sie sie und klicken Sie dann auf Weiter.

   Wenn Sie die Vorlage des Stacks wiederverwenden, CloudFormation füllt jeden Parameter mit dem aktuellen Wert im Stack auf, mit Ausnahme der mit dem Attribut deklarierten Parameter. `NoEcho` Um vorhandene Werte für diese Parameter zu verwenden, wählen Sie Use existing value (Vorhandenen Wert verwenden) aus.

   Weitere Informationen zur Maskierung vertraulicher Informationen und `NoEcho` zur Verwendung dynamischer Parameter zur Verwaltung von Geheimnissen finden Sie unter Keine Anmeldeinformationen in Vorlagen einbetten Best Practice.

7. Aktualisieren Sie auf der Seite Stack-Optionen konfigurieren die Stack-Tags, die IAM-Servicerolle, die Stack-Richtlinie, die Rollback-Konfiguration, das Amazon SNS SNS-Benachrichtigungsthema (falls zutreffend) oder die Änderungssätze.

   > **Note**
   >
   > Änderungssätze für verschachtelte Stacks sind standardmäßig aktiviert, wodurch Änderungssets für alle in Ihrer Vorlage angegebenen verschachtelten Stacks erstellt werden. Um einen Änderungssatz nur für den aktuellen Stack zu erstellen, wählen Sie Deaktiviert. Weitere Informationen zu Änderungssätzen für verschachtelte Stacks finden Sie unter. Änderungssätze für verschachtelte Stacks

8. Wenn die Vorlage IAM-Ressourcen enthält, wählen Sie für Capabilities die Option Ich bestätige, dass CloudFormation möglicherweise IAM-Ressourcen erstellt werden. IAM-Ressourcen können Berechtigungen in Ihrem AWS -Konto ändern. Überprüfen Sie diese Ressourcen, um sicherzustellen, dass Sie nur die Aktionen zulassen, die Sie zulassen möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.

9. Wählen Sie Next (Weiter), um fortzufahren.

10. Überprüfen Sie auf der *`stack-name`* Seite Überprüfen die Änderungen für diesen Änderungssatz.

11. Wählen Sie Absenden aus.

Sie werden auf die Registerkarte Changes (Änderungen) der Detailseite des Änderungssatzes umgeleitet. Während der Änderungssatz CloudFormation generiert wird, lautet der Status des Änderungssatzes`CREATE_PENDING`. Sobald der Änderungssatz erstellt wurde, ändert CloudFormation den Status in `CREATE_COMPLETE`. Im Abschnitt Änderungen werden alle Änderungen CloudFormation aufgeführt, die es an Ihrem Stack vornehmen wird. Weitere Informationen finden Sie unter Einen Änderungssatz für einen CloudFormation Stack anzeigen.

Wählen Sie in der Spalte Änderungen auf Eigenschaftsebene die Option Details anzeigen aus, um die auf Eigenschaftsebene vorgenommenen Änderungen anzuzeigen.

Falls der Änderungssatz (`FAILED`Berichtsstatus) CloudFormation nicht erstellt werden kann, korrigieren Sie den im Feld Status angezeigten Fehler und erstellen Sie den Änderungssatz anschließend neu.

12. Nachdem Sie sich vergewissert haben, dass die Änderungen korrekt aussehen, wählen Sie Änderungssatz ausführen

Create a change set for nested stacks (console)

Um einen Änderungssatz für verschachtelte Stapel zu erstellen

1. Öffnen Sie die AWS CloudFormation Konsole unter /cloudformation. https://console.aws.amazon.com
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie auf der Seite Stacks den laufenden Stack aus, für den Sie einen Änderungssatz erstellen möchten.
4. Wählen Sie im Bereich mit den Stack-Details die Option Stack-Aktionen und dann Change Set erstellen aus.
5. Führen Sie auf der *`stack-name`* Seite Änderungssatz erstellen für einen der folgenden Schritte aus, um Eingabeparameterwerte zu ändern, den Speicherort einer aktualisierten Vorlage anzugeben oder die Vorlage zu ändern:

Aufgabe	Aktion
So ändern Sie Eingabeparameterwerte	Wählen Sie „Bestehende Vorlage verwenden“ und anschließend „Weiter“, um mit der Eingabe oder Änderung von Eingabeparameterwerten fortzufahren.
So geben Sie den Speicherort einer aktualisierten Vorlage an	Wenn Sie die Vorlage geändert haben, wählen Sie Bestehende Vorlage ersetzen und führen Sie dann einen der folgenden Schritte aus: • Für eine in einem Amazon S3-Bucket gespeicherte Vorlage wählen Sie Amazon S3-URL aus. Geben Sie die URL für die Vorlage ein bzw. fügen Sie sie ein und wählen Sie dann Weiter aus. Wenn Sie eine Vorlage in einem Bucket mit aktivierter Versionierung haben, können Sie eine bestimmte Version der Vorlage angeben, indem Sie sie `?versionId=` *`version-id`* an die URL anhängen. Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem Bucket mit aktivierter Versionierung im Amazon Simple Storage Service-Benutzerhandbuch. • Für eine lokal auf Ihrem Computer gespeicherte Vorlage wählen Sie Upload a template file (Vorlagendatei hochladen) aus. Wählen Sie Choose File (Datei auswählen) aus, um zur Datei zu navigieren. Wählen Sie die Datei aus und klicken Sie auf Weiter.
So ändern Sie die Vorlage	Wenn Sie die Vorlage nicht geändert haben, wählen Sie Vorlage in Infrastructure Composer bearbeiten und dann In Infrastructure Composer bearbeiten aus. Sie werden zu AWS-Infrastruktur-Composer umgeleitet. Nachdem Sie die Vorlage geändert haben, wählen Sie Änderungssatz erstellen und dann Bestätigen und fahren fort CloudFormation, um zur *`stack-name`* Seite Änderungssatz erstellen für zurückzukehren, und wählen Sie dann Weiter.

6. Geben Sie auf der Seite „Stack-Details angeben“ einen Namen für den Änderungssatz und optional eine Beschreibung des Änderungssatzes an, um seinen Zweck im Abschnitt „Übersicht“ zu verdeutlichen. Wenn Ihre Vorlage Parameter enthält, geben Sie auf der Seite Specify stack details (Stack-Details angeben) die anwendbaren Eingabeparameterwerte ein oder ändern Sie sie und klicken Sie dann auf Weiter.

   Wenn Sie die Vorlage des Stacks wiederverwenden, CloudFormation füllt jeden Parameter mit dem aktuellen Wert im Stack auf, mit Ausnahme der mit dem Attribut deklarierten Parameter. `NoEcho` Um vorhandene Werte für diese Parameter zu verwenden, wählen Sie Use existing value (Vorhandenen Wert verwenden) aus.

   Weitere Informationen zur Maskierung vertraulicher Informationen sowie `NoEcho` zur Verwendung dynamischer Parameter zur Verwaltung von Geheimnissen finden Sie unter Keine Anmeldeinformationen in Vorlagen einbetten Best Practice.

7. Aktualisieren Sie auf der Seite Stack-Optionen konfigurieren die Stack-Tags, die IAM-Servicerolle, die Stack-Richtlinie, die Rollback-Konfiguration, das Amazon SNS SNS-Benachrichtigungsthema (falls zutreffend) oder die Änderungssätze. Weitere Informationen finden Sie unter Konfigurieren Sie die Stack-Optionen.

   > **Note**
   >
   > Änderungssätze für verschachtelte Stacks sind standardmäßig aktiviert, wodurch Änderungssets für alle in Ihrer Vorlage angegebenen verschachtelten Stacks erstellt werden. Weitere Informationen zu Änderungssätzen für verschachtelte Stacks finden Sie unter. Änderungssätze für verschachtelte Stacks

8. Wenn die Vorlage IAM-Ressourcen enthält, wählen Sie für Capabilities die Option Ich bestätige, dass CloudFormation möglicherweise IAM-Ressourcen erstellt werden. IAM-Ressourcen können Berechtigungen in Ihrem AWS -Konto ändern. Überprüfen Sie diese Ressourcen, um sicherzustellen, dass Sie nur die Aktionen zulassen, die Sie zulassen möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.

9. Wählen Sie Next (Weiter), um fortzufahren.

10. Überprüfen Sie auf der *`stack-name`* Seite Überprüfen die Änderungen für diesen Änderungssatz.

11. Wählen Sie Absenden aus.

> **Note**
>
> CloudFormation Änderungssätze auf Eigenschaftsebene lösen stapelübergreifende Verweise nicht auf, wenn Sie Änderungssätze für verschachtelte Stapel erstellen. Änderungssätze können Ressourcen in einem untergeordneten Stapel zur bedingten Ersetzung kennzeichnen, wenn sie auf die Ausgabe eines übergeordneten Stacks verweisen und der übergeordnete Stapel geändert wurde

Sie werden auf die Registerkarte Changes (Änderungen) der Detailseite des Änderungssatzes umgeleitet. Während der Änderungssatz CloudFormation generiert wird, lautet der Status des Änderungssatzes`CREATE_PENDING`. Sobald der Änderungssatz erstellt wurde, ändert CloudFormation den Status in `CREATE_COMPLETE`. Im Abschnitt Änderungen werden alle Änderungen CloudFormation aufgeführt, die es an Ihrem Stack vornehmen wird. Weitere Informationen finden Sie unter Einen Änderungssatz für einen CloudFormation Stack anzeigen.

Wenn CloudFormation der Änderungssatz nicht erstellt werden kann (meldet `FAILED` den Status), beheben Sie den im Feld Status angezeigten Fehler und erstellen Sie den Änderungssatz anschließend neu.

12. Nachdem Sie sich vergewissert haben, dass die Änderungen korrekt aussehen, wählen Sie Änderungssatz ausführen

So erstellen Sie einen Änderungssatz (AWS CLI)

- Verwenden Sie den create-change-set-Befehl.

  Sie reichen Ihre Änderungen als Befehlsoptionen ein. Sie können neue Parameterwerte, eine geänderte Vorlage oder beides festlegen. Mit dem folgenden Befehl erstellen Sie beispielsweise einen Änderungssatz mit dem Namen `SampleChangeSet` für den `MyStack`-Stack. Der Änderungssatz verwendet die aktuelle Stack-Vorlage, jedoch mit einem anderen Wert für den `Purpose`-Parameter:

```
aws cloudformation create-change-set --stack-name MyStack \
    --change-set-name SampleChangeSet --use-previous-template \
    --parameters \
```

```
    ParameterKey="InstanceType",UsePreviousValue=true
ParameterKey="KeyPairName",UsePreviousValue=true
ParameterKey="Purpose",ParameterValue="production"
```

Einen Änderungssatz für einen CloudFormation Stack anzeigen

Nach dem Erstellen eines Änderungssatzes können Sie die vorgeschlagenen Änderungen vor der Ausführung anzeigen. Sie können die CloudFormation Konsole oder die CloudFormation API verwenden AWS CLI, um Änderungssätze anzuzeigen. Die CloudFormation -Konsole enthält eine Zusammenfassung der Änderungen und eine detaillierte Liste der Änderungen im JSON-Format. Die AWS CloudFormation API AWS CLI und geben eine detaillierte Liste der Änderungen im JSON-Format zurück.

View a change set (console)

Um einen Änderungssatz anzuzeigen

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie auf der Seite Stacks den Namen des Stacks aus, der den Änderungssatz enthält, den Sie anzeigen möchten.
4. Wählen Sie im Navigationsbereich die Option Change Sets (Änderungssätze) aus, um eine Liste der Änderungssätze des Stack anzuzeigen.
5. Wählen Sie den Namen des Änderungssatzes aus, den Sie anzeigen möchten.

   Die CloudFormation Konsole leitet Sie zur Detailseite des Änderungssatzes weiter, auf der Sie den Zeitpunkt der Erstellung des Änderungssatzes, seinen Status, die zur Generierung des Änderungssatzes verwendeten Eingaben und eine Zusammenfassung der Änderungen sehen können.

   Im Abschnitt Changes (Änderungen) stellt jede Zeile eine Ressource dar, die von CloudFormation hinzugefügt, geändert oder entfernt wird.

   - Hinzufügen — CloudFormation erstellt eine Ressource, wenn Sie der Vorlage des Stacks eine Ressource hinzufügen.

- Ändern — CloudFormation ändert eine Ressource, wenn Sie die Eigenschaften einer Ressource in der Vorlage des Stacks ändern.
- Entfernen — CloudFormation löscht eine Ressource, wenn Sie eine Ressource aus der Vorlage des Stacks löschen.

> **Note**
>
> Eine Änderung kann dazu führen, dass die Ressource unterbrochen oder ersetzt (neu erstellt) werden muss. Weitere Informationen zum Verhalten von Ressourcenaktualisierungen finden Sie unter Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen.

Um bestimmte Änderungen anzuzeigen, verwenden Sie die Filteransicht. Filtern Sie beispielsweise nach einem bestimmten Ressourcentyp, z. B. `AWS::EC2::Instance`. Geben Sie zum Filtern nach einer bestimmten Ressource die logische oder physische ID an, z. B. `myWebServer` oder `i-123abcd4`.

6. Wählen Sie im Abschnitt Änderungen in der Spalte Änderungen auf Eigenschaftsebene die Option Details anzeigen aus, um die an Ihrer Ressource vorgenommenen Änderungen am Eigenschaftswert anzuzeigen.
7. Die CloudFormation Konsole leitet Sie zur Seite mit Änderungen auf Eigenschaftsebene für eine Ressource weiter, auf der Sie die Vorlagenkonfiguration der Ressource vor der Ausführung eines Änderungssatzes sehen und sehen können, wie die Vorlagenkonfiguration nach der Ausführung des Änderungssatzes aussehen wird.

   In der Tabelle mit Änderungen auf Eigenschaftsebene werden Pfad, Änderungstyp, Wert vor und Nachher für die betroffenen Eigenschaften angezeigt. Aktivieren Sie in der Tabelle das Kontrollkästchen für jede Änderung, die Sie in der Vorher - und Nachher-Ansicht Ihrer Vorlage hervorheben möchten, um zu sehen, welche Änderungen auf Eigenschaftsebene vorgenommen werden.

   - Hinzufügen — Hinzugefügte Eigenschaften sind grün hervorgehoben.
   - Ändern — Geänderte Eigenschaften sind blau markiert.
   - Entfernen — Entfernte Eigenschaften sind rot markiert.

View a change set for nested stack (console)

So zeigen Sie einen Änderungssatz für verschachtelte Stacks an (Konsole)

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie auf der Seite Stacks den Namen des Stacks aus, der den Änderungssatz enthält, den Sie anzeigen möchten.
4. Wählen Sie im Navigationsbereich Change Sets aus, um eine Liste der Änderungssets des Stacks anzuzeigen.
5. Wählen Sie den Namen des Änderungssatzes aus, den Sie anzeigen möchten.

   Die CloudFormation Konsole leitet Sie zur Detailseite des Änderungssets weiter, auf der Sie den Zeitpunkt der Erstellung des Änderungssatzes, seinen Status, die zur Generierung des Änderungssatzes verwendeten Eingaben und eine Zusammenfassung der Änderungen sehen können.

   Im Abschnitt Changes (Änderungen) steht jede Zeile für eine Ressource, die CloudFormation hinzufügen, ändern oder entfernen wird oder die den Status „Dynamisch“ anzeigen wird.

   - Hinzufügen — CloudFormation erstellt eine Ressource, wenn Sie der Vorlage des Stacks eine Ressource hinzufügen.
   - Ändern — CloudFormation ändert eine Ressource, wenn Sie die Eigenschaften einer Ressource in der Vorlage des Stacks ändern.
   - Entfernen — CloudFormation löscht eine Ressource, wenn Sie eine Ressource aus der Vorlage des Stacks löschen.
   - Dynamisch — Die genaue Aktion zur Änderung der Ressource CloudFormation kann anhand der Vorlage des verschachtelten Stacks nicht ermittelt werden.

   > **Note**
   >
   > Eine Änderung kann dazu führen, dass die Ressource unterbrochen oder ersetzt (neu erstellt) werden muss. Weitere Informationen zum Verhalten von Ressourcenaktualisierungen finden Sie unter Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen.

Um bestimmte Änderungen anzuzeigen, verwenden Sie die Filteransicht. Filtern Sie beispielsweise nach einem bestimmten Ressourcentyp, z. B. **`AWS::CloudFormation::Stack`**. Geben Sie zum Filtern nach einer bestimmten Ressource die logische oder physische ID an, z. B. **`DeadLetterQueue`** oder **`NestedStack`**.

6. Wählen Sie im Abschnitt Änderungen die Option Verschachtelten Änderungssatz anzeigen des verschachtelten Änderungssatzes, den Sie anzeigen möchten.

   Die CloudFormation Konsole leitet Sie zur Detailseite des verschachtelten Änderungssatzes weiter. Sie können Zu Root-Änderungssatz gehen wählen, um den Root-Änderungssatz anzuzeigen, oder Sie können Übergeordneten Änderungssatz anzeigen wählen, um den übergeordneten Änderungssatz anzuzeigen. Weitere Informationen finden Sie unter Änderungssätze für verschachtelte Stacks.

   > **Note**
   >
   > CloudFormation Änderungssätze auf Eigenschaftsebene lösen stapelübergreifende Verweise nicht auf, wenn Sie Änderungssätze für verschachtelte Stapel erstellen. Änderungssätze können Ressourcen in einem untergeordneten Stapel für die bedingte Ersetzung kennzeichnen, wenn sie auf die Ausgabe eines übergeordneten Stacks verweisen und der übergeordnete Stapel geändert wurde

So zeigen Sie einen Änderungssatz an (AWS CLI)

1. Um die ID des Änderungssatzes abzurufen, führen Sie den Befehl change-sets aus.

   Geben Sie den Namen des Stacks an, der den Änderungssatz enthält, den Sie anzeigen möchten, wie im folgenden Beispiel gezeigt:

   ```
   aws cloudformation list-change-sets --stack-name MyStack
   ```

   CloudFormation gibt eine Liste von Änderungssätzen zurück, die der folgenden ähnelt:

   ```
   {
       "Summaries": [
           {
   ```

```
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
            "Status": "CREATE_COMPLETE",
            "ChangeSetName": "SampleChangeSet",
            "CreationTime": "2020-11-18T20:44:05.889Z",
            "StackName": "MyStack",
            "ChangeSetId": "arn:aws:cloudformation:us-
east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000"
        },
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
            "Status": "CREATE_COMPLETE",
            "ChangeSetName": "SampleChangeSet-conditional",
            "CreationTime": "2020-11-18T21:15:56.398Z",
            "StackName": "MyStack",
            "ChangeSetId": "arn:aws:cloudformation:us-
east-1:123456789012:changeSet/SampleChangeSet-conditional/1a2345b6-0000-00a0-
a123-00abc0abc000"
        },
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
            "Status": "CREATE_COMPLETE",
            "ChangeSetName": "SampleChangeSet-replacement",
            "CreationTime": "2020-11-18T21:03:37.706Z",
            "StackName": "MyStack",
            "ChangeSetId": "arn:aws:cloudformation:us-
east-1:123456789012:changeSet/SampleChangeSet-replacement/1a2345b6-0000-00a0-
a123-00abc0abc000"
        }
    ]
}
```

2. Führen Sie den Befehl describe-change-set aus, der die ID des Änderungssatzes enthält, den Sie anzeigen möchten. Zum Beispiel:

```
aws cloudformation describe-change-set \
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

CloudFormation gibt Informationen über den angegebenen Änderungssatz zurück.

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
    "Status": "CREATE_COMPLETE",
    "ChangeSetName": "SampleChangeSet-direct",
    "Parameters": [
        {
            "ParameterValue": "testing",
            "ParameterKey": "Purpose"
        },
        {
            "ParameterValue": "ellioty-useast1",
            "ParameterKey": "KeyPairName"
        },
        {
            "ParameterValue": "t2.micro",
            "ParameterKey": "InstanceType"
        }
    ],
    "Changes": [
        {
            "ResourceChange": {
                "ResourceType": "AWS::EC2::Instance",
                "PhysicalResourceId": "i-1abc23d4",
                "Details": [
                    {
                        "ChangeSource": "DirectModification",
                        "Evaluation": "Static",
                        "Target": {
                            "Attribute": "Tags",
                            "RequiresRecreation": "Never"
                        }
                    }
                ],
                "Action": "Modify",
                "Scope": [
                    "Tags"
                ],
                "LogicalResourceId": "MyEC2Instance",
                "Replacement": "False"
            },
            "Type": "Resource"
        }
```

```
    ],
    "CreationTime": "2020-11-18T23:35:25.813Z",
    "Capabilities": [],
    "StackName": "MyStack",
    "NotificationARNs": [],
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-direct/9edde307-960d-4e6e-ad66-b09ea2f20255"
}
```

Verwenden Sie `--include-property-values` withdescribe-change-set, um die Änderungen auf Eigenschaftsebene aufzulisten.

Der `Changes`-Schlüssel listet Änderungen an Ressourcen auf. Wenn Sie diesen Änderungssatz ausführen CloudFormation würden, würden die Tags der `i-1abc23d4` EC2 Instanz aktualisiert. Eine Beschreibung der einzelnen Felder finden Sie unter dem jeweiligen Change Datentyp in der AWS CloudFormation API-Referenz.

Weitere Beispiele von Änderungssätzen finden Sie unter Beispiel für Änderungssätze für CloudFormation Stapel.

Um Änderungen auf Eigenschaftsebene in einem Änderungssatz anzuzeigen ()AWS CLI

- Der folgende Befehl listet die Änderungen auf Eigenschaftsebene auf, die sich auf einen Änderungssatz für eine `AWS::EC2::NetworkInterface` Ressource beziehen. Dadurch wird die `Ipv4Prefixes` Eigenschaft entfernt, die `Description` für die Ressource geändert und Folgendes hinzugefügt: `Tag`

```
aws cloudformation describe-change-set --include-property-values \
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841
```

Es folgt eine Beispielausgabe.

```
"ChangeSetName": "ExampleChangeSet",
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:803642222207:changeSet/
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",
    "StackId": "arn:aws:cloudformation:us-east-1:803642222207:stack/ExampleStack/
ab664180-f686-11ee-9e29-12cd92393671",
    "StackName": "ExampleStack",
    "Description": null,
```

```
    "Parameters": null,
    "CreationTime": "2024-04-09T18:04:59.935000+00:00",
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE",
    "StatusReason": null,
    "NotificationARNs": [],
    "RollbackConfiguration": {
        "RollbackTriggers": []
    },
    "Capabilities": [],
    "Tags": null,
    "ParentChangeSetId": null,
    "IncludeNestedStacks": true,
    "RootChangeSetId": null,
    "OnStackFailure": null,
{
    "Changes": [
        {
            "Type": "Resource",
            "ResourceChange": {
                "Action": "Modify",
                "LogicalResourceId":
"EC2NetworkInterface00eni067fd35b649a05b7100Tpyls",
                "PhysicalResourceId": "eni-067fd35b649a05b71",
                "ResourceType": "AWS::EC2::NetworkInterface",
                "Replacement": "False",
                "Scope": [
                    "Properties",
                    "Tags"
                ],
                "Details": [
                    {
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "Ipv4Prefixes",
                            "RequiresRecreation": "Never",
                            "Path": "/Properties/Ipv4Prefixes",
                            "BeforeValue": "[]",
                            "AttributeChangeType": "Remove"
                        },
                        "Evaluation": "Static",
                        "ChangeSource": "DirectModification"
                    },
                    {
```

```
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "Description",
                            "RequiresRecreation": "Never",
                            "Path": "/Properties/Description",
                            "BeforeValue": "",
                            "AfterValue": "Description",
                            "AttributeChangeType": "Modify"
                        },
                        "Evaluation": "Static",
                        "ChangeSource": "DirectModification"
                    },
                    {
                        "Target": {
                            "Attribute": "Tags",
                            "RequiresRecreation": "Never",
                            "Path": "/Properties/Tags/0",
                            "AfterValue": "{\"Key\":\"Test\",\"Value\":\"Test\"}",
                            "AttributeChangeType": "Add"
                        },
                        "Evaluation": "Static",
                        "ChangeSource": "DirectModification"
                    }
                ],
                "BeforeContext": "{\"Properties\":{\"Description\":\"\",
\"PrivateIpAddress\":\"172.31.76.2\",\"PrivateIpAddresses\":[{\"PrivateIpAddress
\":\"172.31.76.2\",\"Primary\":\"true\"}],\"SecondaryPrivateIpAddressCount\":\"0\",
\"Ipv6PrefixCount\":\"0\",\"Ipv4Prefixes\":[],\"Ipv4PrefixCount\":\"0\",\"GroupSet
\":[\"sg-05a45689b1059e82d\"],\"Ipv6Prefixes\":[],\"SubnetId\":\"subnet-455e8969\",
\"SourceDestCheck\":\"true\",\"InterfaceType\":\"interface\",\"Tags\":[]},
\"UpdateReplacePolicy\":\"Retain\",\"DeletionPolicy\":\"Retain\"}",
                "AfterContext": "{\"Properties\":{\"Description\":\"Description\",
\"PrivateIpAddress\":\"172.31.76.2\",\"PrivateIpAddresses\":[{\"PrivateIpAddress
\":\"172.31.76.2\",\"Primary\":\"true\"}],\"SecondaryPrivateIpAddressCount
\":\"0\",\"Ipv6PrefixCount\":\"0\",\"Ipv4PrefixCount\":\"0\",\"GroupSet\":
[\"sg-05a45689b1059e82d\"],\"Ipv6Prefixes\":[],\"SubnetId\":\"subnet-455e8969\",
\"SourceDestCheck\":\"true\",\"InterfaceType\":\"interface\",\"Tags\":[{\"Value\":
\"Test\",\"Key\":\"Test\"}]},\"UpdateReplacePolicy\":\"Retain\",\"DeletionPolicy\":
\"Retain\"}"
            }
        }
    ],
    "ChangeSetName": "ExampleChangeSet",
```

```
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/ExampleStack/
ab664180-f686-11ee-9e29-12cd92393671",
```

Führt einen Änderungssatz für einen CloudFormation Stack aus

Um die in einem Änderungssatz beschriebenen Änderungen in Ihrem Stack vorzunehmen, führen Sie den Änderungssatz aus.

> **Important**
>
> CloudFormation Löscht nach der Ausführung eines Änderungssatzes alle zusätzlichen Änderungssätze, die dem Stack zugeordnet sind, da sie für den aktualisierten Stack nicht mehr gültig sind. Wenn ein Update fehlschlägt, müssen Sie einen neuen Änderungssatz erstellen.

Stack-Richtlinien und das Ausführen eines Änderungssatzes

Wenn Sie eine Änderungssatz in einem Stack ausführen, dem eine Stack-Richtlinie zugeordnet ist, erzwingt CloudFormation die Richtlinie bei der Aktualisierung des Stack. Sie können keine temporäre Stack-Richtlinie angeben, die die vorhandene Richtlinie überschreibt, wenn Sie einen Änderungssatz ausführen. Um eine geschützte Ressource zu aktualisieren, müssen Sie die Stack-Richtlinie aktualisieren oder Methode für direktes Aktualisieren anwenden. Weitere Informationen finden Sie unter Stacks direkt aktualisieren.

Execute a change set (console)

Um einen Änderungssatz auszuführen

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie auf der Seite Stacks den Namen des Stacks aus, den Sie aktualisieren möchten.
4. Wählen Sie im Navigationsbereich Change Sets aus, um eine Liste der Änderungssets des Stacks anzuzeigen.

5. Wählen Sie den Namen des Änderungssatzes aus, den Sie anzeigen möchten.
6. Wählen Sie auf der Detailseite des Änderungssets die Option Änderungssatz ausführen aus.

   CloudFormation beginnt sofort mit der Aktualisierung des Stacks. Die CloudFormation Konsole leitet Sie zur Registerkarte Ereignisse weiter, auf der Sie den Fortschritt der Stack-Aktualisierung überwachen können. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

Execute a change set for nested stacks (console)

Um einen Änderungssatz für verschachtelte Stacks auszuführen

1. Öffnen Sie die AWS CloudFormation Konsole unter /cloudformation. https://console.aws.amazon.com
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie auf der Seite Stacks den Namen des Stacks aus, den Sie aktualisieren möchten. Sie müssen den mit dem Root-Änderungssatz verknüpften Stack-Namen auswählen.
4. Wählen Sie im Navigationsbereich Change Sets aus, um eine Liste der Änderungssets des Stacks anzuzeigen.
5. Wählen Sie den Namen des Root-Änderungssatz, den Sie ausführen möchten.
6. Wählen Sie auf der Detailseite des Änderungssets die Option Änderungssatz ausführen aus.

   > **Note**
   >
   > CloudFormation führt die in Ihrem Stamm-Änderungssatz und Ihren verschachtelten Änderungssätzen beschriebenen Änderungen aus, wenn während des Vorgangs die Option Aktiviert für Änderungssätze für verschachtelte Stacks ausgewählt wurde. Einen Änderungssatz für einen CloudFormation Stapel erstellen

   CloudFormation beginnt sofort mit der Aktualisierung des Stacks. Die CloudFormation Konsole leitet Sie zur Registerkarte Ereignisse weiter, auf der Sie den Fortschritt der Stack-Aktualisierung überwachen können. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

So führen Sie einen Änderungssatz aus (AWS CLI)

- Führen Sie den Befehl execute-change-set aus.

  Geben Sie wie im folgenden Beispiel dargestellt die Änderungssatz-ID des Änderungssatzes an, den Sie ausführen möchten:

  ```
  aws cloudformation execute-change-set \
      --change-set-name \
        arn:aws:cloudformation:us-east-1:123456789012:changeSet/
  SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
  ```

  Der Befehl im Beispiel führt einen Änderungssatz mit der ID `arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000` aus.

  Nachdem Sie den Befehl ausgeführt haben, CloudFormation beginnt die Aktualisierung des Stacks. Verwenden Sie den Befehl describe-stacks, um den Status des Stacks zu überprüfen.

Löschen Sie einen Änderungssatz für einen CloudFormation Stack

Durch das Löschen eines Änderungssatzes wird dieser aus der Liste der Änderungssätze für den Stack entfernt. Durch das Löschen eines Änderungssatzes wird verhindert, dass Sie oder ein anderer Benutzer versehentlich einen Änderungssatz ausführen, der nicht angewendet werden sollte. CloudFormation Behält alle Änderungssätze bei, bis Sie den Stapel aktualisieren, sofern Sie sie nicht löschen.

Delete a change set

So löschen Sie einen Änderungssatz (Konsole)

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie auf der Seite Stacks den Namen des Stacks aus, der den Änderungssatz enthält, den Sie löschen möchten.
4. Wählen Sie im Navigationsbereich Change Sets aus, um eine Liste der Änderungssets des Stacks anzuzeigen.

5. Wählen Sie den Namen des Änderungssatzes aus, den Sie löschen möchten.
6. Wählen Sie auf der Detailseite des Änderungssets die Option Änderungssatz löschen aus.

   CloudFormation beginnt sofort, den Änderungssatz aus der Liste der Änderungssätze des Stacks zu löschen, und Sie werden zur Seite „Stacks“ weitergeleitet.

Delete a change set for nested stacks (console)

Um einen Änderungssatz für verschachtelte Stacks zu löschen

1. Öffnen Sie die AWS CloudFormation Konsole unter /cloudformation. https://console.aws.amazon.com
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie auf der Seite Stacks den Stack-Namen aus, der dem Root-Änderungssatz zugeordnet ist.
4. Wählen Sie im Navigationsbereich Change Sets aus, um eine Liste der Change-Sets des Stacks anzuzeigen.
5. Wählen Sie den Namen des Änderungssatzes aus, den Sie löschen möchten.
6. Klicken Sie auf der Detailseite des Änderungssatzes auf Löschen. Wenn Sie Änderungssatz löschen wählen, löschen Sie die gesamte Hierarchie der verschachtelten Änderungssätze.

   > **Note**
   >
   > Der Löschvorgang für Änderungssätze für verschachtelte Stacks ist asynchron und zeigt einen `DELETE_PENDING`-Status gefolgt von einem `DELETE_IN_PROGRESS`-Status an. Nach Abschluss des Vorgangs zum Löschen von Änderungssätzen werden die Änderungssätze aus der Liste entfernt. Verschachtelte Stapel im `REVIEW_IN_PROGRESS`-Status werden auch gelöscht, wenn sie während der Erstellung des Änderungssatzes erstellt wurden.

   CloudFormation beginnt sofort damit, den Änderungssatz aus der Liste der Änderungssätze des Stapels zu löschen.

> **Note**
>
> Wenn Sie verschachtelte Stacks haben, die in einem laufenden Vorgang hängen geblieben sind, finden Sie im Abschnitt zur Fehlerbehebung in Rollback-Fehler bei verschachtelten Stacks weitere Informationen.

So löschen Sie einen Änderungssatz (AWS CLI)

- Führen Sie den Befehl delete-change-set aus und geben Sie dafür wie im folgenden Beispiel dargestellt die ID des Änderungssatzes an, den Sie löschen möchten:

```
aws cloudformation delete-change-set \
    --change-set-name \
      arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

Beispiel für Änderungssätze für CloudFormation Stapel

Dieser Abschnitt enthält Beispiele für die Änderungssätze, die für allgemeine Stack-Änderungen erstellt werden CloudFormation könnten. Sie zeigen, wie Sie eine Vorlage direkt bearbeiten, ein einzelnes Eingabeparameter modifizieren, Ressourcenantworten planen (Austausch), um zu verhindern, dass Sie Daten verlieren, die nicht gesichert wurden, oder dass Anwendungen unterbrochen werden, die in Ihrem Stapel ausgeführt werden, und wie Sie Ressourcen hinzufügen oder entfernen. Um zu veranschaulichen, wie Änderungssätze funktionieren, führen wir Sie schrittweise durch die eingereichten Änderungen und besprechen den Änderungssatz, der sich daraus ergibt. Da jedes Beispiel auf dem vorherigen Beispiel aufbaut und voraussetzt, dass Sie das vorherige Beispiel verstanden haben, empfehlen wir Ihnen, die Beispiele der Reihe nach zu lesen. Eine Beschreibung der einzelnen Felder in einem Änderungssatz finden Sie unter Ändern des Datentyps in der API-Referenz zu AWS CloudFormation .

Sie können die Konsole oder die CloudFormation DescribeChangeSetAPI-Operation verwenden AWS CLI, um die Details des Änderungssatzes anzuzeigen.

Wir haben die folgenden Änderungssätze mit folgender Beispielvorlage aus einem Stack generiert:

```
{
```

```
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "A sample EC2 instance template for testing change sets.",
  "Parameters" : {
    "Purpose" : {
      "Type" : "String",
      "Default" : "testing",
      "AllowedValues" : ["testing", "production"],
      "Description" : "The purpose of this instance."
    },
    "KeyPairName" : {
      "Type": "AWS::EC2::KeyPair::KeyName",
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the
 instance"
    },
    "InstanceType" : {
      "Type" : "String",
      "Default" : "t2.micro",
      "AllowedValues" : ["t2.micro", "t2.small", "t2.medium"],
      "Description" : "The EC2 instance type."
    }
  },
  "Resources" : {
    "MyEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyPairName" },
        "InstanceType" : { "Ref" : "InstanceType" },
        "ImageId" : "ami-8fcee4e5",
        "Tags" : [
          {
            "Key" : "Purpose",
            "Value" : { "Ref" : "Purpose" }
          }
        ]
      }
    }
  }
}
```

Direktes Bearbeiten einer Vorlage

Wenn Sie Ressourcen direkt in der Stack-Vorlage ändern, um einen Änderungssatz zu erstellen, klassifiziert CloudFormation die Änderung als direkte Änderung und nicht als Änderungen, die durch

einen aktualisierten Parameterwert ausgelöst wurden. Der folgende Änderungssatz, durch den der `i-1abc23d4` Instance ein neuer Tag hinzugefügt wurde, ist ein Beispiel für eine direkte Änderung. Alle anderen Eingabewerte, wie die Parameterwerte und Fähigkeiten, bleiben unverändert, sodass wir uns auf die `Changes`-Struktur konzentrieren.

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
    "Status": "CREATE_COMPLETE",
    "ChangeSetName": "SampleChangeSet-direct",
    "Parameters": [
        {
            "ParameterValue": "testing",
            "ParameterKey": "Purpose"
        },
        {
            "ParameterValue": "MyKeyName",
            "ParameterKey": "KeyPairName"
        },
        {
            "ParameterValue": "t2.micro",
            "ParameterKey": "InstanceType"
        }
    ],
    "Changes": [
        {
            "ResourceChange": {
                "ResourceType": "AWS::EC2::Instance",
                "PhysicalResourceId": "i-1abc23d4",
                "Details": [
                    {
                        "ChangeSource": "DirectModification",
                        "Evaluation": "Static",
                        "Target": {
                            "Attribute": "Tags",
                            "RequiresRecreation": "Never"
                        }
                    }
                ],
                "Action": "Modify",
                "Scope": [
                    "Tags"
                ],
```

```
                "LogicalResourceId": "MyEC2Instance",
                "Replacement": "False"
            },
            "Type": "Resource"
        }
    ],
    "CreationTime": "2020-11-18T23:35:25.813Z",
    "Capabilities": [],
    "StackName": "MyStack",
    "NotificationARNs": [],
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-direct/1a2345b6-0000-00a0-a123-00abc0abc000"
}
```

In der `Changes`-Struktur gibt es nur eine `ResourceChange`-Struktur. Diese Struktur beschreibt Informationen wie die Art der Ressource, die geändert CloudFormation wird, die Aktion, die ausgeführt CloudFormation wird, die ID der Ressource, den Umfang der Änderung und ob die Änderung ersetzt werden muss (wobei eine neue Ressource CloudFormation erstellt und dann die alte gelöscht wird). In diesem Beispiel gibt der Änderungssatz an, dass das `Tags` Attribut der `i-1abc23d4` EC2 Instanz geändert CloudFormation wird und dass die Instanz nicht ersetzt werden muss.

CloudFormation Kennzeichnet diese Änderung in der `Details` Struktur als direkte Änderung, bei der die Instanz niemals neu erstellt (ersetzt) werden muss. Sie können diese Änderung getrost ausführen, da Sie wissen, dass dadurch die CloudFormation Instanz nicht ersetzt wird.

CloudFormation zeigt diese Änderung als `Static` Bewertung an. Eine statische Auswertung bedeutet, dass der Wert des Tags bestimmt werden CloudFormation kann, bevor der Änderungssatz ausgeführt wird. In einigen Fällen CloudFormation kann ein Wert erst bestimmt werden, nachdem Sie einen Änderungssatz ausgeführt haben. CloudFormationkennzeichnet diese Änderungen als `Dynamic` Bewertungen. Wenn Sie beispielsweise auf eine aktualisierte Ressource verweisen, die unter bestimmten Bedingungen ersetzt wurde, CloudFormation können Sie nicht feststellen, ob sich der Verweis auf die aktualisierte Ressource ändern wird.

Einen Eingabeparameterwert ändern

Wenn Sie einen Eingabeparameterwert ändern, CloudFormation generiert zwei Änderungen für jede Ressource, die den aktualisierten Parameterwert verwendet. In diesem Beispiel heben wir hervor, wie diese Änderungen aussehen und auf welche Informationen Sie sich konzentrieren sollten. Das folgende Beispiel wurde nur durch Änderung des Werts des `Purpose`-Eingabeparameters erzeugt.

Der `Purpose` Parameter gibt einen Tag-Schlüsselwert für die EC2 Instanz an. In diesem Beispiel wurde der Parameterwert von `testing` in `production` geändert. Der neue Wert wird in der `Parameters`-Struktur angezeigt.

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
    "Status": "CREATE_COMPLETE",
    "ChangeSetName": "SampleChangeSet",
    "Parameters": [
        {
            "ParameterValue": "production",
            "ParameterKey": "Purpose"
        },
        {
            "ParameterValue": "MyKeyName",
            "ParameterKey": "KeyPairName"
        },
        {
            "ParameterValue": "t2.micro",
            "ParameterKey": "InstanceType"
        }
    ],
    "Changes": [
        {
            "ResourceChange": {
                "ResourceType": "AWS::EC2::Instance",
                "PhysicalResourceId": "i-1abc23d4",
                "Details": [
                    {
                        "ChangeSource": "DirectModification",
                        "Evaluation": "Dynamic",
                        "Target": {
                            "Attribute": "Tags",
                            "RequiresRecreation": "Never"
                        }
                    },
                    {
                        "CausingEntity": "Purpose",
                        "ChangeSource": "ParameterReference",
                        "Evaluation": "Static",
                        "Target": {
                            "Attribute": "Tags",
```

```
                            "RequiresRecreation": "Never"
                        }
                    }
                ],
                "Action": "Modify",
                "Scope": [
                    "Tags"
                ],
                "LogicalResourceId": "MyEC2Instance",
                "Replacement": "False"
            },
            "Type": "Resource"
        }
    ],
    "CreationTime": "2020-11-18T23:59:18.447Z",
    "Capabilities": [],
    "StackName": "MyStack",
    "NotificationARNs": [],
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000"
}
```

Die `Changes` Struktur funktioniert ähnlich wie im Direktes Bearbeiten einer Vorlage Beispiel. Es gibt nur eine `ResourceChange` Struktur; sie beschreibt eine Änderung des `Tags` Attributs der `i-1abc23d4` EC2 Instanz.

Doch in der `Details`-Struktur zeigt der Änderungssatz zwei Änderungen für das `Tags`-Attribut, auch wenn nur ein einzelner Parameterwert geändert wurde. Ressourcen, die auf einen geänderten Parameterwert verweisen (mithilfe der intrinsischen `Ref`-Funktion) führen immer zu zwei Änderungen: eine mit einer `Dynamic`-Bewertung und eine andere mit einer `Static`-Bewertung. Sie können diese Arten von Änderungen sehen, indem Sie die folgenden Felder anzeigen:

- Die Änderung der `Static`-Bewertung sehen Sie im Feld `ChangeSource`. In diesem Beispiel entspricht das `ChangeSource`-Feld der `ParameterReference`. Das bedeutet, dass diese Änderung das Ergebnis eines aktualisierten Parameterreferenzwerts ist. Der Änderungssatz muss eine ähnliche `Dynamic`-Bewertungsänderung enthalten.
- Sie finden die passende `Dynamic`-Bewertung, indem Sie die `Target`-Struktur für beide Änderungen vergleichen, die dieselben Informationen enthalten wird. In diesem Beispiel enthalten die `Target`-Strukturen für beide Änderungen dieselben Werte für die Felder `Attribute` und `RequireRecreation`.

Konzentrieren Sie sich für diese Art von Änderungen auf die statische Bewertung, die Ihnen die genauesten Informationen zur Änderung liefert. In diesem Beispiel zeigt die statische Bewertung, dass die Änderung das Ergebnis der Änderung eines Parameterreferenzwerts (`ParameterReference`) ist. Der exakte Parameter, der geändert wurde, wird im Feld `CauseEntity` (der `Purpose`-Parameter) angezeigt.

Ermitteln des Werts des Ersatzfelds

Das `Replacement` Feld in einer `ResourceChange` Struktur gibt an, ob CloudFormation die Ressource neu erstellt werden soll. Die Planung einer Ressourcen-Neuerstellung (Ersatz) verhindert den Verlust von Daten, die nicht gesichert wurden, oder die Unterbrechung von Anwendungen, die in Ihrem Stack ausgeführt werden.

Der Wert im Feld `Replacement` hängt davon ab, ob eine Änderung einen Ersatz erfordert. Dies wird durch das Feld `RequiresRecreation` in der `Target`-Struktur einer Änderung angezeigt. Wenn beispielsweise der Wert des `RequiresRecreation`-Felds `Never` ist, wird der Wert des `Replacement`-Felds `False`. Wenn es jedoch mehrere Änderungen an einer einzelnen Ressource gibt und jede Änderung einen anderen Wert für das `RequiresRecreation` Feld hat, wird die Ressource CloudFormation aktualisiert, wobei das aufdringlichste Verhalten angewendet wird. Mit anderen Worten: Wenn nur eine der vielen Änderungen einen Ersatz erfordert, muss CloudFormation die Ressource ersetzen und setzt das `Replacement`-Feld daher auf `True`.

Der folgende Änderungssatz wurde generiert, indem die Werte für jeden Parameter (`PurposeInstanceType`, und`KeyPairName`) geändert wurden, die alle von der EC2 Instanz verwendet werden. Aufgrund dieser Änderungen muss die Instanz ersetzt CloudFormation werden, da das `Replacement` Feld gleich ist`True`.

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
    "Status": "CREATE_COMPLETE",
    "ChangeSetName": "SampleChangeSet-multiple",
    "Parameters": [
        {
            "ParameterValue": "production",
            "ParameterKey": "Purpose"
        },
        {
            "ParameterValue": "MyNewKeyName",
            "ParameterKey": "KeyPairName"
        },
```

```
        {
            "ParameterValue": "t2.small",
            "ParameterKey": "InstanceType"
        }
    ],
    "Changes": [
        {
            "ResourceChange": {
                "ResourceType": "AWS::EC2::Instance",
                "PhysicalResourceId": "i-7bef86f8",
                "Details": [
                    {
                        "ChangeSource": "DirectModification",
                        "Evaluation": "Dynamic",
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "KeyName",
                            "RequiresRecreation": "Always"
                        }
                    },
                    {
                        "ChangeSource": "DirectModification",
                        "Evaluation": "Dynamic",
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "InstanceType",
                            "RequiresRecreation": "Conditionally"
                        }
                    },
                    {
                        "ChangeSource": "DirectModification",
                        "Evaluation": "Dynamic",
                        "Target": {
                            "Attribute": "Tags",
                            "RequiresRecreation": "Never"
                        }
                    },
                    {
                        "CausingEntity": "KeyPairName",
                        "ChangeSource": "ParameterReference",
                        "Evaluation": "Static",
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "KeyName",
```

```
                            "RequiresRecreation": "Always"
                        }
                    },
                    {
                        "CausingEntity": "InstanceType",
                        "ChangeSource": "ParameterReference",
                        "Evaluation": "Static",
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "InstanceType",
                            "RequiresRecreation": "Conditionally"
                        }
                    },
                    {
                        "CausingEntity": "Purpose",
                        "ChangeSource": "ParameterReference",
                        "Evaluation": "Static",
                        "Target": {
                            "Attribute": "Tags",
                            "RequiresRecreation": "Never"
                        }
                    }
                ],
                "Action": "Modify",
                "Scope": [
                    "Tags",
                    "Properties"
                ],
                "LogicalResourceId": "MyEC2Instance",
                "Replacement": "True"
            },
            "Type": "Resource"
        }
    ],
    "CreationTime": "2020-11-18T00:39:35.974Z",
    "Capabilities": [],
    "StackName": "MyStack",
    "NotificationARNs": [],
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-multiple/1a2345b6-0000-00a0-a123-00abc0abc000"
}
```

Identifizieren Sie die Änderung, durch welche die Ressource ersetzt werden muss, indem Sie jede einzelne Änderung (die statischen Bewertungen in der `Details`-Struktur) anzeigen. In diesem Beispiel hat jede Änderung einen anderen Wert für das `RequireRecreation` Feld, aber die Änderung an der `KeyName` Eigenschaft hat das aufdringlichste Aktualisierungsverhalten und erfordert immer eine Neuerstellung. CloudFormation ersetzt die Instanz, weil der Schlüsselname geändert wurde.

Wäre der Schlüsselname unverändert geblieben, hätte die Änderung an der Eigenschaft `InstanceType` das Aktualisierungsverhalten mit den größten Auswirkungen gehabt (`Conditionally`), und das Feld `Replacement` wäre `Conditionally`. Die Bedingungen, unter denen die Instanz CloudFormation ersetzt wird, finden Sie im Aktualisierungsverhalten für die `InstanceType` Eigenschaft des AWS::EC2::InstanceRessourcentyps.

Hinzufügen und Entfernen von Ressourcen

Das folgende Beispiel wurde generiert, indem eine geänderte Vorlage eingereicht wurde, die die EC2 Instanz entfernt und eine Auto Scaling Scaling-Gruppe sowie eine Startkonfiguration hinzufügt.

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
    "Status": "CREATE_COMPLETE",
    "ChangeSetName": "SampleChangeSet-addremove",
    "Parameters": [
        {
            "ParameterValue": "testing",
            "ParameterKey": "Purpose"
        },
        {
            "ParameterValue": "MyKeyName",
            "ParameterKey": "KeyPairName"
        },
        {
            "ParameterValue": "t2.micro",
            "ParameterKey": "InstanceType"
        }
    ],
    "Changes": [
        {
            "ResourceChange": {
                "Action": "Add",
                "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
```

```
                "Scope": [],
                "Details": [],
                "LogicalResourceId": "AutoScalingGroup"
            },
            "Type": "Resource"
        },
        {
            "ResourceChange": {
                "Action": "Add",
                "ResourceType": "AWS::AutoScaling::LaunchConfiguration",
                "Scope": [],
                "Details": [],
                "LogicalResourceId": "LaunchConfig"
            },
            "Type": "Resource"
        },
        {
            "ResourceChange": {
                "ResourceType": "AWS::EC2::Instance",
                "PhysicalResourceId": "i-1abc23d4",
                "Details": [],
                "Action": "Remove",
                "Scope": [],
                "LogicalResourceId": "MyEC2Instance"
            },
            "Type": "Resource"
        }
    ],
    "CreationTime": "2020-11-18T01:44:08.444Z",
    "Capabilities": [],
    "StackName": "MyStack",
    "NotificationARNs": [],
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-addremove/1a2345b6-0000-00a0-a123-00abc0abc000"
}
```

In der Struktur `Changes` gibt es drei `ResourceChange`-Strukturen, eine für jede Ressource. Für jede Ressource gibt das `Action` Feld an, ob die Ressource CloudFormation hinzugefügt oder entfernt wird. Die Felder `Scope` und `Details` sind leer, da sie nur für geänderte Ressourcen gelten.

Bei neuen Ressourcen CloudFormation kann der Wert einiger Felder erst bestimmt werden, wenn Sie den Änderungssatz ausgeführt haben. Stellt beispielsweise CloudFormation nicht die physischen

Daten IDs der Auto Scaling Scaling-Gruppe und der Startkonfiguration bereit, da sie noch nicht existieren. CloudFormation erstellt die neuen Ressourcen, wenn Sie den Änderungssatz ausführen.

Änderungen auf Eigenschaftsebene anzeigen

Das folgende Beispiel zeigt Änderungen auf Eigenschaftsebene an der `Tag` Eigenschaft einer EC2 Amazon-Instance. Das Tag `Value` und `Key` wird geändert zu. `Test`

```
"ChangeSetName": "SampleChangeSet",
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/38d91d27-798d-4736-9bf1-fb7c46207807",
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
SampleEc2Template/68edcdc0-f6b6-11ee-966c-126d572cdd11",
    "StackName": "SampleEc2Template",
    "Description": "A sample EC2 instance template for testing change sets.",
    "Parameters": [
        {
            "ParameterKey": "KeyPairName",
            "ParameterValue": "BatchTest"
        },
        {
            "ParameterKey": "Purpose",
            "ParameterValue": "testing"
        },
        {
            "ParameterKey": "InstanceType",
            "ParameterValue": "t2.micro"
        }
    ],
    "CreationTime": "2024-04-09T21:29:10.759000+00:00",
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE",
    "StatusReason": null,
    "NotificationARNs": [],
    "RollbackConfiguration": {
:...skipping...
{
    "Changes": [
        {
            "Type": "Resource",
            "ResourceChange": {
                "Action": "Modify",
                "LogicalResourceId": "MyEC2Instance",
```

```
                "PhysicalResourceId": "i-0cc7856a36315e62b",
                "ResourceType": "AWS::EC2::Instance",
                "Replacement": "False",
                "Scope": [
                    "Tags"
                ],
                "Details": [
                    {
                        "Target": {
                            "Attribute": "Tags",
                            "RequiresRecreation": "Never",
                            "Path": "/Properties/Tags/0/Value",
                            "BeforeValue": "testing",
                            "AfterValue": "Test",
                            "AttributeChangeType": "Modify"
                        },
                        "Evaluation": "Static",
                        "ChangeSource": "DirectModification"
                    },
                    {
                        "Target": {
                            "Attribute": "Tags",
                            "RequiresRecreation": "Never",
                            "Path": "/Properties/Tags/0/Key",
                            "BeforeValue": "Purpose",
                            "AfterValue": "Test",
                            "AttributeChangeType": "Modify"
                        },
                        "Evaluation": "Static",
                        "ChangeSource": "DirectModification"
                    }
                ],
                "BeforeContext": "{\"Properties\":{\"KeyName\":\"BatchTest\",\"ImageId
\":\"ami-8fcee4e5\",\"InstanceType\":\"t2.micro\",\"Tags\":[{\"Value\":\"testing\",
\"Key\":\"Purpose\"}]}}",
                "AfterContext": "{\"Properties\":{\"KeyName\":\"BatchTest\",\"ImageId
\":\"ami-8fcee4e5\",\"InstanceType\":\"t2.micro\",\"Tags\":[{\"Value\":\"Test\",\"Key
\":\"Test\"}]}}"
            }
        }
    ]
```

Die `Details` Struktur zeigt die Werte für `Key` und `Value` vor der Ausführung des Änderungssatzes und ihre Werte nach der Ausführung des Änderungssatzes.

Änderungssätze für verschachtelte Stacks

Mit Änderungssätzen für verschachtelte Stacks können Sie eine Vorschau der Änderungen an Ihren Anwendungs- und Infrastrukturressourcen in der gesamten verschachtelten Stack-Hierarchie anzeigen und mit Aktualisierungen fortfahren, wenn Sie bestätigt haben, dass alle Änderungen wie beabsichtigt sind.

In den folgenden Abschnitten finden Sie weitere Details zu Änderungssätzen für verschachtelte Stacks:

Themen

- Überblick über Änderungssätze und verschachtelte Stacks
- Arbeiten mit Änderungssätzen für verschachtelte Stacks (Konsole)
- Arbeiten mit Änderungssätzen für verschachtelte Stacks (AWS CLI)

Überblick über Änderungssätze und verschachtelte Stacks

Änderungssätze für verschachtelte Stacks kombinieren die folgenden Funktionen, um den Umfang der Vorschau von Änderungen in der gesamten Stack-Hierarchie zu erweitern:

- Ein Änderungssatz ist eine CloudFormation Funktion, die eine Vorschau darauf bietet, wie sich vorgeschlagene Änderungen an einem Stack auf bestehende oder neu erstellte Ressourcen auswirken werden. CloudFormation Stellt nach der Erstellung eines Änderungssatzes eine Liste der vorgeschlagenen Änderungen bereit, indem Ihr Stack mit den Änderungen an den von Ihnen eingereichten Ressourcen verglichen wird. Weitere Informationen zu Änderungssätzen finden Sie unter CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren.
- Ein verschachtelter Stapel ist ein Stapel, der als Teil eines anderen Stacks erstellt wurde. Sie können z. B. sicherheitsbezogene Ressourcen in einem verschachtelten Stack haben und Anwendungs-Ressourcen wie Datenbanken in anderen. Die Partitionierung von Anwendungsmodellen hilft auf diese Weise bei der Wartbarkeit und Wiederverwendung von Code. Weitere Hinweise zu verschachtelten Stacks finden Sie unter. Teilen Sie eine Vorlage mithilfe verschachtelter Stapel in wiederverwendbare Teile auf

Arbeiten mit Änderungssätzen für verschachtelte Stacks (Konsole)

- Änderungssatz erstellen – Erstellt einen Änderungssatz durch Übermitteln von Änderungen von jeder Ebene der Stack-Hierarchie. Sie können eine geänderte Stack-Vorlage oder geänderte Eingabeparameterwerte übermitteln und CloudFormation vergleicht Ihren verschachtelten Stapel mit den Änderungen, die Sie zur Erstellung eines Änderungssatzes übermittelt haben. Änderungssätze für verschachtelte Stacks sind in der CloudFormation -Konsole standardmäßig aktiviert. Weitere Informationen finden Sie unter Einen Änderungssatz für einen CloudFormation Stapel erstellen.

> **Note**
>
> Ein Root-Änderungssatz ist der Änderungssatz, der dem Stack zugeordnet ist, aus dem die gesamte Hierarchie der Änderungssätze erstellt wird. Sie müssen Änderungssätze für verschachtelte Stacks aus dem Root-Änderungssatz ausführen oder löschen. Weitere Informationen finden Sie unter Ausführen von Stack-Operationen auf verschachtelten Stacks.

- Änderungssatz anzeigen – Visualisieren Sie Änderungen an Ressourcen in verschachtelten Stacks, bevor Sie sie ausführen. Sie können die vorgeschlagenen Änderungen im Abschnitt Änderungen Ihres Änderungssatzes anzeigen, indem Sie durch den aktuellen Stack und seine verschachtelten Änderungssätze navigieren. Weitere Informationen finden Sie unter Einen Änderungssatz für einen CloudFormation Stack anzeigen.
- Änderungssatz ausführen –Führen Sie die im Änderungssatz beschriebenen Änderungen aus, die sich auf den aktuellen Stack und seine Nachfolger beziehen. Der Ausführungsvorgang muss aus dem Root-Änderungssatz erfolgen. Weitere Informationen finden Sie unter Führt einen Änderungssatz für einen CloudFormation Stack aus.
- Änderungssatz löschen – Entfernt die Änderungssätze aus dem aktuellen Stack. Das Löschen eines Änderungssatzes verhindert, dass Sie oder ein anderer Benutzer versehentlich einen Änderungssatz initiieren, der nicht angewendet werden sollte. Der Löschvorgang muss aus dem Root-Änderungssatz erfolgen. Weitere Informationen finden Sie unter Löschen Sie einen Änderungssatz für einen CloudFormation Stack.

Arbeiten mit Änderungssätzen für verschachtelte Stacks (AWS CLI)

create-change-set

- create-change-set – Änderungssätze für verschachtelte Stacks sind nicht standardmäßig für die AWS CLI aktiviert. Um einen Änderungssatz für die gesamte Stack-Hierarchie zu erstellen, geben Sie die `--include-nested-stacks` Option an. Weitere Informationen finden Sie unter Einen Änderungssatz für einen CloudFormation Stapel erstellen.

Im folgenden AWS CLI Beispiel wird ein Änderungssatz für den angegebenen Root-Stack erstellt.

```
aws cloudformation create-change-set \
    --stack-name my-root-stack \
    --change-set-name my-root-stack-change-set \
    --template-body file://template.yaml \
    --capabilities CAPABILITY_IAM \
    --include-nested-stacks
```

Es folgt eine Beispielausgabe.

```
{
    "Id":"arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:Stack/my-root-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

describe-change-set

- describe-change-set— Gibt eine Liste der Änderungen zurück, die vorgenommen CloudFormation werden, wenn Sie den Änderungssatz ausführen. Wenn der angegebene Änderungssatz untergeordnete Änderungssätze enthält, die zu verschachtelten Stacks gehören, gibt `ChangeSetId` Informationen über diesen Änderungssatz zurück. Weitere Informationen finden Sie unter Einen Änderungssatz für einen CloudFormation Stack anzeigen.

Das folgende AWS CLI Beispiel beschreibt den Änderungssatz für den angegebenen Root-Stack.

```
aws cloudformation describe-change-set \
    --change-set-name my-root-stack-change-set \
```

```
    --stack-name my-root-stack
```

Es folgt eine Beispielausgabe.

```
{
    "Changes": [
        {
            "Type": "Resource",
            "ResourceChange": {
                "Action": "Modify",
                "LogicalResourceId": "ChildStack",
                "PhysicalResourceId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/my-nested-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99205",
                "ResourceType": "AWS::CloudFormation::Stack",
                "Replacement": "False",
                "ChangeSetId": "arn:aws:cloudformation:us-
west-2:123456789012:changeSet/my-nested-stack-change-set/4eca1a01-e285-
xmpl-8026-9a1967bfb4b0",
                "Scope": [
                    "Properties"
                ],
                "Details": [
                    {
                        "Target": {
                            "Attribute": "Properties",
                            "RequiresRecreation": "Never"
                        },
                        "Evaluation": "Dynamic",
                        "ChangeSource": "Automatic"
                    }
                ]
            }
        }
    ],
    "ChangeSetName": "my-root-stack-change-set",
    "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-
stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-root-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "StackName": "my-root-stack",
    "IncludeNestedStacks": true,
    "ParentChangeSetId": null,
    "RootChangeSetId": null,
```

```
    "Description": null,
    "Parameters": null,
    "CreationTime": "2020-11-18T05:20:56.651Z",
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE",
    "StatusReason": null,
    "NotificationARNs": [

    ],
    "RollbackConfiguration": {

    },
    "Capabilities": [
        "CAPABILITY_IAM"
    ],
    "Tags": null
}
```

Das folgende AWS CLI Beispiel beschreibt den Änderungssatz für den angegebenen verschachtelten Stack.

```
aws cloudformation describe-change-set \
    --change-set-name my-nested-stack-change-set \
    --stack-name my-nested-stack
```

Es folgt eine Beispielausgabe.

```
{
    "Changes": [
        {
            "Type": "Resource",
            "ResourceChange": {
                "Action": "Modify",
                "LogicalResourceId": "function",
                "PhysicalResourceId": "my-function",
                "ResourceType": "AWS::Lambda::Function",
                "Replacement": "False",
                "ChangeSetId": null,
                "Scope": [
                    "Properties"
                ],
                "Details": [
                    {
```

```
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "Timeout",
                            "RequiresRecreation": "Never"
                        },
                        "Evaluation": "Static",
                        "ChangeSource": "DirectModification"
                    }
                ]
            }
        }
    ],
    "ChangeSetName": "my-nested-stack-change-set",
    "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-nested-
stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-nested-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99205",
    "ParentChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
    "RootChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
    "IncludeNestedStacks": true,
    "StackName": "my-nested-stack",
    "Description": null,
    "Parameters": null,
    "CreationTime": "2020-11-18T05:20:56.651Z",
    "ExecutionStatus": "UNAVAILABLE",
    "Status": "CREATE_COMPLETE",
    "StatusReason": "Executable from root change set",
    "NotificationARNs": [

    ],
    "RollbackConfiguration": {

    },
    "Capabilities": [
        "CAPABILITY_IAM"
    ],
    "Tags": null
}
```

execute-change-set

- execute-change-set – Erstellt oder akktualisiert einen Stack mit den Eingabedaten, die beim Erstellen des angegebenen Änderungssatzes bereitgestellt wurden. Um einen Änderungssatz für die gesamte Stack-Hierarchie zu erstellen, müssen Sie die `--include-nested-stacks` Option während des create-change-set Vorgangs angeben. Weitere Informationen finden Sie unter Führt einen Änderungssatz für einen CloudFormation Stack aus.

> **Note**
>
> execute-change-set muss aus dem Root-Änderungssatz ausgeführt werden und wendet den Änderungssatz auf die gesamte Hierarchie der Stacks an.

Im folgenden AWS CLI Beispiel wird ein Änderungssatz für den angegebenen Root-Stack ausgeführt.

```
aws cloudformation execute-change-set \
    --stack-name my-root-stack \
    --change-set-name my-root-stack-change-set
```

delete-change-set

- delete-change-set – Löscht den angegebenen Änderungssatz. Das Löschen von Änderungssätzen stellt sicher, dass niemand den falschen Änderungssatz verwendet. Das Löschen von Änderungssätzen erfolgt asynchron für Änderungssätze, die mit der `--include-nested-stacks` Option erstellt wurden. Weitere Informationen finden Sie unter Löschen Sie einen Änderungssatz für einen CloudFormation Stack.

> **Note**
>
> delete-change-set muss aus dem Root-Änderungssatz ausgeführt werden und löscht die gesamte Hierarchie der Änderungssätze. Verschachtelte Stacks im Status `REVIEW_IN_PROGRESS` werden auch gelöscht, wenn sie während dem create-change-set-Vorgang erstellt wurden.

Im folgenden AWS CLI Beispiel wird der Änderungssatz für den angegebenen Root-Stack gelöscht.

```
aws cloudformation delete-change-set \
```

```
    --stack-name my-root-stack \
    --change-set-name my-root-stack-change-set
```

Stacks direkt aktualisieren

Wenn Sie schnell Aktualisierungen an Ihrem Stack bereitstellen möchten, führen Sie eine direkte Aktualisierung aus. Bei einer direkten Aktualisierung reichen Sie eine Vorlage oder Eingabeparameter ein, die Aktualisierungen der Ressourcen im Stack angeben, und stellen diese CloudFormation sofort bereit. Wenn Sie eine Vorlage für Ihre Aktualisierungen verwenden möchten, können Sie die aktuelle Vorlage ändern und sie lokal oder in einem Amazon S3 S3-Bucket speichern.

Für Ressourceneigenschaften, die Aktualisierungen nicht unterstützen, müssen Sie die aktuellen Werte beibehalten. Verwenden Sie Änderungssets, um eine Vorschau der Änderungen, die an Ihrem Stack vorgenommen CloudFormation werden, anzuzeigen, bevor Sie ihn aktualisieren. Weitere Informationen finden Sie unter CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren.

Je nachdem, welche Eigenschaften Sie aktualisieren, CloudFormation können beim Aktualisieren eines Stacks Ressourcen unterbrochen oder aktualisierte Ressourcen ersetzt werden. Weitere Informationen zum Verhalten von Ressourcenaktualisierungen finden Sie unter Verstehen Sie das Aktualisierungsverhalten von Stack-Ressourcen.

So aktualisieren Sie einen -Stack (Konsole)

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie auf der Seite Stacks den laufenden Stack aus, den Sie aktualisieren möchten.
4. Klicken Sie im Detailbereich des Stacks auf Update (Aktualisieren).
5. Wenn Sie die Stack-Vorlage nicht geändert haben, wählen Sie Bestehende Vorlage verwenden und dann Weiter aus.

   Wenn Sie die Vorlage geändert haben, wählen Sie Bestehende Vorlage ersetzen aus und geben Sie den Speicherort der aktualisierten Vorlage im Abschnitt Vorlage angeben an:

   - Für eine lokal auf Ihrem Computer gespeicherte Vorlage wählen Sie Upload a template file (Vorlagendatei hochladen) aus. Wählen Sie Choose File (Datei auswählen) aus, um zur Datei zu navigieren. Wählen Sie die Datei aus und wählen Sie Next (Weiter) aus.

> **Note**
>
> Wenn Sie eine lokale Vorlagendatei CloudFormation hochladen, lädt sie in einen Amazon Simple Storage Service (Amazon S3) -Bucket in Ihrem AWS-Konto. Wenn Sie noch keinen S3-Bucket haben, der von erstellt wurde CloudFormation, erstellt er einen eigenen Bucket für jede Region, in die Sie eine Vorlagendatei hochladen. Wenn Sie bereits einen S3-Bucket haben, der von CloudFormation in Ihrem erstellt wurde AWS-Konto, CloudFormation fügt die Vorlage diesem Bucket hinzu.
> Überlegungen, die Sie bei S3-Buckets beachten sollten, die von erstellt wurden CloudFormation
>
> - Auf die Buckets kann jeder zugreifen, der über Amazon S3 S3-Berechtigungen in Ihrem AWS-Konto Unternehmen verfügt.
> - CloudFormation erstellt die Buckets mit standardmäßig aktivierter serverseitiger Verschlüsselung, wodurch alle im Bucket gespeicherten Objekte verschlüsselt werden.
>
> Sie können die Verschlüsselungsoptionen für Buckets, die erstellt CloudFormation wurden, direkt verwalten, z. B. mithilfe der Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/oder der AWS CLI. Weitere Informationen finden Sie unter Einstellung des standardmäßigen serverseitigen Verschlüsselungsverhaltens für Amazon S3 S3-Buckets im Amazon Simple Storage Service-Benutzerhandbuch.
> - Sie können Ihren eigenen Bucket verwenden und dessen Berechtigungen durch manuelles Hochladen von Vorlagen in Amazon S3 verwalten. Geben Sie die Amazon S3-URL einer Vorlagendatei an, wenn Sie einen Stack erstellen oder aktualisieren.

- Für eine in einem Amazon S3-Bucket gespeicherte Vorlage wählen Sie Amazon S3-URL aus. Geben Sie die URL für die Vorlage ein bzw. fügen Sie sie ein und wählen Sie dann Weiter aus.

  Wenn Sie eine Vorlage in einem Bucket mit aktivierter Versionierung haben, können Sie eine bestimmte Version der Vorlage angeben, indem Sie sie an die URL anhängen? `versionId=`*`version-id`*. Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem Bucket mit aktivierter Versionierung im Amazon Simple Storage Service-Benutzerhandbuch.

Wenn Syntaxprobleme festgestellt werden, zeigt die Konsole Fehlermeldungen an, die Ihnen helfen, die Vorlage zu korrigieren.

6. Wenn Ihre Vorlage Parameter enthält, geben Sie auf der Seite Specify stack details (Stack-Details angeben) die Parameterwerte an oder ändern Sie sie und wählen Sie dann Next (Weiter) aus.

   CloudFormation füllt jeden Parameter mit dem Wert auf, der aktuell im Stack festgelegt ist, mit Ausnahme der mit dem `NoEcho` Attribut deklarierten Parameter. Sie können jedoch weiterhin aktuelle Werte verwenden, indem Sie die Option Vorhandenen Wert verwenden aktivieren.

   Weitere Informationen `NoEcho` zum Maskieren vertraulicher Informationen sowie zur Verwendung dynamischer Parameter zur Verwaltung von Geheimnissen finden Sie unter Keine Anmeldeinformationen in Vorlagen einbetten Best Practice.

7. Auf der Seite Configure stack options (Stack-Optionen konfigurieren) können Sie die Tags und Berechtigungen aktualisieren, die auf den Stack angewendet werden, sowie erweiterte Optionen wie z. B. Stack-Richtlinie, Rollback-Konfiguration ändern oder das Amazon-SNS-Benachrichtigungsthema aktualisieren. Weitere Informationen zu diesen Optionen finden Sie unter Konfigurieren Sie die Stack-Optionen.

8. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie unter Capabilities die Option Ich bestätige, dass diese Vorlage IAM-Ressourcen erstellen kann, um anzugeben, dass Sie IAM-Ressourcen in der Vorlage verwenden möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.

9. Wählen Sie Next (Weiter), um fortzufahren.

10. Überprüfen Sie die Stack-Informationen und die Änderungen, die Sie übermittelt haben.

    Überprüfen Sie, ob Sie die richtigen Informationen gesendet haben, z. B. die richtigen Parameterwerte oder die richtige Vorlagen-URL.

    Vergewissern Sie sich im Bereich „Vorschau des Änderungssatzes“, dass alle erwarteten Änderungen vorgenommen CloudFormation werden. Sie können beispielsweise überprüfen, ob die CloudFormation Ressourcen, die Sie hinzufügen, entfernen oder ändern wollten, hinzugefügt, entfernt und geändert werden. CloudFormation generiert diese Vorschau, indem ein Änderungssatz für den Stapel erstellt wird. Weitere Informationen finden Sie unter CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren.

11. Wenn Sie Ihre Änderungen vorgenommen haben, wählen Sie Update stack (Stack aktualisieren) aus.

> **Note**
>
> An dieser Stelle haben Sie auch die Möglichkeit, den Änderungssatz anzusehen, um Ihre vorgeschlagenen Updates sorgfältiger zu überprüfen. Wählen Sie dazu „Änderungssatz anzeigen“ statt „Stapel aktualisieren“. CloudFormation zeigt den Änderungssatz an, der auf der Grundlage Ihrer Aktualisierungen generiert wurde. Wenn Sie bereit sind, den Stack zu aktualisieren, wählen Sie Execute (Ausführen) aus.

CloudFormation zeigt die Seite mit den Stack-Details für Ihren Stack an, wobei der Bereich Ereignisse ausgewählt ist. Ihr Stack hat jetzt den Status`UPDATE_IN_PROGRESS`. Nachdem CloudFormation die Aktualisierung des Stacks erfolgreich abgeschlossen wurde, wird der Stack-Status auf gesetzt`UPDATE_COMPLETE`.

Schlägt die Stack-Aktualisierung fehl CloudFormation, werden die Änderungen automatisch rückgängig gemacht und der Stack-Status wird auf gesetzt`UPDATE_ROLLBACK_COMPLETE`.

> **Note**
>
> Sie können ein Update stornieren, solange es sich im `UPDATE_IN_PROGRESS` Status befindet. Weitere Informationen finden Sie unter Ein Stack-Update abbrechen.

Um einen Stack über die Befehlszeile zu aktualisieren

Verwenden Sie einen der folgenden Befehle:

- update-stack (AWS CLI)
- Update-CFNStack (AWS Tools for Windows PowerShell)

Beispiele für die Verwendung der Befehlszeile zum Aktualisieren eines Stacks finden Sie unterBeispiele für CloudFormation Stack-Operationsbefehle für AWS CLI und PowerShell.

Ein Stack-Update abbrechen

Nachdem eine Stack-Aktualisierung begonnen hat, können Sie die Stack-Aktualisierung stornieren, wenn der Stack den Status `UPDATE_IN_PROGRESS` aufweist. Nach Abschluss einer Aktualisierung können Sie diese nicht mehr stornieren. Sie können einen Stack jedoch mit früheren Einstellungen erneut aktualisieren.

Wenn Sie eine Stack-Aktualisierung stornieren, wird der Stack auf die Stack-Konfiguration vor Beginn der Stack-Aktualisierung zurückgesetzt.

Themen

- Um ein Stack-Update abzubrechen (Konsole)
- Um ein Stack-Update abzubrechen (AWS CLI)

Um ein Stack-Update abzubrechen (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm aus, AWS-Region wo sich der Stapel befindet.
3. Wählen Sie auf der Seite Stacks den Stack aus, der gerade aktualisiert wird. Sein Status muss sein. `UPDATE_IN_PROGRESS`
4. Wählen Sie Stack-Aktionen und anschließend Cancel update stack (Aktualisierung des Stacks abbrechen) aus.
5. Um mit der Stornierung der Aktualisierung fortzufahren, wählen Sie Cancel update (Aktualisierung abbrechen) aus. Wählen Sie Cancel (Abbrechen) aus, um die Aktualisierung fortzusetzen.

Der Stack wird in den Status `UPDATE_ROLLBACK_IN_PROGRESS` versetzt. Nach Abschluss der Stornierung wird der Stack auf `UPDATE_ROLLBACK_COMPLETE` gesetzt.

Um ein Stack-Update abzubrechen (AWS CLI)

Verwenden Sie den Befehl cancel-update-stack, um eine Aktualisierung zu stornieren. Weitere Informationen finden Sie unter Brechen Sie ein Stack-Update ab.

Löschen Sie einen Stack aus der CloudFormation Konsole

Wenn Sie die Ressourcen in einem Stack nicht mehr benötigen, können Sie den gesamten Stack löschen.

CloudFormation Löscht beim Löschen eines Stacks alle Ressourcen in diesem Stapel, es sei denn, Sie haben eine Löschrichtlinie verwendet, um bestimmte Ressourcen beizubehalten. Weitere Informationen finden Sie unter DeletionPolicy-Attribut.

Um einen Stapel zu löschen (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm aus, AWS-Region wo sich der Stapel befindet.
3. Wählen Sie auf der Seite Stapel den Stapel aus, den Sie löschen möchten. Der Stack muss aktuell ausgeführt werden.
4. Wählen Sie Delete (Löschen).
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).

> **Note**
>
> Der Stack-Löschvorgang kann nicht gestoppt werden, sobald die Stack-Löschung begonnen hat. Der Stack wird in den Status `DELETE_IN_PROGRESS` versetzt. Nachdem die Löschung des Stapels abgeschlossen ist, befindet sich der Stack im Status `DELETE_COMPLETE`. Stapel im `DELETE_COMPLETE` Status werden standardmäßig nicht in der CloudFormation Konsole angezeigt. Um gelöschte Stacks anzuzeigen, müssen Sie den Stack-Anzeigefilter ändern, wie unter Gelöschte Stacks von der CloudFormation Konsole aus anzeigen beschrieben.

Um das Löschen eines Stacks zu erzwingen (Konsole)

Das Löschen eines Stacks kann fehlschlagen, weil eine Ressource im Stapel nicht gelöscht werden kann. Zum Beispiel schlägt CloudFormation das Löschen einer Ressource fehl, von der auch ein anderer Stapel abhängt. Alle Ressourcen, die nicht gelöscht worden sind bleiben erhalten, bis Sie den Stack erfolgreich löschen können. Wenn das Löschen fehlschlägt und ein `DELETE_FAILED` Status zurückgegeben wird, können Sie es mit einer von zwei Methoden erneut versuchen.

1. Wählen Sie auf der Seite Stacks in der CloudFormation Konsole den Stack aus, dessen Löschung Sie erzwingen möchten.
2. Wählen Sie im Bereich mit den Stack-Details die Option Löschen erneut versuchen aus.
3. Wählen Sie zwischen den folgenden Optionen:
   - Diesen Stapel löschen, aber Ressourcen beibehalten: Mit dieser Option können Sie die spezifischen Ressourcen auswählen, die ursprünglich nicht gelöscht werden konnten, die Sie aber während des erzwungenen Löschvorgangs des Stacks beibehalten möchten.
   - Löschen des gesamten Stacks erzwingen: Diese Option behält alle Ressourcen bei, die nicht gelöscht werden konnten, und behält die Abhängigkeiten dieser Ressourcen bei.
4. Wählen Sie Löschen, um den Vorgang zum erzwungenen Löschen mit Ihrer Auswahl zu starten.

Um die einbehaltenen Ressourcen zu überprüfen (Konsole)

Nach dem Löschen des Stacks können Sie die Ressourcen, die aufbewahrt wurden, in der Konsole anzeigen.

1. Wählen Sie in der Stack-Liste den Filterstatus und dann Gelöscht aus.
2. Wählen Sie den gelöschten Stapel aus.
3. Wählen Sie die Registerkarte Resources (Ressourcen) aus.
4. Für alle einbehaltenen Ressourcen wird der `DELETE_SKIPPED` Status angezeigt.
5. Wählen Sie die einbehaltene Ressource aus, die Sie überprüfen möchten.

Um einen Stapel über die Befehlszeile zu löschen

Verwenden Sie einen der folgenden Befehle:

- delete-stack (AWS CLI)
- Remove-CFNStack (AWS Tools for Windows PowerShell)

Beispiele für die Verwendung der Befehlszeile zum Löschen eines Stacks finden Sie unterBeispiele für CloudFormation Stack-Operationsbefehle für AWS CLI und PowerShell.

Zugehörige Ressourcen

Hilfe zur Behebung von Fehlern beim Löschen von Stacks finden Sie im Thema Stack-Fehler löschen Problembehandlung.

Informationen zum Schutz von Stacks vor versehentlichem Löschen finden Sie unter CloudFormation Stapel vor dem Löschen schützen.

Gelöschte Stacks von der CloudFormation Konsole aus anzeigen

Standardmäßig zeigt die CloudFormation Konsole keine Stapel mit dem Status. `DELETE_COMPLETE` Um Informationen zu gelöschten Stacks anzuzeigen, müssen Sie die Stack-Anzeigeeinstellung ändern.

So zeigen Sie gelöschte Stacks an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm aus, AWS-Region wo sich der gelöschte Stapel befindet.
3. Wählen Sie auf der Seite „Stacks“ im Drop-down-Menü „Filterstatus“ die Option Gelöscht aus.

CloudFormation listet alle Ihre Stapel mit dem Status auf. `DELETE_COMPLETE`

Weitere Informationen finden Sie auch unter

- Löschen Sie einen Stack aus der CloudFormation Konsole
- Stack-Informationen von der CloudFormation Konsole aus anzeigen

Stack-Fortschritt überwachen

In diesem Abschnitt wird beschrieben, wie Sie eine Stack-Bereitstellung überwachen, die gerade ausgeführt wird. CloudFormation bietet eine detaillierte, chronologische Liste der Bereitstellungsereignisse, in der der Fortschritt und alle während der Bereitstellung aufgetretenen Probleme aufgeführt sind.

Themen

- CloudFormation Stack-Ereignisse anzeigen

- Sehen Sie sich einen Zeitplan für eine CloudFormation Stack-Bereitstellung an
- Ereignisse bei der CloudFormation Stackerstellung verstehen
- Überwachen Sie den Fortschritt eines Stack-Updates
- Rollback einer Aktualisierung fortsetzen
- Ermitteln Sie die Ursache eines Stack-Fehlers
- Wählen Sie aus, wie mit Ausfällen bei der Bereitstellung von Ressourcen umgegangen werden soll

CloudFormation Stack-Ereignisse anzeigen

Sie können Stack-Ereignisse anzeigen, um den Fortschritt und Status Ihres Stacks und der Ressourcen im Stack zu überwachen. Mithilfe von Stack-Ereignissen können Sie nachvollziehen, wann Ressourcen erstellt, aktualisiert oder gelöscht werden und ob die Stack-Bereitstellung wie erwartet verläuft.

Themen

- Stack-Ereignisse anzeigen (Konsole)
- Stack-Ereignisse anzeigen (AWS CLI)
- Stack-Statuscodes

Stack-Ereignisse anzeigen (Konsole)

Um Stack-Ereignisse anzuzeigen

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die Datei aus, in der AWS-Region Sie den Stapel erstellt haben.
3. Wählen Sie auf der Seite Stacks der CloudFormation Konsole den Stack-Namen aus. CloudFormation zeigt die Stack-Details für den ausgewählten Stack an.
4. Wählen Sie die Registerkarte Ereignisse, um die Stack-Ereignisse CloudFormation anzuzeigen, die für Ihren Stack generiert wurden.

CloudFormation aktualisiert die Stack-Ereignisse automatisch jede Minute. CloudFormation Zeigt außerdem das Badge Neue Ereignisse verfügbar an, wenn neue Stack-Ereignisse auftreten. Wählen

Sie das Aktualisierungssymbol, um diese Ereignisse in die Liste zu laden. Verfolgen Sie die Stack-Erstellungsereignisse, um die Abfolge der Ereignisse nachzuvollziehen, die zur Erstellung des Stacks (oder zu einem Fehler im Fall von Stack-Debugging) geführt haben.

Während der Erstellung Ihres Stacks wird er auf der Seite Stacks mit dem Status „Stacks“ aufgeführt. `CREATE_IN_PROGRESS` Nachdem Ihr Stack erfolgreich erstellt wurde, ändert sich sein Status auf`CREATE_COMPLETE`.

Weitere Informationen erhalten Sie unter Ereignisse bei der CloudFormation Stackerstellung verstehen und Überwachen Sie den Fortschritt eines Stack-Updates.

Stack-Ereignisse anzeigen (AWS CLI)

Alternativ können Sie den describe-stack-eventsBefehl verwenden, während der Stack erstellt wird, um Ereignisse so anzuzeigen, wie sie gemeldet wurden.

Der folgende describe-stack-events Befehl beschreibt die *`my-stack`* Stack-Ereignisse.

```
aws cloudformation describe-stack-events --stack-name my-stack
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "StackEvents": [
        {
            "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
            "EventId": "7b755820-7edf-11f0-ab15-0673b09f3847",
            "StackName": "my-stack",
            "LogicalResourceId": "my-stack",
            "PhysicalResourceId": "arn:aws:cloudformation:aws-
region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2025-08-21T22:37:56.243000+00:00",
            "ResourceStatus": "CREATE_COMPLETE",
            "ClientRequestToken": "token"
        },
        {
            "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
            "EventId": "WebServer-CREATE_COMPLETE-2025-08-21T22:37:54.356Z",
            "StackName": "my-stack",
```

```
            "LogicalResourceId": "WebServer",
            "PhysicalResourceId": "i-099df76cb31b866a9",
            "ResourceType": "AWS::EC2::Instance",
            "Timestamp": "2025-08-21T22:37:54.356000+00:00",
            "ResourceStatus": "CREATE_COMPLETE",
            "ResourceProperties": "{\"UserData\":
\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\",\"ImageId\":\"ami-0bbc328167dee8f3c\",\"InstanceType\":\"t2.micro\",
\"SecurityGroupIds\":[\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\"],\"Tags\":
[{\"Value\":\"Bootstrap Tutorial Web Server\",\"Key\":\"Name\"}]}",
            "ClientRequestToken": "token"
        },
        {
            "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
            "EventId": "WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:31.226Z",
            "StackName": "my-stack",
            "LogicalResourceId": "WebServer",
            "PhysicalResourceId": "i-099df76cb31b866a9",
            "ResourceType": "AWS::EC2::Instance",
            "Timestamp": "2025-08-21T22:37:31.226000+00:00",
            "ResourceStatus": "CREATE_IN_PROGRESS",
            "ResourceStatusReason": "Resource creation Initiated",
            "ResourceProperties": "{\"UserData\":
\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\",\"ImageId\":\"ami-0bbc328167dee8f3c\",\"InstanceType\":\"t2.micro\",
\"SecurityGroupIds\":[\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\"],\"Tags\":
[{\"Value\":\"Bootstrap Tutorial Web Server\",\"Key\":\"Name\"}]}",
            "ClientRequestToken": "token"
        },
        {
            "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
            "EventId": "WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:29.210Z",
            "StackName": "my-stack",
            "LogicalResourceId": "WebServer",
            "PhysicalResourceId": "",
            "ResourceType": "AWS::EC2::Instance",
            "Timestamp": "2025-08-21T22:37:29.210000+00:00",
            "ResourceStatus": "CREATE_IN_PROGRESS",
            "ResourceProperties": "{\"UserData\":
\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\",\"ImageId\":\"ami-0bbc328167dee8f3c\",\"InstanceType\":\"t2.micro\",
```

```
\"SecurityGroupIds\":[\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\"],\"Tags\":
[{\"Value\":\"Bootstrap Tutorial Web Server\",\"Key\":\"Name\"}]}",
            "ClientRequestToken": "token"
        },
        {
            "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
            "EventId": "WebServerSecurityGroup-
CREATE_COMPLETE-2025-08-21T22:37:28.803Z",
            "StackName": "my-stack",
            "LogicalResourceId": "WebServerSecurityGroup",
            "PhysicalResourceId": "my-stack-WebServerSecurityGroup-n8A43bQT1ty2",
            "ResourceType": "AWS::EC2::SecurityGroup",
            "Timestamp": "2025-08-21T22:37:28.803000+00:00",
            "ResourceStatus": "CREATE_COMPLETE",
            "ResourceProperties": "{\"GroupDescription\":\"Allow HTTP access from my IP
 address\",\"SecurityGroupIngress\":[{\"CidrIp\":\"0.0.0.0/0\",\"Description\":\"HTTP
\",\"FromPort\":\"80\",\"ToPort\":\"80\",\"IpProtocol\":\"tcp\"}]}",
            "ClientRequestToken": "token"
        },
        {
            "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
            "EventId": "WebServerSecurityGroup-
CREATE_IN_PROGRESS-2025-08-21T22:37:22.626Z",
            "StackName": "my-stack",
            "LogicalResourceId": "WebServerSecurityGroup",
            "PhysicalResourceId": "my-stack-WebServerSecurityGroup-n8A43bQT1ty2",
            "ResourceType": "AWS::EC2::SecurityGroup",
            "Timestamp": "2025-08-21T22:37:22.626000+00:00",
            "ResourceStatus": "CREATE_IN_PROGRESS",
            "ResourceStatusReason": "Resource creation Initiated",
            "ResourceProperties": "{\"GroupDescription\":\"Allow HTTP access from my IP
 address\",\"SecurityGroupIngress\":[{\"CidrIp\":\"0.0.0.0/0\",\"Description\":\"HTTP
\",\"FromPort\":\"80\",\"ToPort\":\"80\",\"IpProtocol\":\"tcp\"}]}",
            "ClientRequestToken": "token"
        },
        {
            "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
            "EventId": "WebServerSecurityGroup-
CREATE_IN_PROGRESS-2025-08-21T22:37:20.186Z",
            "StackName": "my-stack",
            "LogicalResourceId": "WebServerSecurityGroup",
```

```
            "PhysicalResourceId": "",
            "ResourceType": "AWS::EC2::SecurityGroup",
            "Timestamp": "2025-08-21T22:37:20.186000+00:00",
            "ResourceStatus": "CREATE_IN_PROGRESS",
            "ResourceProperties": "{\"GroupDescription\":\"Allow HTTP access from my IP
 address\",\"SecurityGroupIngress\":[{\"CidrIp\":\"0.0.0.0/0\",\"Description\":\"HTTP
\",\"FromPort\":\"80\",\"ToPort\":\"80\",\"IpProtocol\":\"tcp\"}]}",
            "ClientRequestToken": "token"
        },
        {
            "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
            "EventId": "64740fe0-7edf-11f0-8a36-06453a64f325",
            "StackName": "my-stack",
            "LogicalResourceId": "my-stack",
            "PhysicalResourceId": "arn:aws:cloudformation:aws-
region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2025-08-21T22:37:17.819000+00:00",
            "ResourceStatus": "CREATE_IN_PROGRESS",
            "ResourceStatusReason": "User Initiated",
            "ClientRequestToken": "token"
        }
    ]
}
```

Die neuesten Ereignisse werden zuerst ausgegeben. Die folgende Tabelle enthält eine Beschreibung der Felder, die vom Befehl describe-stack-events zurückgegeben werden:

Feld	Beschreibung
`EventId`	Ereignis-Kennung.
`StackName`	Name des Stacks, zu dem das Ereignis gehört.
`StackId`	Kennung des Stacks, zu dem das Ereignis gehört.
`LogicalResourceId`	Logische Kennung der Ressource.
`PhysicalResourceId`	Physische Kennung der Ressource.
`ResourceProperties`	Eigenschaften der Ressource.

Feld	Beschreibung
`ResourceType`	Typ der Ressource.
`Timestamp`	Uhrzeit, zu der das Ereignis aufgetreten ist.
`ResourceStatus`	Der Status der Ressource, der einer der folgenden Statuscodes sein kann: `CREATE_COMPLETE` | | `CREATE_FAILED` | `CREATE_IN_PROGRESS` | `DELETE_COMPLETE` | `DELETE_FAILED` | `DELETE_IN_PROGRESS` | `DELETE_SKIPPED` | `IMPORT_COMPLETE` | `IMPORT_IN_PROGRESS` | `IMPORT_ROLLBACK_COMPLETE` | `IMPORT_ROLLBACK_FAILED` | `IMPORT_ROLLBACK_IN_PROGRESS` | `REVIEW_IN_PROGRESS` | `ROLLBACK_COMPLETE` | `ROLLBACK_FAILED` | `ROLLBACK_IN_PROGRESS` | `UPDATE_COMPLETE` | `UPDATE_COMPLETE_CLEANUP_IN_PROGRESS` | `UPDATE_FAILED` | `UPDATE_IN_PROGRESS` | `UPDATE_ROLLBACK_COMPLETE` | `UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS` | `UPDATE_ROLLBACK_FAILED` | `UPDATE_ROLLBACK_IN_PROGRESS` Der Status `DELETE_SKIPPED` gilt für Ressourcen, deren Richtlinienattribut zum Löschen "retain" (beibehalten) lautet.
`DetailedStatus`	Der detaillierte Status des Stacks. Wenn vorhanden, `CONFIGURATION_COMPLETE` ist die Konfigurationsphase der Stack-Ressourcen abgeschlossen und die Stabilisierung der Ressourcen ist im Gange.
`ResourceStatusReason`	Weitere Informationen zum Status.

Stack-Statuscodes

Die folgende Tabelle beschreibt die Stack-Statuscodes:

Stack-Status und optionaler detaillierter Status	Beschreibung
`CREATE_COMPLETE`	Erfolgreiche Erstellung eines oder mehrerer Stacks.
`CREATE_IN_PROGRESS`	Laufende Erstellung eines oder mehrerer Stacks.
`CREATE_FAILED`	Fehlgeschlagene Erstellung eines oder mehrerer Stacks. Untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen. Mögliche Gründe für eine fehlgesch lagene Erstellung sind unzureichende Berechtigungen für die Arbeit mit allen Ressourcen im Stack, von einem AWS Dienst abgelehnte Parameterwerte oder ein Timeout bei der Ressourcenerstellung.
`DELETE_COMPLETE`	Erfolgreiche Löschung eines oder mehrerer Stacks. Gelöschte Stacks bleiben 90 Tage lang erhalten und anzeigbar.
`DELETE_FAILED`	Fehlgeschlagene Löschung eines oder mehrerer Stacks. Da der Löschvorgang fehlgeschlagen ist, werden möglicherweise einige Ressourcen noch ausgeführt. Sie können jedoch nicht mit dem Stack arbeiten oder ihn aktualisieren. Löschen Sie den Stack erneut oder untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen.
`DELETE_IN_PROGRESS`	Laufende Entfernung eines oder mehrerer Stacks.
`REVIEW_IN_PROGRESS`	Laufende Erstellung eines oder mehrerer Stacks mit einer erwarteten `StackId`, aber ohne Vorlagen oder Ressourcen. **Important** Ein Stack mit diesem Statuscode wird auf die maximal mögliche Anzahl von Stacks angerechnet.
`ROLLBACK_COMPLETE`	Erfolgreiche Entfernung eines oder mehrerer Stacks nach einer fehlgeschlagenen Stack-Erstellung oder nach einer explizit abgebrochenen Stack-Erstellung. Der Stack wird in

Stack-Status und optionaler detaillierter Status	Beschreibung
	einen früheren funktionsfähigen Zustand zurückgesetzt. Alle Ressourcen, die während des Create Stack-Vorgangs erstellt wurden, werden gelöscht. Dieser Status existiert nur, nachdem das Erstellen eines Stacks fehlgeschlagen ist. Es bedeutet, dass alle Vorgänge aus dem teilweise erstellten Stack entsprechend bereinigt wurden. In diesem Status kann nur einen Löschbefehl ausgeführt werden.
`ROLLBACK_FAILED`	Fehlgeschlagene Entfernung eines oder mehrerer Stacks nach einer fehlgeschlagenen Stack-Erstellung oder nach einer explizit abgebrochenen Stack-Erstellung. Löschen Sie den Stack oder untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen.
`ROLLBACK_IN_PROGRESS`	Laufende Entfernung eines oder mehrerer Stacks nach einer fehlgeschlagenen Stack-Erstellung oder nach einer explizit abgebrochenen Stack-Erstellung.
`UPDATE_COMPLETE`	Erfolgreiche Aktualisierung eines oder mehrerer Stacks.
`UPDATE_COMPLETE_CLEANUP_IN_PROGRESS`	Laufende Entfernung alter Ressourcen für einen oder mehrere Stacks nach einem erfolgreichen Stack-Update. Bei Stack-Updates, bei denen Ressourcen ersetzt werden müssen, CloudFormation erstellt zuerst die neuen Ressourcen und löscht dann die alten Ressourcen, um Unterbrechungen im Stack zu vermeiden. In diesem Zustand wurde der Stack aktualisiert und ist nutzbar, löscht aber CloudFormation immer noch die alten Ressourcen.
`UPDATE_FAILED`	Nicht erfolgreiche Aktualisierung eines oder mehrerer Stacks. Untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen.
`UPDATE_IN_PROGRESS`	Laufende Aktualisierung eines oder mehrerer Stacks.

Stack-Status und optionaler detaillierter Status	Beschreibung
`UPDATE_ROLLBACK_COMPLETE`	Erfolgreiches Zurücksetzen eines oder mehrerer Stacks in einen früheren funktionsfähigen Zustand nach einer fehlgeschlagenen Stack-Aktualisierung.
`UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS`	Laufende Entfernung neuer Ressourcen für einen oder mehrere Stacks nach einem fehlgeschlagenen Stack-Update. In diesem Zustand wurde der Stack in seinen vorherigen Betriebszustand zurückgesetzt und ist nutzbar, löscht aber CloudFormation immer noch alle neuen Ressourcen, die er während der Stack-Aktualisierung erstellt hat.
`UPDATE_ROLLBACK_FAILED`	Fehlgeschlagenes Zurücksetzen eines oder mehrerer Stacks in einen früheren funktionsfähigen Zustand nach einer fehlgeschlagenen Stack-Aktualisierung. In diesem Status können Sie den Stack löschen oder das Rollback fortsetzen. Möglicherweise müssen Sie Fehler beseitigen, um den Stack wieder in einen arbeitsfähigen Zustand zu versetzen. Oder Sie können Kontakt aufnehmen, Support um den Stack wieder in einen verwendbaren Zustand zu versetzen.
`UPDATE_ROLLBACK_IN_PROGRESS`	Laufendes Zurücksetzen eines oder mehrerer Stacks in den früheren funktionsfähigen Zustand nach einer fehlgeschlagenen Stack-Aktualisierung.
`IMPORT_IN_PROGRESS`	Der Importvorgang wird derzeit ausgeführt.
`IMPORT_COMPLETE`	Der Importvorgang wurde für alle Ressourcen im Stack erfolgreich abgeschlossen, die `resource import` unterstützen.
`IMPORT_ROLLBACK_IN_PROGRESS`	Der Import wird zur vorherigen Vorlagenkonfiguration zurückgesetzt.

Stack-Status und optionaler detaillierter Status	Beschreibung
`IMPORT_ROLLBACK_FAILED`	Der Import-Rollback-Vorgang ist für mindestens eine Ressource im Stack fehlgeschlagen. Die Ergebnisse werden für die CloudFormation erfolgreich importierten Ressourcen verfügbar sein.
`IMPORT_ROLLBACK_COMPLETE`	Der Import wurde erfolgreich auf die vorherige Vorlagenkonfiguration zurückgesetzt.

Sehen Sie sich einen Zeitplan für eine CloudFormation Stack-Bereitstellung an

Das Diagramm mit dem Zeitplan für die Stack-Bereitstellung bietet eine visuelle Darstellung eines Zeitplans für die Stack-Bereitstellung. In dieser Ansicht werden der Bereitstellungsstatus für den Stack und jede seiner Ressourcen sowie die Zeiten angezeigt, zu denen sich die einzelnen Status geändert haben. Der Status der Stack-Bereitstellung wird durch eine entsprechende Farbe dargestellt.

Themen

- Erläuterung des Zeitplans für die Stack-Bereitstellung
- Das Diagramm mit der Zeitleiste für die Stack-Bereitstellung anzeigen (Konsole)

Erläuterung des Zeitplans für die Stack-Bereitstellung

Die folgende Abbildung zeigt das Zeitdiagramm für eine Stack-Bereitstellung, die aufgrund einer fehlgeschlagenen EC2 Amazon-Instance-Ressource fehlgeschlagen ist.

Die Namen der Stack-Ressourcen befinden sich auf der linken Seite des Diagramms, und Datum und Uhrzeit im Verhältnis zu den Bereitstellungszeiten befinden sich oben im Diagramm.

Jede Ressource beginnt mit dem Status In Bearbeitung. Die Statusleiste ändert sich bei jeder erfolgreichen Bereitstellung zu Abgeschlossen. Die Statusleiste ändert sich in Fehlgeschlagen, wenn eine Ressource nicht bereitgestellt werden kann. Wenn eine Ressource nicht bereitgestellt werden

kann und auch die Stack-Bereitstellung fehlschlägt, erhält die Ressource, die für den Fehler bei der Stack-Bereitstellung verantwortlich ist, den Status Likely root failure.

Nachdem der Stack-Bereitstellungsvorgang fehlgeschlagen ist, beginnt die erfolgreich bereitgestellte Ressource mit dem Rollback und wechselt in den Status Rollback in progress. Der Status ändert sich in Rollback abgeschlossen, nachdem das Rollback der Ressource abgeschlossen ist.

Durch die Auswahl der einzelnen Ressourcen erhalten Sie detailliertere Informationen zum Bereitstellungszeitplan:

Bei der Auswahl einer Ressource werden der Typ, die Startzeit der Bereitstellung, die Endzeit der Bereitstellung und die Gesamtdauer der Bereitstellung angezeigt. In den Drop-down-Menüs unten finden Sie außerdem die Startzeit, die Endzeit und die Dauer jedes Bereitstellungsstatus. Wenn die Ressource nicht bereitgestellt werden konnte, wird ein Fehlergrund angegeben.

Weitere Informationen zu Stack-Status finden Sie unterStack-Statuscodes.

Das Diagramm mit der Zeitleiste für die Stack-Bereitstellung anzeigen (Konsole)

So zeigen Sie ein Diagramm mit der Zeitleiste für die Stack-Bereitstellung an:

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die Datei aus, in der AWS-Region Sie den Stapel erstellt haben.
3. Wählen Sie auf der Stacks-Seite der CloudFormation Konsole den Stack-Namen aus. CloudFormation zeigt die Stack-Details für den ausgewählten Stack an.
4. Wählen Sie die Registerkarte Ereignisse, um die Stack-Ereignisse CloudFormation anzuzeigen, die für Ihren Stack generiert wurden.
5. Wählen Sie die Schaltfläche „Timeline-Diagramm“, um das Timeline-Diagramm für Ihren Stack anzuzeigen.

Ereignisse bei der CloudFormation Stackerstellung verstehen

Während der Stack-Bereitstellung treten mehrere Ereignisse auf, um die in der Stack-Vorlage definierten Ressourcen zu erstellen, zu konfigurieren und zu validieren. Wenn Sie diese

Ereignisse verstehen, können Sie Ihren Stack-Erstellungsprozess optimieren und Bereitstellungen rationalisieren.

- Ereignisse bei der Erstellung von Ressourcen — Wenn jede Ressource den Erstellungsprozess startet, wird der Status eines `CREATE_IN_PROGRESS` Ereignisses festgelegt. Dieses Ereignis weist darauf hin, dass die Ressource bereitgestellt wird.
- Eventuelle Konsistenzprüfung — Ein erheblicher Teil der Zeit für die Erstellung eines Stacks wird für die Durchführung einer abschließenden Konsistenzprüfung der vom Stack erstellten Ressourcen aufgewendet. Während dieser Phase führt der Service interne Konsistenzprüfungen durch, um sicherzustellen, dass die Ressource voll funktionsfähig ist und die jeweils definierten Servicestabilisierungskriterien erfüllt. AWS-Service
- Ereignis „Konfiguration abgeschlossen“ — Wenn jede Ressource die abschließende Phase der Konsistenzprüfung der Bereitstellung abgeschlossen hat, wird ein detaillierter Status des `CONFIGURATION_COMPLETE` Ereignisses festgelegt.
- Ereignis „Abschluss der Ressourcenerstellung“ — Nachdem die Ressource wie angegeben erstellt und konfiguriert wurde und die Konfiguration mit den Angaben in der Vorlage übereinstimmt, wird der Status des `CREATE_COMPLETE` Ereignisses festgelegt.

Sie können das `CONFIGURATION_COMPLETE` Ereignis nutzen, um Ihren Stack-Erstellungsprozess in Szenarien zu optimieren, in denen eine abschließende Konsistenzprüfung der Ressourcen nicht erforderlich ist, z. B. bei der Validierung einer Stack-Konfiguration vor der Produktion oder bei der stapelübergreifenden Bereitstellung. Sie können dieses Ereignis auf verschiedene Arten nutzen. Sie können es beispielsweise als visuelles Signal verwenden, um das Warten auf den Abschluss der Ressourcen- oder Stack-Konsistenzprüfung zu überspringen. Oder Sie könnten es verwenden, um mithilfe von Continuous Integration and Continuous Delivery (CI/CD) einen automatisierten Mechanismus zu erstellen, um zusätzliche Aktionen auszulösen.

> **Important**
>
> Die Nutzung `CONFIGURATION_COMPLETE` dieses Ereignisses beschleunigt zwar die Zeit für die Erstellung von Stacks, Sie sollten sich jedoch der damit verbundenen Kompromisse bewusst sein. Erstens wird es nur für eine Teilmenge von Ressourcentypen unterstützt, die die Drift-Erkennung unterstützen. Eine Liste der Ressourcentypen, die die Drift-Erkennung unterstützen, finden Sie unterUnterstützung für Ressourcentypen. Dieser Ansatz ist möglicherweise nicht für alle Szenarien geeignet, insbesondere dann, wenn Ressourcen letztendlich gründliche Konsistenzprüfungen erfordern, um die volle Betriebsbereitschaft

in der gesamten Cloud-Umgebung sicherzustellen (z. B. in Produktionsumgebungen). Wir empfehlen, Ihre Bereitstellungsanforderungen und die Wichtigkeit der Konsistenzprüfungen für jede Ressource sorgfältig zu prüfen. Nutzen Sie das `CONFIGURATION_COMPLETE` Ereignis, um die Bereitstellungsgeschwindigkeit zu optimieren, ohne die Integrität und Zuverlässigkeit Ihrer Infrastruktur zu gefährden.
Da nicht garantiert werden kann, dass das `CONFIGURATION_COMPLETE` Ereignis stattfindet, sollten alle Szenarien, in denen es verwendet wird, darauf vorbereitet sein, auch ein `CREATE_COMPLETE` Ereignis zu behandeln, für das kein `CONFIGURATION_COMPLETE` Ereignis festgelegt wurde.

Wenn die Stack-Bereitstellung beginnt, starten `AWS::ECR::Repository` sowohl die Ressourcen als auch die `AWS::ECS::Cluster` Ressourcen den Erstellungsprozess (`ResourceStatus=CREATE_IN_PROGRESS`. Wenn der `AWS::ECR::Repository` Ressourcentyp die eventuelle Konsistenzprüfung (`DetailedStatus=CONFIGURATION_COMPLETE`) gestartet hat, kann die `AWS::ECS::TaskDefinition` Ressource den Erstellungsprozess starten. In ähnlicher Weise beginnt die `AWS::ECS::TaskDefinition` Ressource, sobald die `AWS::ECS::Service` Ressource mit der abschließenden Konsistenzprüfung beginnt, den Erstellungsprozess.

`CREATE_IN_PROGRESS`- und **`CREATE_COMPLETE`**-Ereignisse

- [Stapel]: `CREATE_IN_PROGRESS`
- [Ressource]: ECR-Repository `CREATE_IN_PROGRESS`
- [Ressource]: ECS-Cluster `CREATE_IN_PROGRESS`
- [Ressource]: ECR-Repository`CREATE_IN_PROGRESS`, `CONFIGURATION_COMPLETE`
- [Ressource]: ECS-Aufgabendefinition `CREATE_IN_PROGRESS`
- [Ressource]: ECS-Cluster`CREATE_IN_PROGRESS`, `CONFIGURATION_COMPLETE`
- [Ressource]: ECS-Aufgabendefinition`CREATE_IN_PROGRESS`, `CONFIGURATION_COMPLETE`
- [Ressource]: ECS-Dienst `CREATE_IN_PROGRESS`
- [Ressource]: ECR-Repository `CREATE_COMPLETE`
- [Ressource]: ECS-Cluster `CREATE_COMPLETE`
- [Ressource]: ECS-Dienst`CREATE_IN_PROGRESS`, `CONFIGURATION_COMPLETE`
- [Stapel]:`CREATE_IN_PROGRESS`, `CONFIGURATION_COMPLETE`

- [Ressource]: ECS-Aufgabendefinition `CREATE_COMPLETE`
- [Ressource]: ECS-Dienst `CREATE_COMPLETE`
- [Stapel]: `CREATE_COMPLETE`

Überwachen Sie den Fortschritt eines Stack-Updates

Sie können den Fortschritt der Aktualisierung eines Stacks überwachen, indem Sie die Ereignisse des Stacks anzeigen. Auf der Registerkarte Ereignisse des Stacks werden alle wichtigen Schritte bei der Erstellung und Aktualisierung des Stacks nach dem Zeitpunkt des jeweiligen Ereignisses sortiert angezeigt, wobei die neuesten Ereignisse ganz oben stehen. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

Themen

- Ereignisse, die während eines erfolgreichen Stack-Updates generiert wurden
- Ereignisse, die bei einem Fehlschlagen eines Ressourcen-Updates generiert werden

Ereignisse, die während eines erfolgreichen Stack-Updates generiert wurden

Der Start des Stack-Aktualisierungsprozesses ist mit einem `UPDATE_IN_PROGRESS` Ereignis für den Stack gekennzeichnet:

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_IN_PROGRESS
```

Als Nächstes kommen die Ereignisse, die den Beginn und den Abschluss der Aktualisierung jeder Ressource darstellen, die in der Updatevorlage geändert wurde. Die Aktualisierung einer AWS::RDS::DBInstanceRessource mit dem Namen `MyDB` würde beispielsweise zu den folgenden Einträgen führen:

```
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

Das `UPDATE_IN_PROGRESS` Ereignis wird protokolliert, wenn CloudFormation gemeldet wird, dass mit der Aktualisierung der Ressource begonnen wurde. Das `UPDATE_COMPLETE`-Ereignis wird protokolliert, wenn die Ressource erfolgreich erstellt wurde.

Wenn der Stack erfolgreich aktualisiert CloudFormation wurde, wird das folgende Ereignis angezeigt:

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_COMPLETE
```

> **Important**
>
> Wenn bei Stack-Aktualisierungsvorgängen eine vorhandene Ressource ersetzt CloudFormation werden muss, wird zunächst eine neue Ressource erstellt und dann die alte Ressource gelöscht. Es kann jedoch Fälle geben, in denen die alte Ressource nicht gelöscht werden CloudFormation kann (z. B. wenn der Benutzer nicht berechtigt ist, eine Ressource eines bestimmten Typs zu löschen).
>
> CloudFormation unternimmt drei Versuche, die alte Ressource zu löschen. Wenn die alte Ressource nicht gelöscht werden CloudFormation kann, wird die alte Ressource aus dem Stapel entfernt und der Stapel wird weiter aktualisiert. Wenn das Stack-Update abgeschlossen ist, wird CloudFormation ein `UPDATE_COMPLETE` Stack-Ereignis ausgelöst, `StatusReason` das jedoch ein Ereignis enthält, das besagt, dass eine oder mehrere Ressourcen nicht gelöscht werden konnten. CloudFormation löst außerdem ein `DELETE_FAILED` Ereignis für die spezifische Ressource aus, in dem ein entsprechendes `StatusReason` Ereignis ausführlicher beschrieben wird, warum die Ressource nicht gelöscht CloudFormation werden konnte.
>
> Die alte Ressource ist immer noch vorhanden und es fallen weiterhin Gebühren an, sie ist aber nicht mehr zugänglich über CloudFormation. Um die alte Ressource zu löschen, greifen Sie direkt über die Konsole oder API für den zugrunde liegenden Service auf die alte Ressource zu.
>
> Dies gilt auch für Ressourcen, die Sie aus der Stack-Vorlage entfernt haben und daher während des Stack-Updates aus dem Stack gelöscht werden.

Ereignisse, die bei einem Fehlschlagen eines Ressourcen-Updates generiert werden

Wenn die Aktualisierung einer Ressource fehlschlägt, wird ein `UPDATE_FAILED` Ereignis CloudFormation gemeldet, das einen Grund für den Fehler enthält. Wenn in Ihrer Aktualisierungsvorlage beispielsweise eine Eigenschaftsänderung angegeben ist, die von der Ressource nicht unterstützt wird, z. B. die Reduzierung der `AllocatedStorage` Größe einer AWS::RDS::DBInstanceRessource, würden Ereignisse wie die folgenden auftreten:

```
2011-09-30 09:36 PDT AWS::RDS::DBInstance MyDB UPDATE_FAILED Size cannot be less than
 current size; requested: 5; current: 10
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

Wenn ein Ressourcenupdate CloudFormation fehlschlägt, werden alle Ressourcen, die während des Upgrades aktualisiert wurden, auf ihre Konfigurationen vor dem Update zurückgesetzt. Hier finden Sie ein Beispiel für die Ereignisse, die während des Rollbacks eines Updates angezeigt würden:

```
2011-09-30 09:38 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_COMPLETE
2011-09-30 09:38 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:37 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
2011-09-30 09:37 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_IN_PROGRESS The
 following resource(s) failed to update: [MyDB]
```

Rollback einer Aktualisierung fortsetzen

Wenn CloudFormation versucht wird, ein Stack-Update rückgängig zu machen, kann es manchmal nicht alle während des Aktualisierungsvorgangs vorgenommenen Änderungen rückgängig machen. Dies wird als `UPDATE_ROLLBACK_FAILED` Staat bezeichnet. Sie haben beispielsweise einen Stack, der mit dem Rollback zu einer alten Datenbankinstanz beginnt, die außerhalb von gelöscht wurde CloudFormation. Da CloudFormation nicht weiß, dass die Datenbank gelöscht wurde, wird davon ausgegangen, dass die Datenbank-Instance noch vorhanden ist und versucht, sie wiederherstellen, wodurch das Rollback der Aktualisierung fehlschlägt.

Ein Stack im `UPDATE_ROLLBACK_FAILED` Status kann nicht aktualisiert werden, aber er kann auf einen funktionierenden Status (`UPDATE_ROLLBACK_COMPLETE`) zurückgesetzt werden. Nachdem Sie den Stack auf seine ursprünglichen Einstellungen zurückgesetzt haben, können Sie erneut versuchen, ihn zu aktualisieren.

In den meisten Fällen müssen Sie den Fehler beheben, der das Fehlschlagen des Rollback der Aktualisierung verursacht hat, bevor Sie mit dem Zurücksetzen des Stacks fortfahren können. In anderen Fällen können Sie auch ohne Änderungen ein Rollback der Aktualisierung durchführen, z. B., wenn bei einem Stack-Vorgang eine Zeitüberschreitung auftritt.

> **Note**
>
> Wenn Sie verschachtelte Stacks verwenden, wird beim Rollback des übergeordneten Stacks versucht, auch ein Rollback für alle untergeordneten Stacks durchzuführen.

So setzen Sie den Rollback-Vorgang für eine Aktualisierung fort (Konsole)

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm aus, AWS-Region wo sich der Stapel befindet.
3. Wählen Sie auf der Seite Stacks den Stack aus, den Sie aktualisieren möchten, wählen Sie Stack-Aktionen und dann Continue update rollback aus.

   Wenn keine der Lösungen Behebung von Fehlern funktioniert hat, können Sie die erweiterte Option verwenden, um die Ressourcen zu überspringen, für die ein Rollback nicht erfolgreich durchgeführt CloudFormation werden konnte. Sie müssen die logische Reihenfolge IDs der Ressourcen, die Sie überspringen möchten, nachschlagen und eingeben. Geben Sie nur Ressourcen an, die während des Forward-Updates `UpdateRollback` und nicht während des Forward-Updates in diesen `UPDATE_FAILED` Status übergegangen sind.

   > **Warning**
   >
   > CloudFormation setzt den Status der angegebenen Ressourcen auf `UPDATE_COMPLETE` und setzt das Rollback des Stacks fort. Nachdem der Rollback-Vorgang abgeschlossen ist, entspricht der Status der übersprungenen Ressourcen nicht dem Status der Ressourcen in der Stack-Vorlage. Bevor Sie eine weitere Stack-Aktualisierung durchführen, müssen Sie den Stack oder die Ressourcen aktualisieren, sodass ihr Status konsistent ist. Andernfalls schlagen nachfolgende Aktualisierungen fehl und der Stack lässt sich nicht mehr wiederherstellen.

   Geben Sie die Mindestanzahl von Ressourcen, die erforderlich ist, um einen erfolgreichen Rollback-Vorgang mit dem Stack durchzuführen. Beispielsweise kann eine fehlgeschlagene Ressourcenaktualisierung dazu führen, dass abhängige Ressourcen fehlschlagen. In diesem Fall ist es u. U. nicht erforderlich, die abhängigen Ressourcen zu überspringen.

   Verwenden Sie das folgende Format, um Ressourcen, die Teil verschachtelter Stacks sind, zu überspringen: *`NestedStackName`*`.`*`ResourceLogicalID`*. Wenn Sie die logische ID einer Stack-Ressource (`Type: AWS::CloudFormation::Stack`) in der `ResourcesToSkip`-Liste angeben möchten, dass muss sich der zugehörige eingebettete Stapel in einem der folgenden Zustände befinden: `DELETE_IN_PROGRESS`, `DELETE_COMPLETE` oder `DELETE_FAILED`.

So setzen Sie den Rollback einer Aktualisierung fort (AWS CLI)

- Verwenden Sie den Befehl continue-update-rollback mit der Option `--stack-name`, um die ID des Stacks anzugeben, bei dem Sie den Rollback-Vorgang fortsetzen möchten.

Setzt das Rollback nach fehlgeschlagenen Aktualisierungen des verschachtelten Stacks fort

Wenn Sie mehrere ineinander verschachtelte Stapel haben, müssen Sie möglicherweise Ressourcen auf mehreren verschachtelten Ebenen überspringen, um die gesamte Stack-Hierarchie wieder funktionsfähig zu machen.

Sie haben beispielsweise einen Root-Stack namens`WebInfra`, der zwei kleinere Stapel enthält: und. `WebInfra-Compute` `WebInfra-Storage` Diese beiden Stapel enthalten auch ihre eigenen verschachtelten Stapel.

Wenn bei einem Update etwas schief geht und der Aktualisierungsvorgang fehlschlägt, kann es sein, dass die gesamte Stack-Hierarchie den `UPDATE_ROLLBACK_FAILED` Status annimmt, wie in der folgenden Abbildung dargestellt.

> **Note**
>
> In diesem Beispiel werden der Einfachheit halber verkürzte Stack-Namen verwendet. Namen untergeordneter Stacks werden in der Regel durch eindeutige zufällige Zeichenketten generiert CloudFormation und enthalten diese, sodass tatsächliche Namen möglicherweise nicht benutzerfreundlich sind.

Um den Root-Stack mithilfe des `continue-update-rollback` Befehls in einen betriebsbereiten Zustand zu versetzen, müssen Sie die `--resources-to-skip` Option zum Überspringen von Ressourcen verwenden, für die ein Rollback fehlgeschlagen ist.

Im folgenden continue-update-rollback Beispiel wird ein Rollback-Vorgang nach einem zuvor fehlgeschlagenen Stack-Update wieder aufgenommen. In diesem Beispiel umfasst die `--resources-to-skip` Option die folgenden Elemente:

- *`myCustom`*
- *`WebInfra-Compute-Asg.myAsg`*

- *WebInfra-Compute-LB.myLoadBalancer*
- *WebInfra-Storage.DB*

Für die Ressourcen des Root-Stacks müssen Sie nur die logische ID angeben, zum Beispiel*myCustom*. Für die Ressourcen, die in verschachtelten Stacks enthalten sind, müssen Sie jedoch sowohl den Namen des verschachtelten Stacks als auch seine logische ID, getrennt durch einen Punkt, angeben. Beispiel, *WebInfra-Compute-Asg.myAsg*.

```
aws cloudformation continue-update-rollback --stack-name WebInfra \
    --resources-to-skip myCustom WebInfra-Compute-Asg.myAsg WebInfra-Compute-
LB.myLoadBalancer WebInfra-Storage.DB
```

Um den Stacknamen eines verschachtelten Stacks zu ermitteln

Sie finden ihn in der Stack-ID oder dem Amazon-Ressourcennamen (ARN) des untergeordneten Stacks.

Das folgende ARN-Beispiel bezieht sich auf einen Stack mit dem Namen`WebInfra-Storage-Z2VKC706XKXT`.

```
arn:aws:cloudformation:us-east-1:123456789012:stack/WebInfra-Storage-Z2VKC706XKXT/
ea9e7f90-54f7-11e6-a032-028f3d2330bd
```

Um die logische ID eines verschachtelten Stacks zu finden

Die logische ID eines untergeordneten Stacks ist in der Vorlagendefinition seines übergeordneten Stacks enthalten. Im Diagramm befindet sich der `LogicalId` des `WebInfra-Storage-DB` untergeordneten Stacks `DB` in seinem übergeordneten `WebInfra-Storage` Stapel.

In der CloudFormation Konsole finden Sie die logische ID auch in der Spalte Logische ID für die Stack-Ressource auf der Registerkarte Ressourcen oder auf der Registerkarte Ereignisse. Weitere Informationen finden Sie unter Stack-Informationen von der CloudFormation Konsole aus anzeigen.

Ermitteln Sie die Ursache eines Stack-Fehlers

Wenn Ihre Stack-Erstellung fehlschlägt, CloudFormation kann es Ihnen helfen, das Ereignis zu ermitteln, das wahrscheinlich die Hauptursache für den Stack-Ausfall ist. Abhängig vom Szenario und Ihren Berechtigungen können AWS CloudTrail Ereignisse möglicherweise weitere Details zur Grundursache liefern, falls der angegebene Statusgrund unter Ereignisse nicht klar ist.

Um die Hauptursache eines Stack-Fehlers zu ermitteln

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der Seite Stacks den fehlgeschlagenen Stack aus.
3. Wählen Sie den Tab „Ereignisse“.
4. Wählen Sie „Ursache ermitteln“. CloudFormation analysiert den Fehler und gibt das Ereignis an, das die wahrscheinliche Ursache für den Fehler darstellt, indem dem jeweiligen Ereignisstatus die Bezeichnung Wahrscheinliche Hauptursache hinzugefügt wird. Weitere Informationen zum Status finden Sie in der CloudFormation Konsole unter Grund des Status.
5. Wählen Sie den Status „Fehlgeschlagen“ mit der Bezeichnung „Wahrscheinliche Ursache“, um mehr über die Ursache des Fehlers zu erfahren. Je nach Szenario und Ihren Berechtigungen können Sie möglicherweise ein detailliertes CloudTrail Ereignis überprüfen. Dies sind die folgenden möglichen Ergebnisse bei der Auswahl des Status
   - CloudTrail Ereignisse im Zusammenhang mit diesem Problem sind verfügbar und können bei der Lösung helfen. CloudTrail Ereignisse anzeigen.
   - Wir konnten keine CloudTrail Ereignisse im Zusammenhang mit diesem Problem finden, die bei der Lösung helfen könnten.
   - Ihre aktuellen Berechtigungen erlauben keinen Zugriff auf CloudTrail Ereignisse. Weitere Informationen.
   - Schauen Sie in ein paar Minuten erneut nach, während Sie nach verfügbaren CloudTrail Ereignissen suchen.
   - Beim Abrufen der CloudTrail Ereignisse ist ein Fehler aufgetreten. Rufen Sie die CloudTrail Konsole auf, um die manuelle Überprüfung durchzuführen.
6. Wenn der unter Statusgrund angegebene Grund nicht klar ist und für die Ursache ein Link zur CloudTrail Konsole angezeigt wird, öffnen Sie den Link, um das Ereignis anzuzeigen und eine detaillierte Ursache zu finden.

Weitere Informationen zu CloudTrail Ereignissen finden Sie unter Grundlegendes zu CloudTrail Ereignissen und CloudTrail Datensatzinhalten.

Weitere Informationen zum CloudTrail Ereignisverlauf finden Sie unter Arbeiten mit dem CloudTrail Ereignisverlauf.

> **Note**
>
> Verschachtelte Stacks unterstützen die Ursachenerkennung nicht.

Wählen Sie aus, wie mit Ausfällen bei der Bereitstellung von Ressourcen umgegangen werden soll

Wenn Ihr Stack-Vorgang fehlschlägt, müssen Sie Ressourcen, die bereits erfolgreich bereitgestellt wurden, nicht rückgängig machen und jedes Mal von vorne beginnen. Stattdessen können Sie Probleme mit dem `UPDATE_FAILED` Status `CREATE_FAILED` oder beheben und die Bereitstellung dann an dem Punkt fortsetzen, an dem das Problem aufgetreten ist.

Dazu müssen Sie die Option Erfolgreich bereitgestellte Ressourcen beibehalten aktivieren. Diese Option ist für alle Stack-Bereitstellungen und Change-Set-Operationen verfügbar.

- Wenn Sie bei der Stack-Erstellung die Option Erfolgreich bereitgestellte Ressourcen beibehalten wählen, CloudFormation wird der Status der erfolgreich erstellten Ressourcen beibehalten und die fehlgeschlagenen Ressourcen so belassen, bis der nächste Aktualisierungsvorgang ausgeführt wird.
- Wenn Sie bei Aktualisierungs- und Änderungssatzvorgängen die Option Erfolgreich bereitgestellte Ressourcen beibehalten wählen, wird der Status erfolgreicher Ressourcen beibehalten, während ausgefallene Ressourcen auf ihren letzten bekannten stabilen Zustand zurückgesetzt werden. Fehlgeschlagene Ressourcen befinden sich in einem `UPDATE_FAILED`-Status. Ressourcen ohne letzten bekannten stabilen Zustand werden beim nächsten Stack-Vorgang gelöscht.

Themen

- Überblick über Optionen für Stack-Ausfälle
- Erforderliche Bedingungen für das Anhalten des Stack-Rollbacks
- Erfolgreich bereitgestellte Ressourcen beibehalten (Konsole)
- Erfolgreich bereitgestellte Ressourcen beibehalten (AWS CLI)

Überblick über Optionen für Stack-Ausfälle

Bevor Sie einen Vorgang über die CloudFormation Konsole, die API oder ausführen AWS CLI, geben Sie das Verhalten bei einem Ausfall bereitgestellter Ressourcen an. Fahren Sie dann mit

dem Bereitstellungsprozess Ihrer Ressourcen fort, ohne weitere Änderungen vorzunehmen. Im Falle eines Betriebsausfalls CloudFormation stoppt es beim ersten Fehler in jedem unabhängigen Bereitstellungspfad. CloudFormation identifiziert Abhängigkeiten zwischen Ressourcen, um unabhängige Bereitstellungsaktionen zu parallelisieren. Anschließend werden Ressourcen auf jedem unabhängigen Bereitstellungspfad bereitgestellt, bis ein Fehler auftritt. Ein Fehler in einem Pfad hat keine Auswirkungen auf andere Bereitstellungspfade. CloudFormation setzt die Bereitstellung der Ressourcen bis zum Abschluss fort oder stoppt bei einem anderen Fehler.

Beheben Sie alle Probleme, um den Bereitstellungsprozess fortzusetzen. CloudFormation führt die erforderlichen Aktualisierungen durch, bevor erneut versucht wird, Provisionierungsaktionen für Ressourcen durchzuführen, die zuvor nicht erfolgreich bereitgestellt werden konnten. Sie beheben Probleme, indem Sie Retry-, Update- oder Rollback-Vorgänge einsenden. Wenn Sie beispielsweise eine EC2 Amazon-Instance bereitstellen und die EC2 Instance während eines Erstellungsvorgangs fehlschlägt, sollten Sie den Fehler untersuchen, anstatt die ausgefallene Ressource sofort rückgängig zu machen. Sie können System-Statusprüfungen und Instance-Statusprüfungen überprüfen und dann den Retry-Vorgang auswählen, sobald die Probleme gelöst sind.

Wenn ein Stack-Vorgang fehlschlägt und Sie Preserve successfully provisioned resources (Erfolgreich bereitgestellte Ressourcen beibehalten) aus dem Stack failure options (Stack-Fehleroptionen)-Menü angegeben haben, können Sie die folgenden Optionen auswählen.

- Retry – Versuchte den Bereitstellungsvorgang für ausgefallene Ressourcen erneut und setzt die Bereitstellung der Vorlage fort, bis der Stack-Vorgang erfolgreich abgeschlossen ist oder der nächste Fehler auftritt. Wählen Sie diese Option, wenn die Ressource aufgrund eines Problems, für das keine Änderungen an der Vorlage erforderlich sind, nicht bereitgestellt werden konnte, z. B. eine AWS Identity and Access Management (IAM-) Genehmigung.
- Update – Ressourcen, die bereitgestellt wurden, werden bei Vorlagen-Aktualisierungen aktualisiert. Ressourcen, die nicht erstellt oder aktualisiert wurden, werden erneut versucht. Wählen Sie diese Option, wenn die Ressource aufgrund von Vorlagenfehlern nicht bereitgestellt werden konnte und Sie die Vorlage geändert haben. Wenn Sie einen Stack aktualisieren, der sich in einem `FAILED`-Status befindet, müssen Sie unter Stack failure options (Stack-Fehleroptionen) Preserve successfully provisioned resources (Erfolgreich bereitgestellte Ressourcen beibehalten) auswählen, um mit der Aktualisierung Ihres Stacks fortzufahren.
- Rollback — CloudFormation Setzt den Stack auf den letzten bekannten stabilen Zustand zurück.

Erforderliche Bedingungen für das Anhalten des Stack-Rollbacks

Um zu CloudFormation verhindern, dass die erfolgreich erstellten Ressourcen automatisch zurückgesetzt und gelöscht werden, müssen die folgenden Bedingungen erfüllt sein.

1. Wenn Sie den Stack erstellen oder aktualisieren, müssen Sie die Option Erfolgreich bereitgestellte Ressourcen beibehalten auswählen. Dies weist darauf hin, dass die erfolgreich erstellten Ressourcen CloudFormation nicht gelöscht werden sollen, auch wenn der gesamte Stack-Vorgang fehlschlägt.
2. Der Stack-Vorgang muss fehlgeschlagen sein, was bedeutet, dass der Stack-Status entweder `CREATE_FAILED` oder lautet`UPDATE_FAILED`.

> **Note**
>
> Unveränderliche Update-Typen werden nicht unterstützt.

Erfolgreich bereitgestellte Ressourcen beibehalten (Konsole)

Create stack

So können Sie erfolgreich bereitgestellte Ressourcen während eines Stack-Erstellungsvorgangs beibehalten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der Seite „Stacks“ oben rechts die Option „Stack erstellen“ und anschließend „Mit neuen Ressourcen (Standard)“ aus.
3. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Bestehende Vorlage auswählen aus.
4. Wählen Sie unter Specify template (Vorlage angeben) entweder die URL für den S3-Bucket, der Ihre Stack-Vorlage enthält, aus oder laden Sie eine Stack-Vorlagendatei hoch. Wählen Sie anschließend Weiter.
5. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) einen Stack-Namen in das Feld Stack name ein.
6. Geben Sie im Abschnitt Parameter die Parameter an, die in Ihrer Stack-Vorlage definiert sind.

Sie können beliebige Parameter mit Standardwerten verwenden oder ändern.

7. Wenn Sie mit den Parameterwerten zufrieden sind, wählen Sie Next (Weiter).
8. Auf der Seite Configure stack options (Stack-Optionen konfigurieren) können Sie zusätzliche Optionen für Ihren Stack einrichten.
9. Für Stack failure options (Stack-Fehleroptionen), wählen Sie Preserve successfully provisioned resources (Erfolgreich bereitgestellte Ressourcen beibehalten).
10. Wenn Sie mit den Stack-Optionen zufrieden sind, wählen Sie Next (Weiter).
11. Überprüfen Sie Ihren Stack auf der Seite Review (Überprüfen) und wählen Sie Create stack (Stack erstellen).

Ergebnisse: Ressourcen, die nicht erstellt werden konnten, übertragen den Stack-Status auf `CREATE_FAILED`, um zu verhindern, dass der Stack zurückgesetzt wird, wenn der Stack-Vorgang auf einen Fehler trifft. Ressourcen, die erfolgreich bereitgestellt wurden, befinden sich in einem `CREATE_COMPLETE`-Status. Sie können den Stack auf der Registerkarte Stack events (Stack-Ereignisse) überwachen.

Update stack

Beibehaltung erfolgreich bereitgestellter Ressourcen während eines Stack-Aktualisierungsvorgangs.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie den Stack aus, den Sie aktualisieren möchten, und anschließend Update (Aktualisieren).
3. Wählen Sie auf der Seite Update stack (Stack aktualisieren) mit einer der folgenden Optionen eine Stack-Vorlage aus:

   - Verwenden Sie die vorhandene Vorlage
   - Replace current template (Aktuelle Vorlage ersetzen)
   - Vorlage in Infrastructure Composer bearbeiten

   Übernehmen Sie Ihre Einstellungen und wählen Sie Next (Weiter).
4. Auf der Seite Specify stack details (Stack-Details angeben), legen Sie die Parameter fest, die in Ihrer Stack-Vorlage definiert sind.

Sie können beliebige Parameter mit Standardwerten verwenden oder ändern.

5. Wenn Sie mit den Parameterwerten zufrieden sind, wählen Sie Next (Weiter).
6. Auf der Seite Configure stack options (Stack-Optionen konfigurieren) können Sie zusätzliche Optionen für Ihren Stack einrichten.
7. Wählen Sie für Behaviour on provisioning failure (Verhalten bei Bereitstellungsausfall) Preserve successfully provisioned resources (Erfolgreich bereitgestellte Ressourcen beibehalten).
8. Wenn Sie mit den Stack-Optionen zufrieden sind, wählen Sie Next (Weiter).
9. Überprüfen Sie Stack auf der Seite Review (Überprüfen) und wählen Sie Update stack (Stack aktualisieren).

Ergebnisse: Ressourcen, die nicht aktualisiert werden konnten, übertragen den Stack-Status auf `UPDATE_FAILED` und setzen Sie auf den letzten bekannten stabilen Zustand zurück. Ressourcen ohne einen letzten bekannten stabilen Status werden CloudFormation beim nächsten Stack-Vorgang gelöscht. Ressourcen, die erfolgreich bereitgestellt wurden, befinden sich in einem `CREATE_COMPLETE`- oder `UPDATE_COMPLETE`-Status. Sie können den Stack auf der Registerkarte Stack events (Stack-Ereignisse) überwachen.

Change set

> **Note**
>
> Sie können einen Änderungssatz für einen Stack mit dem Status von `CREATE_FAILED` oder `UPDATE_FAILED` initiieren nicht aber für einen Status von `UPDATE_ROLLBACK_FAILED`.

So können Sie erfolgreich bereitgestellte Ressourcen während eines Änderungssatz-Vorgangs beibehalten

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie den Stack mit dem Änderungssatz, den Sie starten möchten, und klicken Sie dann auf die Registerkarte Change sets (Änderungssätze).
3. Wählen Sie den Änderungssatz aus und klicken Sie dann auf Execute (Ausführen).

4. Wählen Sie für Execute change set (Änderungssatz ausführen) die Option Preserve successfully provisioned resources (Erfolgreich bereitgestellte Ressourcen beibehalten).
5. Wählen Sie Execute change set (Änderungssatz ausführen).

Ergebnisse: Ressourcen, die nicht aktualisiert werden konnten, übertragen den Stack-Status auf `UPDATE_FAILED` und setzen Sie auf den letzten bekannten stabilen Zustand zurück. Ressourcen ohne einen letzten bekannten stabilen Status werden CloudFormation beim nächsten Stack-Vorgang gelöscht. Ressourcen, die erfolgreich bereitgestellt wurden, befinden sich in einem `CREATE_COMPLETE`- oder `UPDATE_COMPLETE`-Status. Sie können den Stack auf der Registerkarte Stack events (Stack-Ereignisse) überwachen.

Erfolgreich bereitgestellte Ressourcen beibehalten (AWS CLI)

Create stack

So können Sie erfolgreich bereitgestellte Ressourcen während eines Stack-Erstellungsvorgangs beibehalten

Geben Sie die `--disable-rollback`-Option oder die `on-failure DO_NOTHING`-Aufzählung während eines create-stack-Vorgangs ein.

1. Geben Sie einen Stack-Namen und eine Vorlage für den create-stack-Befehl mit der `--disable-rollback`-Option an.

```
aws cloudformation create-stack --stack-name myteststack \
    --template-body file://template.yaml \
    --disable-rollback
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

2. Beschreiben Sie den Status des Stacks mit dem describe-stacks Befehl.

```
aws cloudformation describe-stacks --stack-name myteststack
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
    "Stacks":  [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "Description": "AWS CloudFormation Sample Template",
            "Tags": [],
            "Outputs": [],
            "StackStatusReason": “The following resource(s) failed to create:
 [MyBucket]”,
            "CreationTime": "2013-08-23T01:02:15.422Z",
            "Capabilities": [],
            "StackName": "myteststack",
            "StackStatus": "CREATE_FAILED",
            "DisableRollback": true
        }
    ]
}
```

Update stack

Um erfolgreich bereitgestellte Ressourcen während eines Stack-Aktualisierungsvorgangs beizubehalten

1. Geben Sie einen vorhandenen Stack-Namen und eine Vorlage für den update-stack-Befehl mit der `--disable-rollback`-Option an.

```
aws cloudformation update-stack --stack-name myteststack \
    --template-url https://s3.amazonaws.com/amzn-s3-demo-bucket/updated.template
 --disable-rollback
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

2. Beschreiben Sie den Status des Stacks entweder mit dem describe-stack-events Befehl describe-stacks oder.

```
aws cloudformation describe-stacks --stack-name myteststack
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
    "Stacks":  [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "Description": "AWS CloudFormation Sample Template",
            "Tags": [],
            "Outputs": [],
            "CreationTime": "2013-08-23T01:02:15.422Z",
            "Capabilities": [],
            "StackName": "myteststack",
            "StackStatus": "UPDATE_COMPLETE",
            "DisableRollback": true
        }
    ]
}
```

Change set

> **Note**
>
> Sie können einen Änderungssatz für einen Stack mit dem Status von `CREATE_FAILED` oder `UPDATE_FAILED` starten nicht aber für einen Status von `UPDATE_ROLLBACK_FAILED`.

So können Sie erfolgreich bereitgestellte Ressourcen während eines Änderungssatz-Vorgangs beibehalten

Geben Sie die `--disable-rollback` Option während eines execute-change-setVorgangs an.

1. Geben Sie einen Stack-Namen und eine Vorlage für den execute-change-set-Befehl mit der `--disable-rollback`-Option an.

```
aws cloudformation execute-change-set --stack-name myteststack \
    --change-set-name my-change-set --template-body file://template.yaml
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
 "Id": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/my-change-set/
bc9555ba-a949-xmpl-bfb8-f41d04ec5784",
 "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

2. Starten Sie den Änderungssatz mit Option `--disable-rollback`.

```
aws cloudformation execute-change-set --stack-name myteststack \
    --change-set-name my-change-set --disable-rollback
```

3. Ermitteln Sie den Status des Stacks entweder mit dem describe-stack-events Befehl describe-stacks oder.

```
aws cloudformation describe-stack-events --stack-name myteststack
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
   "StackEvents": [
     {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
        "EventId": "49c966a0-7b74-11ea-8071-024244bb0672",
        "StackName": "myteststack",
        "LogicalResourceId": " MyBucket",
        "PhysicalResourceId": "myteststack-MyBucket-abcdefghijk1",
        "ResourceType": "AWS::S3::Bucket",
        "Timestamp": "2020-04-10T21:43:17.015Z",
        "ResourceStatus": "UPDATE_FAILED"
        "ResourceStatusReason": "User XYZ is not allowed to perform
 S3::UpdateBucket on MyBucket"
     }
}
```

4. Korrigieren Sie Berechtigungsfehler und versuchen Sie den Vorgang erneut.

```
aws cloudformation update-stack --stack-name mytestsstack \
    --use-previous-template --disable-rollback
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
mytestsstack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

5. Beschreiben Sie den Status des Stacks entweder mit dem describe-stack-events Befehl describe-stacks oder.

```
aws cloudformation describe-stacks --stack-name mytestsstack
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
    "Stacks":  [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
mytestsstack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "Description": "AWS CloudFormation Sample Template",
            "Tags": [],
            "Outputs": [],
            "CreationTime": "2013-08-23T01:02:15.422Z",
            "Capabilities": [],
            "StackName": "mytestsstack",
            "StackStatus": "UPDATE_COMPLETE",
            "DisableRollback": true
        }
    ]
}
```

Einen Stapel rückgängig machen

Sie können den Befehl rollback-stack verwenden, um einen Stack mit einem `CREATE_FAILED` oder `UPDATE_FAILED` Stack-Status auf seinen letzten stabilen Zustand zurückzusetzen.

Mit dem folgenden rollback-stack Befehl wird der angegebene Stack zurückgesetzt.

```
aws cloudformation rollback-stack --stack-name mytestsstack
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
mytestsstack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

> **Note**
>
> Der rollback-stack-Vorgang löscht einen Stack, wenn er keinen letzten bekannten stabilen Zustand enthält.

Setzen Sie Ihren CloudFormation Stack bei einem Alarmverstoß mit Rollback-Triggern zurück

Mit Rollback-Triggern können Sie den Status Ihrer Anwendung während der Erstellung und Aktualisierung des Stacks AWS CloudFormation überwachen und diesen Vorgang rückgängig machen, wenn die Anwendung den Schwellenwert der von Ihnen angegebenen Alarme überschreitet. Für jeden Rollback-Trigger, den Sie erstellen, geben Sie den CloudWatch Alarm an, CloudFormation der überwacht werden soll. CloudFormation überwacht die angegebenen Alarme während der Erstellung oder Aktualisierung des Stacks und für den angegebenen Zeitraum, nachdem alle Ressourcen bereitgestellt wurden. Wenn einer der Alarme während des Stack-Vorgangs oder des Überwachungszeitraums in den `ALARM` Status wechselt, CloudFormation wird der gesamte Stack-Vorgang zurückgesetzt.

Sie können eine Überwachungszeit vom Standardwert 0 auf bis zu 180 Minuten festlegen. Während dieser Zeit überwacht CloudFormation alle Rollback-Trigger, nachdem die Stack-Erstellung oder -Aktualisierung alle erforderlichen Ressourcen bereitgestellt hat. Wenn einer der Alarme während des Stack-Betriebs oder während dieser Überwachungsperiode in den `ALARM` Status wechselt, CloudFormation wird der gesamte Stack-Vorgang zurückgesetzt. Wenn dann bei Aktualisierungsvorgängen der Überwachungszeitraum abläuft, ohne dass Alarme in den `ALARM` Status CloudFormation wechseln, werden die alten Ressourcen wie gewohnt

gelöscht. Wenn Sie eine Überwachungszeit festlegen, aber keine Rollback-Trigger angeben, warten Sie CloudFormation trotzdem den angegebenen Zeitraum, bevor Sie alte Ressourcen für Aktualisierungsvorgänge bereinigen. Sie können diesen Überwachungszeitraum nutzen, um etwaige manuelle Stack-Überprüfungen durchzuführen, sowie gegebenenfalls zum manuellen Abbruch der Stack-Erstellung oder des Updates. Wenn Sie eine Überwachungszeit auf 0 Minuten setzen, überwacht CloudFormation weiterhin die Rollback-Trigger während der Stack-Erstellung und Update-Operationen und setzt die Operation zurück, wenn ein Alarm in den Status `ALARM` übergeht. Wenn für Update-Operationen keine Abbruchalarme auftreten, beginnt es anschließend, alte Ressourcen sofort zu entsorgen, nachdem die Operation abgeschlossen ist.

Standardmäßig werden Stack-Operationen CloudFormation nur zurückgesetzt, wenn ein Alarm in den Status wechselt, nicht `INSUFFICIENT_DATA` in den `ALARM` Status. Um den Stack-Vorgang auch CloudFormation rückgängig zu machen, wenn ein Alarm in den `INSUFFICIENT_DATA` Status wechselt, bearbeiten Sie den CloudWatch Alarm so, dass fehlende Daten als behandelt werden`breaching`. Weitere Informationen finden Sie unter Konfiguration der Behandlung fehlender Daten durch CloudWatch Alarme im CloudWatch Amazon-Benutzerhandbuch.

CloudFormation überwacht keine Rollback-Trigger, wenn während eines Aktualisierungsvorgangs ein Stack zurückgesetzt wird.

Sie können maximal fünf Rollback-Auslöer hinzufügen. Um einen Rollback-Trigger hinzuzufügen, geben Sie den Amazon-Ressourcennamen (ARN) des CloudWatch Alarms an. Derzeit können die `AWS::CloudWatch::Alarm` und `AWS::CloudWatch::CompositeAlarm` Typen als Rollback-Auslöser verwendet werden. Weitere Informationen zu CloudWatch Alarmen finden Sie unter Verwenden von CloudWatch Alarmen im CloudWatch Amazon-Benutzerhandbuch.

Wenn ein bestimmter CloudWatch Alarm fehlt, schlägt der gesamte Stack-Vorgang fehl und wird zurückgesetzt.

Beachten Sie, dass für den Zugriff auf Anmeldeinformationen CloudWatch erforderlich sind. Diese Anmeldeinformationen müssen über Berechtigungen für den Zugriff auf AWS Ressourcen verfügen, z. B. zum Abrufen von CloudWatch Metrikdaten über Ihre Ressourcen. Weitere Informationen finden Sie unter Authentifizierung und Zugriffskontrolle für CloudWatch im CloudWatch Amazon-Benutzerhandbuch.

Fügen Sie Rollback-Trigger während der Stack-Erstellung oder Aktualisierung hinzu

Um Rollback-Trigger während der Stack-Erstellung oder Aktualisierung hinzuzufügen (Konsole)

1. Wenn Sie einen Stack erstellen oder aktualisieren, erweitern Sie auf der Seite Stack-Optionen konfigurieren unter Erweiterte Optionen den Abschnitt Rollback-Konfiguration.
2. Geben Sie eine Überwachungszeit zwischen `0` – `180` Minuten ein. Der Standardwert ist `0`.
3. Geben Sie den ARN des CloudWatch Alarms oder des zusammengesetzten Alarms an, den Sie als Rollback-Trigger verwenden möchten, und wählen Sie CloudWatch Alarm-ARN hinzufügen.

   Das Folgende ist beispielsweise ein ARN für einen CloudWatch Alarm oder einen zusammengesetzten Alarm,`arn:aws:cloudwatch:`*`us-east-1`*`:`*`123456789012`*`:alarm:`*`MyAlarmName`*.
4. Wählen Sie Weiter und überprüfen Sie die Details Ihres Stacks.
5. Wenn Sie bereit sind, wählen Sie Senden aus, um den Stack zu erstellen oder zu aktualisieren.

Um Rollback-Trigger während der Stack-Erstellung oder Aktualisierung hinzuzufügen ()AWS CLI

Verwenden Sie den update-stackBefehl create-stackoder mit der `--rollback-configuration` Option.

Der folgende update-stack Befehl dient beispielsweise *`MyCompositeAlarm`* als Rollback-Trigger mit einem Überwachungszeitraum von 5 Minuten:

```
aws cloudformation update-stack --stack-name MyStack \
  --use-previous-template \
  --rollback-configuration \
  "RollbackTriggers=[{Arn=arn:aws:cloudwatch:us-
east-1:123456789012:alarm:MyCompositeAlarm,Type=AWS::CloudWatch::CompositeAlarm}],MonitoringTim
```

Fügen Sie einem Änderungssatz Rollback-Trigger hinzu

Um Rollback-Trigger zu einem Änderungssatz hinzuzufügen (Konsole)

1. Wenn Sie einen Änderungssatz erstellen, erweitern Sie auf der Seite Stack-Optionen konfigurieren unter Erweiterte Optionen den Abschnitt Rollback-Konfiguration.

2. Geben Sie eine Überwachungszeit zwischen `0` – `180` Minuten ein. Der Standardwert ist `5`.
3. Geben Sie den ARN des CloudWatch Alarms oder des zusammengesetzten Alarms an, den Sie als Rollback-Trigger verwenden möchten, und wählen Sie CloudWatch Alarm-ARN hinzufügen.

   Das Folgende ist beispielsweise ein ARN für einen CloudWatch Alarm oder einen zusammengesetzten Alarm,`arn:aws:cloudwatch:`*`us-east-1`*`:123456789012:alarm:`*`MyAlarmName`*.
4. Wählen Sie Weiter und überprüfen Sie die Details Ihres Änderungssatzes.
5. Wenn Sie bereit sind, wählen Sie Änderungssatz erstellen aus, um den Änderungssatz zu erstellen.

Um einem Änderungssatz Rollback-Trigger hinzuzufügen ()AWS CLI

Verwenden Sie den Befehl create-change-set mit der Option `--rollback-configuration`.

Rollback-Trigger für einen Stack anzeigen

Informationen zum Anzeigen von Rollback-Auslösern für einen Stack finden Sie im Abschnitt Rollback-Konfiguration.

1. Wählen Sie auf der Seite Stacks den Stack, den Sie anzeigen möchten, aus der Liste auf der linken Seite aus.
2. Erweitern Sie auf der Registerkarte Stack-Info den Abschnitt Rollback-Konfiguration, um die Rollback-Trigger anzuzeigen.

Rollback-Trigger für einen Änderungssatz anzeigen

Informationen zum Anzeigen von Rollback-Auslöser für einen Änderungssatz finden Sie im Abschnitt Rollback configuration (Rollback-Konfiguration).

1. Wählen Sie auf der Seite Stacks den Stack, den Sie anzeigen möchten, aus der Liste auf der linken Seite aus.
2. Wählen Sie den Tab Änderungssätze und dann den Änderungssatz aus, den Sie anzeigen möchten.
3. Wählen Sie die Registerkarte Eingabe und sehen Sie sich den Abschnitt zur Rollback-Konfiguration an.

Erkennen Sie mit Drift-Erkennung nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen

Selbst wenn Sie Ihre Ressourcen verwalten CloudFormation, können Benutzer diese Ressourcen außerhalb von ändern. CloudFormation Benutzer können Ressourcen direkt bearbeiten, indem sie den zugrunde liegenden Service verwenden, der die Ressource erstellt hat. Sie können beispielsweise die EC2 Amazon-Konsole verwenden, um eine Server-Instance zu 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 CloudFormation können Änderungen, die außerhalb von vorgenommen werden, die Aktualisierung oder Löschung von Stacks erschweren. Mithilfe der Drift-Erkennung können Sie Stack-Ressourcen identifizieren, an denen außerhalb der CloudFormation Verwaltung Konfigurationsänderungen vorgenommen wurden. Sie können dann Korrekturmaßnahmen ergreifen, damit Ihre Stack-Ressourcen wieder mit ihren Definitionen in der Stack-Vorlage synchronisiert sind, z. B. die abgewichenen Ressourcen direkt aktualisieren, sodass sie mit ihrer Vorlagendefinition übereinstimmen. Die Auflösung von Abweichungen trägt dazu bei, die Konsistenz der Konfiguration und erfolgreiche Stack-Operationen zu gewährleisten.

Themen

- Was ist Abweichung?
- Statuscodes für die Abweichungserkennung
- Überlegungen zur Erkennung von Abweichungen
- Drift auf einem ganzen CloudFormation Stack erkennen
- Erkennen von Abweichungen bei einzelnen Stack-Ressourcen
- Beheben von Abweichungen mit einer Importoperation

Was ist Abweichung?

Mit der Abweichungserkennung können Sie feststellen, ob die tatsächliche Konfiguration eines Stacks von seiner erwarteten Konfiguration abweicht. Wird verwendet CloudFormation , um Abweichungen bei einem gesamten Stack oder bei einzelnen Ressourcen innerhalb des Stacks zu erkennen. Eine Ressource gilt als abgewichen, wenn einer ihrer tatsächlichen Eigenschaftswerte von den erwarteten Eigenschaftswerten abweicht. Dies gilt auch, wenn die Eigenschaft oder Ressource gelöscht wurde. Ein Stack gilt als abgewichen, wenn eine oder mehrere seiner Ressourcen abgewichen sind.

CloudFormation Ermittelt die erwarteten Eigenschaftswerte der Ressource, wie sie in der Stack-Vorlage definiert sind, und alle Werte, die als Vorlagenparameter angegeben wurden, um festzustellen, ob sich eine Ressource verändert hat. CloudFormation vergleicht dann diese erwarteten Werte mit den tatsächlichen Werten dieser Ressourceneigenschaften, so wie sie derzeit im Stapel vorhanden sind. Eine Ressource gilt als abgewichen, wenn eine oder mehrere ihrer Eigenschaften gelöscht wurden oder ihr Wert geändert wurde.

CloudFormation generiert detaillierte Informationen zu jeder Ressource im Stapel, die sich verändert hat.

CloudFormation erkennt Abweichungen bei den AWS Ressourcen, die die Drift-Erkennung unterstützen. Ressourcen, die die Abweichungserkennung nicht unterstützen, erhalten den Abweichungsstatus NOT_CHECKED. Eine Liste der AWS Ressourcen, die die Drift-Erkennung unterstützen, finden Sie unterUnterstützung für Ressourcentypen.

CloudFormation Unterstützt außerdem die Drift-Erkennung bei privaten Ressourcentypen, die bereitstellbar sind, d. h., deren Bereitstellungstyp entweder oder `FULLY_MUTABLE` ist. `IMMUTABLE` Um eine Abweichungserkennung für einen privaten Ressourcentyp ausführen zu können, muss die Standardversion des Ressourcentyps, der in Ihrem Konto registriert ist, bereitstellbar sein. Weitere Informationen zum Ressourcenbereitstellungstyp finden Sie unter dem `ProvisioningType` Parameter der DescribeTypeAktion in der AWS CloudFormation API-Referenz und des DescribeTypeBefehls in der AWS CLI Befehlsreferenz. Weitere Informationen zu privaten Ressourcen finden Sie unterErweiterungen mit der CloudFormation Registry verwalten.

Sie können die Abweichungserkennung für Stacks mit folgenden Status durchführen: `CREATE_COMPLETE`, `UPDATE_COMPLETE`, `UPDATE_ROLLBACK_COMPLETE` und `UPDATE_ROLLBACK_FAILED`.

Bei der Erkennung von Drift auf einem Stack CloudFormation wird keine Drift bei verschachtelten Stacks erkannt, die zu diesem Stapel gehören. Weitere Informationen finden Sie unter Teilen Sie eine Vorlage mithilfe verschachtelter Stapel in wiederverwendbare Teile auf. Stattdessen können Sie eine Abweichungserkennung direkt für den verschachtelten Stack durchführen.

> **Note**
>
> CloudFormation bestimmt nur die Drift für Eigenschaftswerte, die explizit festgelegt wurden, entweder über die Stack-Vorlage oder durch Angabe von Vorlagenparametern. Dies gilt nicht für Standardwerte für Ressourceneigenschaften. Damit eine Ressourceneigenschaft

zur Bestimmung der Drift CloudFormation nachverfolgt werden kann, legen Sie den Eigenschaftswert explizit fest, auch wenn Sie ihn auf den Standardwert setzen.

Statuscodes für die Abweichungserkennung

Die Tabellen in diesem Abschnitt beschreiben die verschiedenen Statusarten, die bei der Abweichungserkennung verwendet werden:

- Drift detection operation status (Abweichungserkennungs-Operationsstatus) beschreibt den aktuellen Zustand des Driftvorgangs.
- Abweichungsstatus

  Bei Stack-Sets beschreibt dies den Abweichungsstatus des Stack-Sets als Ganzes basierend auf dem Abweichungsstatus der Stack-Instances, die zu ihm gehören.

  Bei Stack-Instances beschreibt dies den Abweichungsstatus der Stack-Instance basierend auf dem Abweichungsstatus des zugehörigen Stacks.

  Bei Stacks beschreibt dies den Abweichungsstatus des Stacks als Ganzes basierend auf dem Abweichungsstatus seiner Ressourcen.
- Resource drift status (Ressourcen-Abweichungsstatus) beschreibt den Abweichungsstatus einer einzelnen Ressource.

In der folgenden Tabelle sind die Statuscodes aufgeführt, die Operationen zur Erkennung von Stack-Drift CloudFormation zugewiesen werden.

Drift Detection Operation Status (Abweichungserkennungs-Operationsstatus)	Beschreibung
`DETECTION_COMPLETE`	Der Vorgang zur Erkennung der Stack-Abweichung wurde für alle Ressourcen im Stack, die die Abweichungserkennung unterstützen, erfolgreich abgeschlossen.
`DETECTION_FAILED`	Die Stack-Abweichungserkennung ist für mindestens eine Ressource im Stack fehlgeschlagen. Die Ergebnisse werden

Drift Detection Operation Status (Abweichungserkennungs-Operationsstatus)	Beschreibung
	für Ressourcen verfügbar sein, bei denen die Drifterkennung CloudFormation erfolgreich abgeschlossen wurde.
`DETECTION_IN_PROGRESS`	Der Vorgang zur Erkennung der Stack-Abweichung ist derzeit im Gange.

In der folgenden Tabelle sind die Drift-Statuscodes aufgeführt, die CloudFormation Stacks zugewiesen werden.

Abweichungsstatus	Beschreibung
`DRIFTED`	Bei Stacks: Der Stack unterscheidet sich von der erwarteten Vorlagenkonfiguration oder ist von ihr abgewichen. Ein Stack gilt als abgewichen, wenn eine oder mehrere seiner Ressourcen abgewichen sind. Bei Stack-Instances: Eine Stack-Instance gilt als abgewichen, wenn der ihr zugeordnete Stack abgewichen ist. Bei Stack-Sets: Ein Stack-Set gilt als abgewichen, wenn eine oder mehrere Stack-Instances abgewichen sind.
`NOT_CHECKED`	CloudFormation hat nicht geprüft, ob der Stack, das Stack-Set oder die Stack-Instance von der erwarteten Vorlagenkonfiguration abweicht.
`IN_SYNC`	Die aktuelle Konfiguration jeder unterstützten Ressource entspricht der erwarteten Vorlagenkonfiguration. Ein Stack, Stack-Set oder eine Stack-Instance ohne Ressourcen, die die Abweichungserkennung unterstützen, hat ebenfalls den Status IN_SYNC.

In der folgenden Tabelle sind die Drift-Statuscodes aufgeführt, die den Stack-Ressourcen CloudFormation zugewiesen werden.

Ressourcen-Abweichungsstatus	Beschreibung
`DELETED`	Die Ressource unterscheidet sich von ihrer erwarteten Vorlagenkonfiguration, da die Ressource gelöscht wurde.
`MODIFIED`	Die Ressource unterscheidet sich von ihrer erwarteten Vorlagenkonfiguration.
`NOT_CHECKED`	CloudFormation hat nicht geprüft, ob die Ressource von der erwarteten Vorlagenkonfiguration abweicht.
`IN_SYNC`	Die aktuelle Konfiguration der Ressource entspricht der erwarteten Vorlagenkonfiguration.

In der folgenden Tabelle sind die Differenztyp-Statuscodes aufgeführt, die Ressourceneigenschaften CloudFormation zugewiesen werden, die sich von ihrer erwarteten Vorlagenkonfiguration unterscheiden.

Eigenschaftsabweichungstypen	Beschreibung
`ADD`	Ein Wert wurde zu einer Ressourceneigenschaft hinzugefügt, die ein Array- oder ein Listen-Datentyp ist.
`REMOVE`	Die Eigenschaft wurde aus der aktuellen Ressourcenkonfigur ation entfernt.
`NOT_EQUAL`	Der aktuelle Eigenschaftswert unterscheidet sich von seinem erwarteten Wert, der in der Stack-Vorlage definiert ist.

Überlegungen zur Erkennung von Abweichungen

Um die Abweichungserkennung für einen Stack erfolgreich durchführen zu können, muss ein Benutzer über die folgenden Berechtigungen verfügen:

- Leseberechtigung für jede Ressource im Stack, die die Abweichungserkennung unterstützt. Wenn der Stack beispielsweise eine `AWS::EC2::Instance`-Ressource enthält, müssen Sie die Berechtigung `ec2:DescribeInstances` haben, um die Abweichungserkennung für den Stack durchzuführen.
- `cloudformation:DetectStackDrift`
- `cloudformation:DetectStackResourceDrift`
- `cloudformation:BatchDescribeTypeConfigurations`

Weitere Informationen zum Einstellen von Berechtigungen in CloudFormation finden Sie unter. Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management

In bestimmten Ausnahmefällen CloudFormation kann es sein, dass nicht immer genaue Drift-Ergebnisse zurückgegeben werden können. Sie sollten sich dieser Sonderfälle bewusst sein, um die Ergebnisse Ihrer Abweichungserkennung richtig interpretieren zu können.

- In bestimmten Fällen werden Objekte, die in Eigenschafts-Arrays enthalten sind, als Abweichung gemeldet, während sie in Wirklichkeit Standardwerte sind, die der Eigenschaft vom zugrunde liegenden Service, der für die Ressource verantwortlich ist, zur Verfügung gestellt werden.
- Bestimmte Ressourcen haben Zuordnungsbeziehungen mit verwandten Ressourcen, sodass eine Ressource Eigenschaftswerte für eine andere Ressource hinzufügen oder entfernen könnte, die in derselben oder einer anderen Vorlage definiert sind. So können beispielsweise die Ressourcen `AWS::EC2::SecurityGroupIngress` und `AWS::EC2::SecurityGroupEgress` verwendet werden, um Werte in `AWS::EC2::SecurityGroup`-Ressourcen hinzuzufügen und zu entfernen. CloudFormation Analysiert in diesen Fällen die Stack-Vorlage auf Anlagen, bevor der Drift-Vergleich durchgeführt wird. Diese Analyse CloudFormation kann jedoch nicht stapelübergreifend durchgeführt werden und liefert daher möglicherweise keine genauen Drift-Ergebnisse, wenn sich Ressourcen, die angehängt sind, in unterschiedlichen Stapeln befinden.

  Die folgenden Ressourcen unterstützen die Abweichungserkennung und ermöglichen oder erfordern Zuordnungen aus anderen Ressourcen:

Ressourcentyp	Zuordnungsressourcentyp
`AWS::SNS::Topic`	`AWS::SNS::Subscription`
`AWS::IAM::User`	`AWS::IAM::UserToGroupAddition`

Ressourcentyp	Zuordnungsressourcentyp

`AWS::IAM::Group`
`AWS::IAM::Role`
`AWS::IAM::User`
	`AWS::IAM::ManagedPolicy`
	`AWS::IAM::Policy`
	`AWS::IAM::RolePolicy`

`AWS::ElasticLoadBalancingV2::Listener`	`AWS::ElasticLoadBalancingV2::ListenerCertificate`
`AWS::EC2::SecurityGroup`	`AWS::EC2::SecurityGroupEgress` `AWS::EC2::SecurityGroupIngress`

- CloudFormation führt keine Drift-Erkennung für das `KMSKeyId` Eigentum von Ressourcen durch. Da AWS KMS Schlüssel durch mehrere Aliase referenziert werden können, CloudFormation kann für diese Eigenschaft keine gleichbleibend genauen Drift-Ergebnisse garantiert werden.
- Es gibt bestimmte Ressourceneigenschaften, die Sie in Ihrer Stack-Vorlage angeben können und die naturgemäß nicht mit den Eigenschaften in den resultierenden Stack-Ressourcen verglichen werden können. CloudFormation Diese Eigenschaften können daher nicht in die Ergebnisse der Abweichungserkennung einbezogen werden. Solche Eigenschaften lassen sich in zwei Kategorien einteilen:
  - Eigenschaftswerte, die CloudFormation nicht ihrem ursprünglichen Ressourceneigenschaftswert in der Stackvorlage zugeordnet werden können.

    CloudFormation Kann beispielsweise den Quellcode einer Lambda-Funktion nicht dem CodeEigenschaftstyp der AWS::Lambda::FunctionRessource zuordnen und CloudFormation kann ihn daher nicht in die Ergebnisse der Drifterkennung einbeziehen.
  - Eigenschaftswerte, die der Service, der für die Ressource verantwortlich ist, nicht zurückgibt.

    Es gibt bestimmte Eigenschaftswerte, die von vornherein nie von dem Service zurückgegeben werden, zu dem die Ressource gehört. Diese enthalten in der Regel vertrauliche Informationen, wie Passwörter oder andere sensible Daten, die nicht preisgegeben werden sollten. Beispielsweise gibt der IAM-Dienst niemals den Wert der `Password` Eigenschaft des Eigenschaftstyps zurück und CloudFormation kann ihn daher nicht in die AWS::IAM::User LoginProfileErgebnisse der Drifterkennung einbeziehen.

  - Objekte in einem Array können Service-Standards sein – es muss sich nicht um eine manuell hinzugefügte Abweichung handeln.
- Wenn Sie auf falsch positive Ergebnisse stoßen, senden Sie uns Ihre Kommentare über den Feedback-Link in der CloudFormation Konsole oder kontaktieren Sie uns über AWS re:Post
- Einige Eigenschaften können Eingabewerte haben, die gleich, aber nicht identisch sind. Um Fehler zu vermeiden, sollten Sie sicherstellen, dass Ihre erwartete Konfiguration mit der tatsächlichen Konfiguration übereinstimmt.
  - Beispielsweise kann die erwartete Konfiguration der Ressourceneigenschaft 1 024 MB und die tatsächliche Konfiguration derselben Ressourceneigenschaft 1 GB betragen. 1 024 MB und 1 GB sind gleich, aber nicht identisch.

    Wenn die Abweichungserkennung für diese Ressourceneigenschaft ausgeführt wird, werden abweichende Ergebnisse signalisiert.

    Ändern Sie die erwartete Konfiguration der Ressourceneigenschaft zu 1 024 MB und führen Sie dann die Abweichungserkennung aus, um diesen Fehler zu vermeiden.

Drift auf einem ganzen CloudFormation Stack erkennen

Das Durchführen einer Abweichungserkennung für einen Stack ermittelt, ob der Stack von seiner erwarteten Vorlagenkonfiguration abgewichen ist, und gibt detaillierte Informationen über den Abweichungsstatus jeder Ressource im Stack zurück, die die Abweichungserkennung unterstützt.

Um Drift auf einem ganzen Stack zu erkennen, verwenden Sie AWS Management Console

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Liste der Stacks den Stack aus, für den Sie die Abweichungserkennung durchführen möchten. Wählen Sie im Stack-Detailbereich die Option Stack actions (Stack-Aktionen) und dann Detect drift (Abweichung erkennen) aus.

   CloudFormation zeigt eine Informationsleiste an, die angibt, dass die Drifterkennung für den ausgewählten Stack initiiert wurde.
3. Warten Sie, bis der Vorgang zur Drifterkennung CloudFormation abgeschlossen ist. Wenn die Drift-Erkennung abgeschlossen ist, werden der Drift-Status und die Uhrzeit der letzten Drift-

Prüfung für Ihren Stack CloudFormation aktualisiert. Diese Felder werden im Abschnitt Overview (Übersicht) des Bereichs Stack info (Stack-Informationen) der Stack-Detailseite aufgeführt.

Der Vorgang der Abweichungserkennung kann mehrere Minuten dauern, abhängig von der Anzahl der im Stack enthaltenen Ressourcen. Sie können nur einen einzigen Vorgang zur Drifterkennung auf einem bestimmten Stack gleichzeitig ausführen. CloudFormation setzt die Drifterkennung fort, auch wenn Sie die Informationsleiste schließen.

4. Überprüfen Sie die Ergebnisse der Abweichungserkennung für den Stack und seine Ressourcen. Wenn Sie Ihren Stack ausgewählt haben, wählen Sie im Menü Stack actions (Stack-Aktionen) die Option View drift results (Abweichungsergebnisse anzeigen) aus.

   CloudFormation listet den gesamten Drift-Status des Stacks sowie den Zeitpunkt auf, zu dem die Drift-Erkennung zuletzt auf dem Stack oder einer seiner einzelnen Ressourcen ausgelöst wurde. Ein Stack gilt als abgewichen, wenn eine oder mehrere seiner Ressourcen abgewichen sind.

   Im Abschnitt Status der Ressourcenabweichung CloudFormation werden alle Stack-Ressourcen, ihr Drift-Status und der Zeitpunkt aufgeführt, zu dem die Drift-Erkennung zuletzt auf der Ressource initiiert wurde. Die logische ID und die physische ID jeder Ressource werden angezeigt, um Ihnen bei der Identifizierung zu helfen. Bei Ressourcen mit dem Status MODIFIZIERT werden außerdem Details zur Ressourcenabweichung CloudFormation angezeigt.

   Sie können die Ressourcen basierend auf ihrem Abweichungsstatus mithilfe der Spalte Drift status (Abweichungsstatus) sortieren.

   - So zeigen Sie die Details zu einer geänderten Ressource an

     - Wenn Sie die geänderten Ressourcen ausgewählt haben, wählen Sie View drift details (Abweichungsdetails anzeigen) aus.

       CloudFormation zeigt die Drift-Detailseite für diese Ressource an. Diese Seite führt den erwarteten und aktuellen Eigenschaftswert der Ressource sowie alle Unterschiede zwischen den beiden auf.

       Zum Hervorheben eines Unterschieds wählen Sie im Abschnitt Differences (Unterschiede) den Eigenschaftsnamen aus.

       - Zusätzliche Eigenschaften werden in der Spalte Current (Aktuell) des Abschnitts Details grün hinterlegt.

- Gelöschte Eigenschaften werden in der Spalte Expected (Erwartet) des Abschnitts Details rot markiert.
- Eigenschaften, deren Wert geändert wurde, sind in den beiden Spalten Expected (Erwartet) und Current (Aktuell) gelb markiert.

Um Drift auf einem gesamten Stack zu erkennen, verwenden Sie AWS CLI

> **Important**
>
> Prüfen Sie den Zeitpunkt der letzten Drift-Prüfung für den Stack und vergewissern Sie sich, dass der Zeitpunkt vor dem in den Ergebnissen zur Ressourcenabweichung angegebenen Zeitstempel liegt, um die Verwendung veralteter Daten zu verhindern.

Verwenden Sie die folgenden AWS CLI Befehle, um Abweichungen auf einem gesamten Stack mithilfe von zu erkennen: AWS CLI

- detect-stack-drift, um einen Abweichungserkennungsvorgang für einen Stack einzuleiten.
- describe-stack-drift-detection-status, um den Status des Stapelabweichungserkennungs-Vorgangs zu überwachen.
- describe-stack-resource-drifts, um die Details der Stapelabweichungserkennung zu überprüfen.

1. Verwenden Sie die detect-stack-drift, um eine Abweichung für einen gesamten Stack zu erkennen. Geben Sie den Stack-Namen oder ARN an. Sie können auch die Logik IDs bestimmter Ressourcen angeben, die Sie als Filter für diesen Vorgang zur Drifterkennung verwenden möchten.

   ```
   aws cloudformation detect-stack-drift --stack-name my-stack-with-resource-drift
   ```

   Ausgabe:

   ```
   {
       "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample"
   }
   ```

2. Da die Erkennung der Stack-Abweichung lange dauern kann, verwenden Sie describe-stack-drift-detection-status, um den Status des Vorgangs zu überwachen. Dieser Befehl übernimmt die ID der Stack-Abweichungserkennung, die vom Befehl detect-stack-drift zurückgegeben wird.

   Im folgenden Beispiel haben wir die vom obigen Beispiel detect-stack-drift zurückgegebene Stack-Abweichungserkennungs-ID verwendet und als Parameter an describe-stack-drift-detection-status übergeben. Der Parameter gibt Vorgangsdetails zurück, die zeigen, dass der Abweichungserkennungs-Vorgang abgeschlossen ist, eine einzelne Stack-Ressource abgewichen ist und dass der gesamte Stack als Ergebnis als abgewichen gilt.

```
aws cloudformation describe-stack-drift-detection-status --stack-drift-detection-
id 624af370-311a-11e8-b6b7-500cexample
```

   Ausgabe:

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-
resource-drift/489e5570-df85-11e7-a7d9-50example",
    "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample",
    "StackDriftStatus": "DRIFTED",
    "Timestamp": "2018-03-26T17:23:22.279Z",
    "DetectionStatus": "DETECTION_COMPLETE",
    "DriftedStackResourceCount": 1
}
```

3. Wenn die Stapelabweichungserkennung abgeschlossen ist, verwenden Sie den Befehl describe-stack-resource-drifts, um die Ergebnisse zu überprüfen, einschließlich der tatsächlichen und erwarteten Eigenschaftswerte für Ressourcen, die abgewichen sind.

   Im folgenden Beispiel wird die `--stack-resource-drift-status-filters` Option verwendet, um Stack-Drift-Informationen für die Ressourcen anzufordern, die geändert oder gelöscht wurden. Die Anfrage gibt Informationen über die eine Ressource zurück, die geändert wurde, einschließlich Details über zwei ihrer Eigenschaften, deren Werte geändert wurden. Es wurden keine Ressourcen gelöscht.

```
aws cloudformation describe-stack-resource-drifts --stack-name my-stack-with-
resource-drift --stack-resource-drift-status-filters MODIFIED DELETED
```

   Ausgabe:

```
{
    "StackResourceDrifts": [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-
stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
            "ActualProperties": "{\"ReceiveMessageWaitTimeSeconds\":0,
\"DelaySeconds\":120,\"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",
\"maxReceiveCount\":12},\"MessageRetentionPeriod\":345600,\"MaximumMessageSize
\":262144,\"VisibilityTimeout\":60,\"QueueName\":\"my-stack-with-resource-drift-
Queue-494PBHCO76H4\"}",
            "ResourceType": "AWS::SQS::Queue",
            "Timestamp": "2018-03-26T17:23:34.489Z",
            "PhysicalResourceId": "https://sqs.us-
east-1.amazonaws.com/099908667365/my-stack-with-resource-drift-
Queue-494PBHCO76H4",
            "StackResourceDriftStatus": "MODIFIED",
            "ExpectedProperties": "{\"ReceiveMessageWaitTimeSeconds\":0,
\"DelaySeconds\":20,\"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",
\"maxReceiveCount\":10},\"MessageRetentionPeriod\":345600,\"MaximumMessageSize
\":262144,\"VisibilityTimeout\":60,\"QueueName\":\"my-stack-with-resource-drift-
Queue-494PBHCO76H4\"}",
            "PropertyDifferences": [
                {
                    "PropertyPath": "/DelaySeconds",
                    "ActualValue": "120",
                    "ExpectedValue": "20",
                    "DifferenceType": "NOT_EQUAL"
                },
                {
                    "PropertyPath": "/RedrivePolicy/maxReceiveCount",
                    "ActualValue": "12",
                    "ExpectedValue": "10",
                    "DifferenceType": "NOT_EQUAL"
                }
            ],
            "LogicalResourceId": "Queue"
        }
    ]
}
```

Erkennen von Abweichungen bei einzelnen Stack-Ressourcen

Sie können Abweichungen von bestimmten Ressourcen innerhalb eines Stacks statt des gesamten Stacks erkennen. Dies ist besonders nützlich, wenn Sie nur feststellen müssen, ob bestimmte Ressourcen wieder mit ihren erwarteten Vorlagenkonfigurationen übereinstimmen.

Bei der Drifterkennung für eine Ressource werden CloudFormation auch der gesamte Stack-Driftstatus und gegebenenfalls die Uhrzeit der letzten Drift-Prüfung aktualisiert. Nehmen wir zum Beispiel an, ein Stack hat den Driftstatus`IN_SYNC`. Sie haben eine Drift-Erkennung für eine oder mehrere Ressourcen CloudFormation durchgeführt, die in diesem Stack enthalten sind, und stellen CloudFormation fest, dass eine oder mehrere dieser Ressourcen driftet sind. CloudFormation aktualisiert den Status der Stack-Drift auf`DRIFTED`. Nehmen wir umgekehrt an, Sie haben einen Stack mit einem Drift-Status von, der auf eine einzelne gedriftete Ressource `DRIFTED` zurückzuführen ist. Wenn Sie diese Ressource auf ihre erwarteten Eigenschaftswerte zurücksetzen und dann erneut eine Drift bei der Ressource feststellen, CloudFormation werden sowohl der Status der Ressourcenabweichung als auch der Stack-Drift-Status auf aktualisiert, `IN_SYNC` ohne dass Sie erneut eine Drift im gesamten Stapel feststellen müssen.

Um Abweichungen bei einer einzelnen Ressource zu erkennen, verwenden Sie den AWS Management Console

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie aus der Liste der Stacks den Stack aus, der die Ressource enthält. CloudFormation zeigt die Stack-Details für diesen Stack an.
3. Wählen Sie im linken Navigationsbereich unter Stacks die Option Stack Actions (Stack-Aktionen) und dann Detect drift (Abweichung erkennen) aus.
4. Wählen Sie unter Resource drift status (Ressourcenabweichungsstatus) die Ressource aus und wählen Sie dann Detect drift for resource (Abweichung für Ressource erkennen) aus.

   CloudFormation führt eine Drift-Erkennung für die ausgewählte Ressource durch. Bei Erfolg werden der Drift-Status der Ressource und, falls erforderlich, der gesamte Stack-Driftstatus CloudFormation aktualisiert. CloudFormation aktualisiert außerdem den Zeitstempel für den Zeitpunkt, zu dem die Drift-Erkennung zuletzt für die Ressource und den gesamten Stack durchgeführt wurde. Wenn die Ressource geändert wurde, CloudFormation werden detaillierte Drift-Informationen zu den erwarteten und aktuellen Eigenschaftswerten der Ressource angezeigt.

5. Überprüfen Sie die Ergebnisse der Abweichungserkennung für die Ressource.

   - So zeigen Sie die Details zu einer geänderten Ressource an

     - Wenn Sie die geänderten Ressourcen ausgewählt haben, wählen Sie View drift details (Abweichungsdetails anzeigen) aus.

       CloudFormation zeigt die Drift-Details für diese Ressource an, einschließlich der erwarteten und aktuellen Eigenschaftswerte der Ressource sowie aller Unterschiede zwischen den beiden.

       Zum Hervorheben eines Unterschieds wählen Sie im Abschnitt Differences (Unterschiede) den Eigenschaftsnamen aus.

       - Zusätzliche Eigenschaften werden in der Spalte Current (Aktuell) des Abschnitts Details grün hinterlegt.
       - Gelöschte Eigenschaften werden in der Spalte Expected (Erwartet) des Abschnitts Details rot markiert.
       - Eigenschaften, deren Wert geändert wurde, sind in den beiden Spalten Expected (Erwartet) und Current (Aktuell) gelb markiert.

Um Abweichungen bei einer einzelnen Ressource zu erkennen, verwenden Sie den AWS CLI

- > **Important**
  > Prüfen Sie den Zeitpunkt der letzten Drift-Prüfung für die Stack-Ressource und vergewissern Sie sich, dass dieser Zeitpunkt vor dem in den Ergebnissen zur Ressourcenabweichung angegebenen Zeitstempel liegt, um die Verwendung veralteter Daten zu verhindern.

  Verwenden Sie den Befehl, um Abweichungen bei einer einzelnen Ressource mithilfe von zu erkennen. AWS CLIdetect-stack-resource-drift Geben Sie die logische ID der Ressource sowie den Stack an, in dem sie enthalten ist.

Im folgenden Beispiel wird eine Drift-Erkennungsoperation für eine bestimmte Stack-Ressource ausgeführt,`my-drifted-resource`. Die Antwort gibt Informationen zurück, die bestätigen, dass die Ressource geändert wurde, einschließlich Details zu zwei ihrer Eigenschaften, deren Werte geändert wurden.

```
aws cloudformation detect-stack-resource-drift \
    --stack-name my-stack-with-resource-drift \
    --logical-resource-id my-drifted-resource
```

Ausgabe:

```
{
    "StackResourceDrift": {
        "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-
with-resource-drift/489e5570-df85-11e7-a7d9-50example",
        "ActualProperties": "{\"ReceiveMessageWaitTimeSeconds\":0,\"DelaySeconds
\":120,\"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",
\"maxReceiveCount\":12},\"MessageRetentionPeriod\":345600,\"MaximumMessageSize
\":262144,\"VisibilityTimeout\":60,\"QueueName\":\"my-stack-with-resource-drift-
Queue-494PBHCO76H4\"}",
        "ResourceType": "AWS::SQS::Queue",
        "Timestamp": "2018-03-26T18:54:28.462Z",
        "PhysicalResourceId": "https://sqs.us-east-1.amazonaws.com/099908667365/my-
stack-with-resource-drift-Queue-494PBHCO76H4",
        "StackResourceDriftStatus": "MODIFIED",
        "ExpectedProperties": "{\"ReceiveMessageWaitTimeSeconds\":0,
\"DelaySeconds\":20,\"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",
\"maxReceiveCount\":10},\"MessageRetentionPeriod\":345600,\"MaximumMessageSize
\":262144,\"VisibilityTimeout\":60,\"QueueName\":\"my-stack-with-resource-drift-
Queue-494PBHCO76H4\"}",
        "PropertyDifferences": [
            {
                "PropertyPath": "/DelaySeconds",
                "ActualValue": "120",
                "ExpectedValue": "20",
                "DifferenceType": "NOT_EQUAL"
            },
            {
                "PropertyPath": "/RedrivePolicy/maxReceiveCount",
```

```
                "ActualValue": "12",
                "ExpectedValue": "10",
                "DifferenceType": "NOT_EQUAL"
            }
        ],
        "LogicalResourceId": "my-drifted-resource"
    }
}
```

Beheben von Abweichungen mit einer Importoperation

Es kann Fälle geben, in denen die Konfiguration einer Ressource von der beabsichtigten Konfiguration abgewichen ist und Sie die neue Konfiguration als vorgesehene Konfiguration akzeptieren möchten. In den meisten Fällen würden Sie die Abweichungsergebnisse beheben, indem Sie die Ressourcendefinition in der Stack-Vorlage mit einer neuen Konfiguration aktualisieren und dann eine Stack-Aktualisierung ausführen. Wenn die neue Konfiguration jedoch eine Ressourceneigenschaft aktualisiert, die ersetzt werden muss, wird die Ressource während der Stack-Aktualisierung neu erstellt. Wenn Sie die vorhandene Ressource beibehalten möchten, können Sie die Ressourcenimportfunktion verwenden, um die Ressource zu aktualisieren und die Abweichungsergebnisse zu beheben, ohne dass die Ressource ersetzt wird.

Das Beheben der Abweichung für eine Ressource durch eine Importoperation besteht aus den folgenden grundlegenden Schritten:

- Fügen Sie der Ressource ein DeletionPolicy Attribut hinzu, das auf Retain festgelegt ist. Dadurch wird sichergestellt, dass die vorhandene Ressource beibehalten und nicht gelöscht wird, wenn sie aus dem Stack entfernt wird.
- Entfernen Sie die Ressource aus der Vorlage und führen Sie eine Stack-Aktualisierungsoperation aus. Dadurch wird die Ressource aus dem Stack entfernt, aber nicht gelöscht.
- Beschreiben Sie den tatsächlichen Zustand der Ressource in der Stack-Vorlage und importieren Sie dann die vorhandene Ressource wieder in den Stack. Dadurch wird die Ressource wieder in den Stack eingefügt und die Unterschiede in den Eigenschaften behoben, die die Abweichungsergebnisse verursacht haben.

Weitere Informationen zum Ressourcenimport finden Sie unterManuelles Importieren von AWS Ressourcen in einen CloudFormation Stack. Eine Liste der Ressourcen, die den Import unterstützen, finden Sie unterUnterstützung für Ressourcentypen.

In diesem Beispiel verwenden wir die folgende Vorlage mit der Bezeichnung `templateToImport.json`.

Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
         "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "BillingMode": "PROVISIONED",
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
```

```
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
                    }
                ],
                "BillingMode": "PROVISIONED",
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        }
    }
}
```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  GamesTable:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      TableName: Games
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
```

```
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
```

Gehen wir in diesem Beispiel davon aus, dass ein Benutzer eine Ressource außerhalb von geändert hat CloudFormation. Nachdem eine Abweichungserkennung ausgeführt wurde, haben wir festgestellt, dass für `GamesTable` `BillingMode` in `PAY_PER_REQUEST` geändert wurde. Weitere Hinweise zur Drifterkennung finden Sie unterErkennen Sie mit Drift-Erkennung nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen.

Unser Stack ist jetzt veraltet, unsere Ressourcen sind live, aber wir wollen die vorgesehene Ressourcenkonfiguration beibehalten. Dies ist möglich, indem wir die Abweichung durch eine Importoperation beheben, ohne die Services zu unterbrechen.

Beheben Sie die Drift mit einem Importvorgang über die CloudFormation Konsole

Schritt 1. Aktualisieren des Stacks mit der Löschrichtlinie Retain (Beibehalten)

So aktualisieren Sie den Stack mithilfe eines **`DeletionPolicy`**-Attributs mit der Option **`Retain`**

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der Seite Stacks den Stack mit der Abweichung aus.
3. Wählen Sie Update (Aktualisieren) und dann im Bereich mit den Stackdetails die Option Replace current template (Aktuelle Vorlage ersetzen).
4. Geben Sie auf der Seite Specify template (Vorlage angeben) mit einer der folgenden Methoden die aktualisierte Vorlage an, die das `DeletionPolicy`-Attribut mit der Option `Retain` enthält:

   - Wählen Sie Amazon S3-URL aus und geben Sie dann die URL für Ihre Vorlage im Textfeld an.
   - Wählen Sie Vorlagendatei hochladen aus und suchen Sie dann nach Ihrer Vorlage.

   Wählen Sie anschließend Weiter.
5. Überprüfen Sie die Seite Specify stack details (Stackdetails angeben) und wählen Sie Next (Weiter).

6. Überprüfen Sie die Seite Configure stack options (Stackoptionen konfigurieren) und wählen Sie Next (Weiter).
7. Wählen Sie auf der *stack-name* Seite „Überprüfen“ die Option Stack aktualisieren aus.

Ergebnisse: Auf der Seite Events (Ereignisse) Ihres Stacks lautet der Status `UPDATE_COMPLETE`.

Um Abweichungen durch einen Importvorgang zu beheben, ohne die Dienste zu unterbrechen, geben Sie a `Retain` DeletionPolicyfür die Ressourcen an, die Sie aus Ihrem Stack entfernen möchten. Im folgenden Beispiel haben wir der `GamesTable` Ressource ein DeletionPolicyAttribut, set to`Retain`, hinzugefügt.

Example JSON

```
    "GamesTable": {
        "Type": "AWS::DynamoDB::Table",
        "DeletionPolicy": "Retain",
        "Properties": {
            "TableName": "Games",
```

Example YAML

```
  GamesTable:
    Type: 'AWS::DynamoDB::Table'
    DeletionPolicy: Retain
    Properties:
      TableName: Games
```

Schritt 2. Entfernen von abgewichenen Ressourcen, zugehörigen Parametern und Ausgaben

So entfernen Sie abgewichene Ressourcen, zugehörige Parameter und Ausgaben

1. Wählen Sie Update (Aktualisieren) und dann im Bereich mit den Stackdetails die Option Replace current template (Aktuelle Vorlage ersetzen).
2. Geben Sie mit einer der folgenden Methoden auf der Seite Specify template (Vorlage angeben) Ihre aktualisierte Vorlage mit ihren Ressourcen, zugehörigen Parametern und Ausgaben an, die aus der Stack-Vorlage entfernt wurden:
   - Wählen Sie Amazon S3-URL aus und geben Sie dann die URL für Ihre Vorlage im Textfeld an.

- Wählen Sie Vorlagendatei hochladen aus und suchen Sie dann nach Ihrer Vorlage.

  Wählen Sie anschließend Weiter.

3. Überprüfen Sie die Seite Specify stack details (Stackdetails angeben) und wählen Sie Next (Weiter).
4. Überprüfen Sie die Seite Configure stack options (Stackoptionen konfigurieren) und wählen Sie Next (Weiter).
5. Wählen Sie auf der *stack-name* Seite „Überprüfen“ die Option Stack aktualisieren aus.

Ergebnisse: Die Logical ID (Logische ID) `GamesTable` hat den Status `DELETE_SKIPPED` auf der Seite Events (Ereignisse) Ihres Stacks.

Warten Sie, bis der Stack-Aktualisierungsvorgang CloudFormation abgeschlossen ist. Entfernen Sie nach Abschluss der Stack-Aktualisierungsoperation die Ressource, die zugehörigen Parameter und Ausgaben aus der Stack-Vorlage. Importieren Sie dann die aktualisierte Vorlage. Nach Abschluss dieser Aktionen sieht die Beispielvorlage jetzt wie folgt aus.

Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
         "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
```

```
                "BillingMode": "PROVISIONED",
                "ProvisionedThroughput":{
                   "ReadCapacityUnits":5,
                   "WriteCapacityUnits":1
                }
            }
        }
    }
}
```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
```

Schritt 3. Aktualisieren der Vorlage, damit Sie dem Live-Status Ihrer Ressourcen entspricht

So aktualisieren Sie die Vorlage, damit sie dem Live-Status von Ressourcen entspricht

1. Um die aktualisierte Vorlage zu importieren, wählen Sie Stack actions (Stack-Aktionen) und dann Import resources into stack (Ressourcen in Stack importieren).

2. Prüfen Sie auf der Seite Import overview (Importübersicht) die Liste der Dinge, die Sie während dieser Operation angeben müssen, und wählen Sie dann Next (Weiter).

3. Geben Sie auf der Seite Specify template (Vorlage angeben) mit einer der folgenden Methoden Ihre aktualisierte Vorlage an:

   - Wählen Sie Amazon S3-URL aus und geben Sie dann die URL für Ihre Vorlage im Textfeld an.
   - Wählen Sie Vorlagendatei hochladen aus und suchen Sie dann nach Ihrer Vorlage.

   Wählen Sie anschließend Weiter.

4. Identifizieren Sie auf der Seite Ressourcen identifizieren jede Zielressource. Weitere Informationen finden Sie unter Ressourcen-Identifikatoren.

   a. Wählen Sie unter Bezeichnereigenschaft den Typ des Ressourcenbezeichners aus. Die Eigenschaft `TableName` identifiziert beispielsweise die Ressource `AWS::DynamoDB::Table`.

   b. Geben Sie unter Identifier value (Bezeichnerwert) den tatsächlichen Eigenschaftswert ein. In der Beispielvorlage lautet der `TableName` für die `GamesTable`-Ressource `Games`.

   c. Wählen Sie Weiter aus.

5. Überprüfen Sie die Seite Specify stack details (Stack-Details angeben) und wählen Sie Next (Weiter).

6. Überprüfen Sie auf der Seite Import overview (Importübersicht), welche Ressourcen importiert werden, und wählen Sie dann Import resources (Ressourcen importieren). Dadurch wird der Ressourcentyp `AWS::DynamoDB::Table` zurück in Ihren Stack importiert.

Ergebnisse: In diesem Beispiel haben wir die Ressourcenabweichung durch eine Importoperation behoben, ohne Services zu unterbrechen. Sie können den Fortschritt einer Importaktion in der CloudFormation Konsole auf der Registerkarte Ereignisse überprüfen. Importierte Ressourcen weisen den Status `IMPORT_COMPLETE` auf, gefolgt vom Status `CREATE_COMPLETE` mit dem Statusgrund Resource import complete (Ressourcenimport abgeschlossen).

Warten Sie, bis der Stack-Aktualisierungsvorgang CloudFormation abgeschlossen ist. Aktualisieren Sie Ihre Vorlage nach Abschluss der Stack-Aktualisierungsoperation so, dass sie dem tatsächlichen, abgewichenen Zustand Ihrer Ressourcen entspricht. Beispiel: `BillingMode` wird auf `PAY_PER_REQUEST` festgelegt und `ReadCapacityUnits` sowie `WriteCapacityUnits` werden auf `0` festgelegt.

Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
         "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "BillingMode": "PROVISIONED",
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
```

```
                    }
                ],
                "BillingMode": "PAY_PER_REQUEST",
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 0,
                    "WriteCapacityUnits": 0
                }
            }
        }
    }
}
```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  GamesTable:
    Type: 'AWS::DynamoDB::Table'
    DeletionPolicy: Retain
    Properties:
      TableName: Games
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PAY_PER_REQUEST
```

```
      ProvisionedThroughput:
        ReadCapacityUnits: 0
        WriteCapacityUnits: 0
```

AWS Ressourcen in einen CloudFormation Stack importieren

Sie können vorhandene Ressourcen in einen CloudFormation Stapel importieren. Dies ist nützlich, wenn Sie damit beginnen möchten, Ressourcen CloudFormation zu verwalten, die außerhalb von erstellt wurden CloudFormation, ohne sie löschen und neu erstellen zu müssen.

CloudFormation bietet die folgenden Optionen für den Import vorhandener Ressourcen in einen Stack:

- Der IaC-Generator ist ein Tool, das Ihre vorhandenen Ressourcen automatisch scannt und auf der Grundlage ihres aktuellen Status eine CloudFormation Vorlage generiert. Diese Vorlage kann dann verwendet werden, um diese Ressourcen in einen Stapel zu importieren.
- Der Ressourcenimport ist ein manueller Prozess, bei dem Sie die vorhandenen Ressourcen in Ihrer CloudFormation Vorlage beschreiben und sie dann in einen Stapel importieren. Bei diesem Ansatz müssen Sie die Ressourceneigenschaften und Konfigurationen in der Vorlage manuell angeben.
- Der automatische Import ist ein automatischer Prozess, bei dem Sie die vorhandenen Ressourcen in Ihrer CloudFormation Vorlage beschreiben und die Ressourcen mit den passenden benutzerdefinierten Namen in einen Stapel CloudFormation importieren.
- Das Stack-Refactoring ist eine Funktion, die die Reorganisation der Ressourcen in Ihren CloudFormation Stacks vereinfacht und gleichzeitig die vorhandenen Ressourceneigenschaften und -daten beibehält. Mit Stack-Refactoring können Sie Ressourcen zwischen Stacks verschieben, monolithische Stacks in kleinere Komponenten aufteilen oder mehrere Stacks zu einem zusammenfassen.

Die Ressourcenimportfunktion ermöglicht nicht nur die CloudFormation Verwaltung vorhandener Ressourcen, sondern kann auch in den folgenden Szenarien nützlich sein:

- Verschieben von Ressourcen zwischen Stacks — Sie können Ressourcen von einem Stapel in einen anderen importieren, sodass Sie Ihre Infrastruktur nach Bedarf neu organisieren können.
- Verschachtelung vorhandener Stacks — Sie können einen vorhandenen Stack als verschachtelten Stack innerhalb eines anderen Stacks importieren, was modulare und wiederverwendbare Infrastrukturdesigns ermöglicht.

CloudFormation unterstützt den Import einer Vielzahl von Ressourcen. Weitere Informationen finden Sie unter Unterstützung für Ressourcentypen.

Themen

- Manuelles Importieren von AWS Ressourcen in einen CloudFormation Stack
- Automatisches Importieren von AWS Ressourcen in einen CloudFormation Stack
- Wiederherstellen eines Importvorgangs

Manuelles Importieren von AWS Ressourcen in einen CloudFormation Stack

Mit dem Ressourcenimport können Sie vorhandene AWS Ressourcen in einen neuen oder vorhandenen CloudFormation Stapel importieren. Während eines Importvorgangs erstellen Sie einen Änderungssatz, der Ihre vorhandenen Ressourcen in einen Stack importiert oder einen neuen Stack aus Ihren vorhandenen Ressourcen erstellt. Während des Imports geben Sie Folgendes an.

- Eine Vorlage, die den gesamten Stack beschreibt, einschließlich der ursprünglichen Stack-Ressourcen und der zu importierenden Ressourcen. Jede zu importierende Ressource muss über ein DeletionPolicyAttribut verfügen.
- Identifikatoren für die Ressourcen, die Sie importieren, mit denen Sie die logischen Elemente IDs in der Vorlage den vorhandenen Ressourcen zuordnen CloudFormation können.

> **Note**
>
> CloudFormation unterstützt mithilfe des Ressourcenimports nur eine Verschachtelungsebene. Dies bedeutet, dass Sie keinen Stack in einen untergeordneten Stack importieren oder einen Stack importieren können, der untergeordnete Stacks enthält.

Themen

- Ressourcen-Identifikatoren
- Ressourcenimportvalidierung
- Statuscodes für den Ressourcenimport
- Überlegungen während eines Importvorgangs
- Weitere Ressourcen

- Erstellen eines Stacks aus vorhandenen Ressourcen
- Importieren vorhandener Ressourcen in einen Stack
- Verschieben von Ressourcen zwischen Stacks
- Verschachteln eines vorhandenen Stacks

Ressourcen-Identifikatoren

Sie geben zwei Werte an, um jede Ressource zu identifizieren, die Sie importieren.

- Eine Bezeichnereigenschaft. Dies ist eine Ressourceneigenschaft, die verwendet werden kann, um jeden Ressourcentyp zu identifizieren. Beispielsweise kann eine `AWS::S3::Bucket`-Ressource mit ihrem `BucketName` identifiziert werden.

  Die Ressourceneigenschaft, mit der Sie die Ressource identifizieren, die Sie importieren, hängt vom Ressourcentyp ab. Sie finden die Ressourceneigenschaft in der CloudFormation Konsole. Nachdem Sie eine Vorlage erstellt haben, die die zu importierende Ressource enthält, können Sie den Importvorgang starten, in dem Sie die Kennungseigenschaften für die Ressourcen finden, die Sie importieren. Für einige Ressourcentypen gibt es möglicherweise mehrere Möglichkeiten, sie zu identifizieren, und Sie können in den Dropdownlisten auswählen, welche Eigenschaft verwendet werden soll.

  Alternativ können Sie die Identifier-Eigenschaften für die Ressourcen, die Sie importieren, abrufen, indem Sie den get-template-summaryCLI-Befehl aufrufen und die S3-URL der Stack-Vorlage als Wert für die `--template-url` Option angeben.
- Ein Bezeichnerwert. Dies ist der tatsächliche Eigenschaftswert der Ressource. Beispielsweise könnte der tatsächliche Wert für die `BucketName`-Eigenschaft `MyS3Bucket` sein.

  Sie können den Wert der Identifier-Eigenschaft von der Servicekonsole für die Ressource abrufen.

Ressourcenimportvalidierung

CloudFormation Führt während eines Importvorgangs die folgenden Validierungen durch.

- Die zu importierende Ressource ist vorhanden.
- Die Eigenschaften und Konfigurationswerte für jede zu importierende Ressource entsprechen dem Ressourcentypschema, das die akzeptierten Eigenschaften, erforderlichen Eigenschaften und unterstützten Eigenschaftswerte definiert.

- Die erforderlichen Eigenschaften werden in der Vorlage angegeben. Die erforderlichen Eigenschaften für jeden Ressourcentyp werden in der Referenz zu AWS Ressourcen- und Eigenschaftstypen beschrieben.
- Die zu importierende Ressource gehört nicht zu einem anderen Stack in derselben Region.

CloudFormation überprüft nicht, ob die Vorlagenkonfiguration mit der tatsächlichen Konfiguration der Ressourceneigenschaften übereinstimmt.

> **Important**
>
> Stellen Sie sicher, dass die in der Vorlage definierten Ressourcen und ihre Eigenschaften mit der beabsichtigten Konfiguration des Ressourcen-Imports übereinstimmen, um unerwartete Änderungen zu vermeiden.

Statuscodes für den Ressourcenimport

In dieser Tabelle werden die verschiedenen Statustypen beschrieben, die mit der Ressourcenimportfunktion verwendet werden.

Status des Importvorgangs	Beschreibung
`IMPORT_IN_PROGRESS`	Der Importvorgang wird ausgeführt.
`IMPORT_COMPLETE`	Der Importvorgang wurde für alle Ressourcen im Stack abgeschlossen.
`IMPORT_ROLLBACK_IN_PROGRESS`	Der Rollback-Importvorgang wird zur vorherigen Vorlagenkonfiguration zurückgesetzt.
`IMPORT_ROLLBACK_FAILED`	Der Import-Rollback-Vorgang ist fehlgeschlagen.
`IMPORT_ROLLBACK_COMPLETE`	Der Import wird zur vorherigen Vorlagenkonfiguration zurückgesetzt.

Überlegungen während eines Importvorgangs

- Nachdem der Import abgeschlossen ist und bevor nachfolgende Stack-Vorgänge ausgeführt werden, empfehlen wir, die Abweichungserkennung für importierte Ressourcen auszuführen. Die Abweichungserkennung stellt sicher, dass die Vorlagenkonfiguration mit der tatsächlichen Konfiguration übereinstimmt. Weitere Informationen finden Sie unter Drift auf einem ganzen CloudFormation Stack erkennen.
- Importvorgänge lassen keine neuen Ressourcenerstellungen, Ressourcenlöschungen oder Änderungen an Eigenschaftskonfigurationen zu.
- Jede zu importierende Ressource muss über ein `DeletionPolicy`-Attribut verfügen, damit der Importvorgang erfolgreich ausgeführt werden kann. Die `DeletionPolicy` kann auf einen beliebigen Wert gesetzt werden. Nur die Ressourcen, die Sie importieren, benötigen ein`DeletionPolicy`. Ressourcen, die bereits Teil des Stacks sind, benötigen keine `DeletionPolicy`.
- Sie können die gleiche Ressource nicht in mehrere Stacks importieren.
- Mit der `cloudformation:ImportResourceTypes`-IAM-Richtlinienbedingung können Sie steuern, mit welchen Ressourcentypen Benutzer während eines Importvorgangs arbeiten können. Weitere Informationen finden Sie unter Schlüssel für die Richtlinienbedingung für CloudFormation.
- Die CloudFormation Stack-Limits gelten beim Import von Ressourcen. Weitere Informationen zu Grenzwerten finden Sie unter CloudFormation Kontingente verstehen.

Weitere Ressourcen

Informationen zur Behebung von Stack-Drift bei einem Ressourcenimport finden Sie unterBeheben von Abweichungen mit einer Importoperation.

Erstellen eines Stacks aus vorhandenen Ressourcen

In diesem Thema erfahren Sie, wie Sie einen Stack aus vorhandenen AWS Ressourcen erstellen, indem Sie diese in einer Vorlage beschreiben. Informationen dazu, wie Sie stattdessen nach vorhandenen Ressourcen suchen und automatisch eine Vorlage generieren können, mit der Sie vorhandene Ressourcen in ein neues Konto importieren CloudFormation oder Ressourcen in einem neuen Konto replizieren können, finden Sie unterGenerieren Sie Vorlagen aus vorhandenen Ressourcen mit dem IaC-Generator.

Voraussetzungen

Sie benötigen Folgendes, um starten zu können:

- Eine Vorlage, die alle Ressourcen beschreibt, die Sie im neuen Stapel haben möchten. Speichern Sie die Vorlage lokal oder in einem Amazon S3 S3-Bucket.
- Geben Sie für jede Ressource, die Sie importieren möchten, Folgendes an:
  - die Eigenschaften und Eigenschaftswerte, die die aktuelle Konfiguration der Ressource definieren.
  - der eindeutige Bezeichner für die Ressource, z. B. der Ressourcenname. Weitere Informationen finden Sie unter Ressourcen-Identifikatoren.
  - das DeletionPolicyAttribut.

Themen

- -Beispielvorlage
- Erstellen Sie einen Stack aus vorhandenen Ressourcen mithilfe des AWS Management Console
- Erstellen eines Stacks aus vorhandenen Ressourcen mithilfe der AWS CLI

-Beispielvorlage

In dieser exemplarischen Vorgehensweise gehen wir davon aus, dass Sie die folgende Beispielvorlage namens verwenden, die zwei DynamoDB-Tabellen angibt`TemplateToImport.json`, die außerhalb von erstellt wurden. CloudFormation `ServiceTable`und `GamesTable` sind die Ziele des Imports.

> **Note**
>
> Diese Vorlage dient nur als Beispiel. Um sie für Ihre eigenen Testzwecke zu verwenden, ersetzen Sie die Beispielressourcen durch Ressourcen aus Ihrem Konto.

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
```

```
                "TableName": "Service",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        }
    }
```

```
}
```

Erstellen Sie einen Stack aus vorhandenen Ressourcen mithilfe des AWS Management Console

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie auf der Seite Stacks die Option Stack erstellen aus und klicken Sie dann auf Mit vorhandenen Ressourcen (Ressourcen importieren).
3. Auf der Seite Importübersicht finden Sie eine Liste der Dinge, die Sie während dieses Vorgangs angeben müssen. Wählen Sie anschließend Weiter.
4. Geben Sie auf der Seite Vorlage angeben Ihre Vorlage mit einer der folgenden Methoden an und wählen Sie dann Weiter aus.
   - Wählen Sie Amazon S3-URL aus und geben Sie dann die URL für Ihre Vorlage im Textfeld an.
   - Wählen Sie Vorlagendatei hochladen aus und suchen Sie dann nach Ihrer Vorlage.
5. Identifizieren Sie auf der Seite Ressourcen identifizieren jede Zielressource. Weitere Informationen finden Sie unter Ressourcen-Identifikatoren.
   a. Wählen Sie unter Bezeichnereigenschaft den Typ des Ressourcenbezeichners aus. Beispielsweise kann die `AWS::DynamoDB::Table`-Ressource mit der `TableName`-Eigenschaft identifiziert werden.
   b. Geben Sie unter Bezeichnerwert den tatsächlichen Eigenschaftswert ein. Beispielsweise ist der `TableName` für die `GamesTable`-Ressource in der Beispielvorlage *`Games`*.
   c. Wählen Sie Weiter aus.
6. Ändern Sie auf der Seite Stack-Details angeben alle Parameter und wählen Sie dann Weiter aus. Dadurch wird automatisch ein Änderungssatz erstellt.

   > **Important**
   > Der Importvorgang schlägt fehl, wenn Sie vorhandene Parameter ändern, die einen Erstellungs-, Aktualisierungs- oder Löschvorgang initiieren.

7. Vergewissern Sie sich auf der ***`stack-name`***Überprüfungsseite, dass die richtigen Ressourcen importiert wurden, und wählen Sie dann Ressourcen importieren aus. Dadurch wird der im letzten Schritt erstellte Änderungssatz automatisch ausgeführt.

Der Bereich Ereignisse der Seite Stack-Details für Ihren neuen Stack wird angezeigt.

8. (Optional) Führen Sie die Abweichungserkennung für den Stack aus, um sicherzustellen, dass die Vorlage und die tatsächliche Konfiguration der importierten Ressourcen übereinstimmen. Weitere Informationen zur Erkennung von Abweichungen finden Sie unterDrift auf einem ganzen CloudFormation Stack erkennen.
9. (Optional) Wenn die importierten Ressourcen nicht mit den erwarteten Vorlagenkonfigurationen übereinstimmen, korrigieren Sie entweder die Vorlagenkonfigurationen oder aktualisieren Sie die Ressourcen direkt. In dieser Anleitung korrigieren wir die Vorlagenkonfigurationen so, dass sie ihren tatsächlichen Konfigurationen entsprechen.

   a. Stellen Sie den Importvorgang für die betroffenen Ressourcen wieder her.

   b. Fügen Sie der Vorlage die Importziele erneut hinzu und stellen Sie sicher, dass die Vorlagenkonfigurationen mit den tatsächlichen Konfigurationen übereinstimmen.

   c. Wiederholen Sie die Schritte 2–8 mit der geänderten Vorlage, um die Ressourcen erneut zu importieren.

Erstellen eines Stacks aus vorhandenen Ressourcen mithilfe der AWS CLI

1. Um zu erfahren, welche Eigenschaften die einzelnen Ressourcentypen in der Vorlage identifizieren, führen Sie den get-template-summary Befehl aus und geben Sie dabei die S3-URL der Vorlage an. Beispielsweise kann die `AWS::DynamoDB::Table`-Ressource mit der `TableName`-Eigenschaft identifiziert werden. Für die `GamesTable`-Ressource in der Beispielvorlage lautet der Wert von `TableName` `Games`. Sie benötigen diese Informationen im nächsten Schritt.

```
aws cloudformation get-template-summary \
    --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
TemplateToImport.json
```

   Weitere Informationen finden Sie unter Ressourcen-Identifikatoren.

2. Erstellen Sie eine Liste der tatsächlichen Ressourcen aus Ihrer Vorlage und ihrer eindeutigen Identifikatoren im folgenden JSON-Zeichenkettenformat.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifie
{"TableName":"Games"}},
```

```
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifi
{"TableName":"Service"}}]
```

Alternativ können Sie die JSON-formatierten Parameter in einer Konfigurationsdatei angeben.

Um beispielsweise zu importieren `ServiceTable` und`GamesTable`, könnten Sie eine *`ResourcesToImport.txt`* Datei erstellen, die die folgende Konfiguration enthält.

```
[
   {
      "ResourceType":"AWS::DynamoDB::Table",
      "LogicalResourceId":"GamesTable",
      "ResourceIdentifier":{
         "TableName":"Games"
      }
   },
   {
      "ResourceType":"AWS::DynamoDB::Table",
      "LogicalResourceId":"ServiceTable",
      "ResourceIdentifier":{
         "TableName":"Service"
      }
   }
]
```

3. Um einen Änderungssatz zu erstellen, verwenden Sie den folgenden create-change-set Befehl und ersetzen Sie den Platzhaltertext. Geben Sie für die Option `--change-set-type` einen Wert von **`IMPORT`** an. Ersetzen Sie für `--resources-to-import` diese Option die JSON-Beispielzeichenfolge durch die tatsächliche JSON-Zeichenfolge, die Sie gerade erstellt haben.

```
aws cloudformation create-change-set \
    --stack-name TargetStack --change-set-name ImportChangeSet \
    --change-set-type IMPORT \
    --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
TemplateToImport.json \
    --resources-to-
import '[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceI
{"TableName":"Games"}},
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifi
{"TableName":"Service"}}]'
```

> **Note**
>
> `--resources-to-import`unterstützt Inline-YAML nicht. Die Anforderungen für das Umgehen von Anführungszeichen in der JSON-Zeichenfolge variieren je nach Terminal. Weitere Informationen finden Sie im Benutzerhandbuch unter Verwenden von Anführungszeichen innerhalb von Zeichenketten.AWS Command Line Interface

Alternativ können Sie eine Datei-URL als Eingabe für die `--resources-to-import` Option verwenden, wie im folgenden Beispiel gezeigt.

```
--resources-to-import file://ResourcesToImport.txt
```

4. Überprüfen Sie den Änderungssatz, um sicherzustellen, dass die richtigen Ressourcen importiert werden.

```
aws cloudformation describe-change-set \
    --change-set-name ImportChangeSet --stack-name TargetStack
```

5. Verwenden Sie den folgenden execute-change-set Befehl und ersetzen Sie den Platzhaltertext, um den Änderungssatz zu initiieren und die Ressourcen zu importieren. Nach erfolgreichem Abschluss des Vorgangs (`IMPORT_COMPLETE`) werden die Ressourcen erfolgreich importiert.

```
aws cloudformation execute-change-set \
    --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Optional) Führen Sie die Abweichungserkennung für den `IMPORT_COMPLETE`-Stack aus, um sicherzustellen, dass die Vorlage und die tatsächliche Konfiguration der importierten Ressourcen übereinstimmen. Weitere Informationen zur Erkennung von Abweichungen finden Sie unterErkennen von Abweichungen bei einzelnen Stack-Ressourcen.

   a. Führt die Drifterkennung für den angegebenen Stack aus.

   ```
   aws cloudformation detect-stack-drift --stack-name TargetStack
   ```

   Bei Erfolg gibt dieser Befehl die folgende Beispielausgabe zurück.

   ```
   { "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
   ```

b. Zeigt den Fortschritt eines Drift-Erkennungsvorgangs für die angegebene Stack-Drift-Erkennungs-ID an.

```
aws cloudformation describe-stack-drift-detection-status \
    --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

c. Zeigt Drift-Informationen für die Ressourcen an, die im angegebenen Stack auf Drift überprüft wurden.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Optional) Wenn die importierten Ressourcen nicht mit den erwarteten Vorlagenkonfigurationen übereinstimmen, korrigieren Sie entweder die Vorlagenkonfigurationen oder aktualisieren Sie die Ressourcen direkt. In dieser Anleitung korrigieren wir die Vorlagenkonfigurationen so, dass sie ihren tatsächlichen Konfigurationen entsprechen.

   a. Stellen Sie den Importvorgang für die betroffenen Ressourcen wieder her.

   b. Fügen Sie der Vorlage die Importziele erneut hinzu und stellen Sie sicher, dass die Vorlagenkonfigurationen mit den tatsächlichen Konfigurationen übereinstimmen.

   c. Wiederholen Sie die Schritte 3 bis 6 mit der geänderten Vorlage, um die Ressourcen erneut zu importieren.

Importieren vorhandener Ressourcen in einen Stack

In diesem Thema erfahren Sie, wie Sie vorhandene AWS Ressourcen in einen vorhandenen Stapel importieren, indem Sie sie in einer Vorlage beschreiben. Informationen dazu, wie Sie stattdessen nach vorhandenen Ressourcen suchen und automatisch eine Vorlage generieren können, mit der Sie vorhandene Ressourcen in ein neues Konto importieren CloudFormation oder Ressourcen in einem neuen Konto replizieren können, finden Sie unterGenerieren Sie Vorlagen aus vorhandenen Ressourcen mit dem IaC-Generator.

Voraussetzungen

Sie benötigen Folgendes, um starten zu können:

- Eine Vorlage, die den gesamten Stack beschreibt, einschließlich der Ressourcen, die bereits Teil des Stacks sind, und der zu importierenden Ressourcen. Speichern Sie die Vorlage lokal oder in einem Amazon S3 S3-Bucket.

Um eine Kopie der Vorlage eines laufenden Stacks zu erhalten

1. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation/.
2. Wählen Sie aus der Liste der Stacks den Stapel aus, aus dem Sie die Vorlage abrufen möchten.
3. Wählen Sie im Bereich mit den Stack-Details die Registerkarte Vorlage und dann In die Zwischenablage kopieren aus.
4. Fügen Sie den Code in einen Texteditor ein, um weitere Ressourcen zur Vorlage hinzuzufügen.

- Fügen Sie für jede Ressource, die Sie importieren möchten, Folgendes hinzu:
  - die Eigenschaften und Eigenschaftswerte, die die aktuelle Konfiguration der Ressource definieren.
  - der eindeutige Bezeichner für die Ressource, z. B. der Ressourcenname. Weitere Informationen finden Sie unter Ressourcen-Identifikatoren.
  - das DeletionPolicyAttribut.

Themen

- -Beispielvorlage
- Importieren Sie eine vorhandene Ressource in einen Stack mit dem AWS Management Console
- Importieren Sie eine vorhandene Ressource in einen Stapel mithilfe der AWS CLI

-Beispielvorlage

In dieser exemplarischen Vorgehensweise gehen wir davon aus, dass Sie die folgende Beispielvorlage namens `TemplateToImport.json` verwenden, die zwei DynamoDB-Tabellen spezifiziert. `ServiceTable`ist derzeit Teil des Stacks und `GamesTable` ist die Tabelle, die Sie importieren möchten.

> **Note**
>
> Diese Vorlage dient nur als Beispiel. Um sie für Ihre eigenen Testzwecke zu verwenden, ersetzen Sie die Beispielressourcen durch Ressourcen aus Ihrem Konto.

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable": {
            "Type": "AWS::DynamoDB::Table",
            "Properties": {
                "TableName": "Service",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
                    }
                ],
```

```
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        }
    }
}
```

Importieren Sie eine vorhandene Ressource in einen Stack mit dem AWS Management Console

> **Note**
>
> Die CloudFormation Konsole unterstützt die Verwendung der systemeigenen Funktion `Fn::Transform` beim Importieren von Ressourcen nicht. Sie können den verwenden, um Ressourcen AWS CLI zu importieren, die diese Funktion verwenden. `Fn::Transform`

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der Seite Stacks den Stack aus, in den Sie Ressourcen importieren möchten.
3. Wählen Sie Stack-Aktionen und dann Ressourcen in Stack importieren aus.
4. Überprüfen Sie die Seite Importübersicht und wählen Sie dann Weiter aus.
5. Geben Sie auf der Seite Vorlage angeben Ihre aktualisierte Vorlage mit einer der folgenden Methoden an und wählen Sie dann Weiter aus.
   - Wählen Sie Amazon S3-URL aus und geben Sie dann die URL für Ihre Vorlage im Textfeld an.
   - Wählen Sie Vorlagendatei hochladen aus und suchen Sie dann nach Ihrer Vorlage.
6. Identifizieren Sie auf der Seite Ressourcen identifizieren jede Zielressource. Weitere Informationen finden Sie unter Ressourcen-Identifikatoren.
   a. Wählen Sie unter Bezeichnereigenschaft den Typ des Ressourcenbezeichners aus. Beispielsweise kann die `AWS::DynamoDB::Table`-Ressource mit der `TableName`-Eigenschaft identifiziert werden.
   b. Geben Sie unter Bezeichnerwert den tatsächlichen Eigenschaftswert ein. Beispielsweise ist der `TableName` für die `GamesTable`-Ressource in der Beispielvorlage *`Games`*.

c. Wählen Sie Weiter aus.

7. Aktualisieren Sie auf der Seite Stack-Details angeben alle Parameter und wählen Sie dann Weiter aus. Dadurch wird automatisch ein Änderungssatz erstellt.

> **Note**
>
> Der Importvorgang schlägt fehl, wenn Sie vorhandene Parameter ändern, die einen Erstellungs-, Aktualisierungs- oder Löschvorgang initiieren.

8. Überprüfen Sie auf der *`stack-name`* Seite Überprüfen die Ressourcen, die Sie importieren möchten, und wählen Sie dann Ressourcen importieren aus. Dadurch wird der im letzten Schritt erstellte Änderungssatz automatisch ausgeführt. Alle Tags auf Stack-Ebene werden zu diesem Zeitpunkt auf importierte Ressourcen angewendet. Weitere Informationen finden Sie unter Konfigurieren Sie die Stack-Optionen.

   Die Seite Ereignisse für den Stack wird angezeigt.

9. (Optional) Führen Sie die Abweichungserkennung für den Stack aus, um sicherzustellen, dass die Vorlage und die tatsächliche Konfiguration der importierten Ressourcen übereinstimmen. Weitere Informationen zur Erkennung von Abweichungen finden Sie unterDrift auf einem ganzen CloudFormation Stack erkennen.
10. (Optional) Wenn die importierten Ressourcen nicht mit den erwarteten Vorlagenkonfigurationen übereinstimmen, korrigieren Sie entweder die Vorlagenkonfigurationen oder aktualisieren Sie die Ressourcen direkt. Weitere Informationen zum Importieren von Ressourcen, die nicht berücksichtigt wurden, finden Sie unterBeheben von Abweichungen mit einer Importoperation.

Importieren Sie eine vorhandene Ressource in einen Stapel mithilfe der AWS CLI

1. Um zu erfahren, welche Eigenschaften die einzelnen Ressourcentypen in der Vorlage identifizieren, führen Sie den get-template-summary Befehl aus und geben Sie dabei die S3-URL der Vorlage an. Beispielsweise kann die `AWS::DynamoDB::Table`-Ressource mit der `TableName`-Eigenschaft identifiziert werden. Für die `GamesTable`-Ressource in der Beispielvorlage lautet der Wert von `TableName` `Games`. Sie benötigen diese Informationen im nächsten Schritt.

```
aws cloudformation get-template-summary \
```

```
    --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
TemplateToImport.json
```

Weitere Informationen finden Sie unter Ressourcen-Identifikatoren.

2. Erstellen Sie eine Liste der tatsächlich zu importierenden Ressourcen und ihrer eindeutigen Identifikatoren im folgenden JSON-Zeichenkettenformat.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifie
{"TableName":"Games"}}]
```

Alternativ können Sie die JSON-formatierten Parameter in einer Konfigurationsdatei angeben.

Um beispielsweise zu importieren`GamesTable`, können Sie eine *`ResourcesToImport.txt`* Datei erstellen, die die folgende Konfiguration enthält.

```
[
  {
      "ResourceType":"AWS::DynamoDB::Table",
      "LogicalResourceId":"GamesTable",
      "ResourceIdentifier": {
        "TableName":"Games"
      }
  }
]
```

3. Um einen Änderungssatz zu erstellen, verwenden Sie den folgenden create-change-set Befehl und ersetzen Sie den Platzhaltertext. Geben Sie für die Option `--change-set-type` einen Wert von **`IMPORT`** an. Ersetzen Sie für `--resources-to-import` diese Option die JSON-Beispielzeichenfolge durch die tatsächliche JSON-Zeichenfolge, die Sie gerade erstellt haben.

```
aws cloudformation create-change-set \
    --stack-name TargetStack --change-set-name ImportChangeSet \
    --change-set-type IMPORT \
    --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
TemplateToImport.json \
    --resources-to-
import '[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","Resource
{"TableName":"Games"}}]'
```

> **Note**
>
> `--resources-to-import`unterstützt Inline-YAML nicht. Die Anforderungen für das Umgehen von Anführungszeichen in der JSON-Zeichenfolge variieren je nach Terminal. Weitere Informationen finden Sie im Benutzerhandbuch unter Verwenden von Anführungszeichen innerhalb von Zeichenketten.AWS Command Line Interface

Alternativ können Sie eine Datei-URL als Eingabe für die `--resources-to-import` Option verwenden, wie im folgenden Beispiel gezeigt.

```
--resources-to-import file://ResourcesToImport.txt
```

4. Überprüfen Sie den Änderungssatz, um sicherzustellen, dass die richtigen Ressourcen importiert werden.

```
aws cloudformation describe-change-set \
    --change-set-name ImportChangeSet --stack-name TargetStack
```

5. Verwenden Sie den folgenden execute-change-set Befehl und ersetzen Sie den Platzhaltertext, um den Änderungssatz zu initiieren und die Ressourcen zu importieren. Alle Tags auf Stack-Ebene werden zu diesem Zeitpunkt auf importierte Ressourcen angewendet. Weitere Informationen finden Sie unter Konfigurieren Sie die Stack-Optionen. Nach erfolgreichem Abschluss des Vorgangs (`IMPORT_COMPLETE`) werden die Ressourcen erfolgreich importiert.

```
aws cloudformation execute-change-set \
    --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Optional) Führen Sie die Abweichungserkennung für den `IMPORT_COMPLETE`-Stack aus, um sicherzustellen, dass die Vorlage und die tatsächliche Konfiguration der importierten Ressourcen übereinstimmen. Weitere Informationen zur Erkennung von Abweichungen finden Sie unterDrift auf einem ganzen CloudFormation Stack erkennen.

   a. Führt die Drifterkennung für den angegebenen Stack aus.

   ```
   aws cloudformation detect-stack-drift --stack-name TargetStack
   ```

   Bei Erfolg gibt dieser Befehl die folgende Beispielausgabe zurück.

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

b. Zeigt den Fortschritt eines Drift-Erkennungsvorgangs für die angegebene Stack-Drift-Erkennungs-ID an.

```
aws cloudformation describe-stack-drift-detection-status \
    --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

c. Zeigt Drift-Informationen für die Ressourcen an, die im angegebenen Stack auf Drift überprüft wurden.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Optional) Wenn die importierten Ressourcen nicht mit den erwarteten Vorlagenkonfigurationen übereinstimmen, korrigieren Sie entweder die Vorlagenkonfigurationen oder aktualisieren Sie die Ressourcen direkt. Weitere Informationen zum Importieren von Ressourcen, die nicht mehr zur Verfügung stehen, finden Sie unterBeheben von Abweichungen mit einer Importoperation.

Verschieben von Ressourcen zwischen Stacks

Mithilfe der `resource import`-Funktion können Sie Ressourcen zwischen Stacks verschieben oder Stacks umgestalten. Sie müssen zuerst eine `Retain`-Löschrichtlinie zu der Ressource hinzufügen, die Sie verschieben möchten, um sicherzustellen, dass die Ressource erhalten bleibt, wenn Sie sie aus dem Quell-Stack entfernen und in den Ziel-Stack importieren.

Wenn Sie mit dem Importieren noch nicht vertraut sind, empfehlen wir Ihnen, zunächst die einführenden Informationen im AWS Ressourcen in einen CloudFormation Stack importieren Thema zu lesen.

> **Important**
>
> Nicht alle Ressourcen unterstützen Importvorgänge. Weitere Informationen finden Sie unter Ressourcen, die Importvorgänge unterstützen bevor Sie eine Ressource aus dem Stack entfernen. Wenn Sie eine Ressource entfernen, die keine Importvorgänge aus dem Stack unterstützt, können Sie die Ressource nicht in einen anderen Stack importieren oder sie wieder in den Quell-Stack bringen.

Refaktorieren Sie einen Stack mit dem AWS Management Console

1. Geben Sie in der Quellvorlage a `Retain` DeletionPolicyfür die Ressource an, die Sie verschieben möchten.

   In der folgenden Beispiel-Quellvorlage ist `Games` das Ziel dieser Umgestaltung.

   Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
```

```
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        }
    }
}
```

2. Öffnen Sie die CloudFormation Konsole, um ein Stack-Update durchzuführen und die Löschrichtlinie anzuwenden.

   a. Wählen Sie auf der Seite Stacks (Stapel) bei ausgewähltem Stack Update (Aktualisieren) aus.

   b. Wählen Sie unter Vorlage vorbereiten die Option Aktuelle Vorlage ersetzen aus.

   c. Geben Sie unter Vorlage angeben die aktualisierte Quellvorlage mit dem `DeletionPolicy`-Attribut für `GamesTable` ein und wählen Sie dann Weiter aus.

      - Wählen Sie Amazon S3-URL aus und geben Sie dann im Textfeld die URL zur aktualisierten Quellvorlage an.
      - Wählen Sie Vorlagendatei hochladen aus und suchen Sie dann nach der aktualisierten Quellvorlagendatei.

   d. Auf der Seite Stack-Details angeben sind keine Änderungen erforderlich. Wählen Sie Weiter aus.

   e. Auf der Seite Stack-Optionen konfigurieren sind keine Änderungen erforderlich. Wählen Sie Weiter aus.

   f. Überprüfen Sie auf der *`SourceStackName`* Seite „Überprüfen“ Ihre Änderungen. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie I acknowledge that this template may create IAM resources, um anzugeben, dass Sie IAM-Ressourcen in der Vorlage verwenden möchten. Weitere Informationen zum Verwenden von IAM-Ressourcen in

Vorlagen finden Sie unter Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management. Aktualisieren Sie dann entweder Ihren Quell-Stack, indem Sie einen Änderungssatz erstellen, oder aktualisieren Sie Ihren Quell-Stack direkt.

3. Entfernen Sie die Ressource, die zugehörigen Parameter und Ausgaben aus der Quellvorlage und fügen Sie sie dann der Zielvorlage hinzu.

   Die Quellvorlage sieht nun wie folgt aus.

   Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        }
    }
}
```

   Die folgende Beispiel-Zielvorlage enthält derzeit die `PlayersTable`-Ressource und enthält nun auch `GamesTable`.

Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "PlayersTable": {
            "Type": "AWS::DynamoDB::Table",
            "Properties": {
                "TableName": "Players",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
```

```
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        }
    }
}
```

4. Wiederholen Sie die Schritte 2–3, um den Quell-Stack erneut zu aktualisieren, um dieses Mal die Zielressource aus dem Stack zu löschen.
5. Führen Sie einen Importvorgang aus, der `GamesTable` dem Ziel-Stack hinzufügt.
   a. Wählen Sie auf der Seite Stacks mit ausgewähltem übergeordneten Stack Stack-Aktionen und dann Ressourcen in Stack importieren aus.
   b. Auf der Seite Importübersicht finden Sie eine Liste der Dinge, die Sie während dieses Vorgangs angeben müssen. Wählen Sie anschließend Weiter.
   c. Führen Sie auf der Seite Vorlage angeben eine der folgenden Aktionen aus und wählen Sie dann Weiter.
      - Wählen Sie Amazon S3-URL aus und geben Sie dann im Textfeld eine URL an.
      - Wählen Sie Vorlagendatei hochladen aus und suchen Sie dann nach einer Datei, die hochgeladen werden soll.
   d. Identifizieren Sie auf der Seite Ressourcen identifizieren die Ressource, die Sie verschieben möchten (in diesem Beispiel `GamesTable`). Weitere Informationen finden Sie unter Ressourcen-Identifikatoren.
      1. Wählen Sie unter Bezeichnereigenschaft den Typ des Ressourcenbezeichners aus. Beispielsweise kann eine `AWS::DynamoDB::Table`-Ressource mit der `TableName`-Eigenschaft identifiziert werden.
      2. Geben Sie unter Bezeichnerwert den tatsächlichen Eigenschaftswert ein. Beispiel, *`GamesTables`*.
      3. Wählen Sie Weiter aus.
   e. Ändern Sie auf der Seite Stack-Details angeben alle Parameter und wählen Sie dann Weiter aus. Dadurch wird automatisch ein Änderungssatz erstellt.

> **Important**
>
> Der Importvorgang schlägt fehl, wenn Sie vorhandene Parameter ändern, die einen Erstellungs-, Aktualisierungs- oder Löschvorgang initiieren.

f. Vergewissern Sie sich auf der *`TargetStackName`* Seite Überprüfen, dass die richtige Ressource importiert wurde, und wählen Sie dann Ressourcen importieren aus. Dadurch wird der im letzten Schritt erstellte Änderungssatz automatisch ausgeführt. Alle Tags auf Stack-Ebene werden zu diesem Zeitpunkt auf importierte Ressourcen angewendet.

g. Der Bereich Ereignisse der Seite Stack-Details für Ihren übergeordneten Stack wird angezeigt.

> **Note**
>
> Es ist nicht erforderlich, die Abweichungserkennung für den übergeordneten Stack nach diesem Importvorgang auszuführen, da die `AWS::CloudFormation::Stack`-Ressource bereits von CloudFormation verwaltet wird.

Refaktorieren Sie einen Stack mit dem AWS CLI

1. Geben Sie in der Quellvorlage a `Retain` DeletionPolicyfür die Ressource an, die Sie verschieben möchten.

   In der folgenden Beispiel-Quellvorlage ist `GamesTable` das Ziel dieser Umgestaltung.

   Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
```

```
                {
                   "AttributeName":"key",
                   "AttributeType":"S"
                }
             ],
             "KeySchema":[
                {
                   "AttributeName":"key",
                   "KeyType":"HASH"
                }
             ],
             "ProvisionedThroughput":{
                "ReadCapacityUnits":5,
                "WriteCapacityUnits":1
             }
          }
       },
       "GamesTable": {
           "Type": "AWS::DynamoDB::Table",
           "DeletionPolicy": "Retain",
           "Properties": {
               "TableName": "Games",
               "AttributeDefinitions": [
                   {
                       "AttributeName": "key",
                       "AttributeType": "S"
                   }
               ],
               "KeySchema": [
                   {
                       "AttributeName": "key",
                       "KeyType": "HASH"
                   }
               ],
               "ProvisionedThroughput": {
                   "ReadCapacityUnits": 5,
                   "WriteCapacityUnits": 1
               }
           }
       }
   }
}
```

2. Aktualisieren Sie den Quell-Stack, um die Löschrichtlinie auf die Ressource anzuwenden.

```
aws cloudformation update-stack --stack-name SourceStackName
```

3. Entfernen Sie die Ressource, die zugehörigen Parameter und Ausgaben aus der Quellvorlage und fügen Sie sie dann der Zielvorlage hinzu.

   Die Quellvorlage sieht nun wie folgt aus.

   Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        }
    }
}
```

   Die folgende Beispiel-Zielvorlage enthält derzeit die `PlayersTable`-Ressource und enthält nun auch `GamesTable`.

Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "PlayersTable": {
            "Type": "AWS::DynamoDB::Table",
            "Properties": {
                "TableName": "Players",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
```

```
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        }
    }
}
```

4. Aktualisieren Sie den Quell-Stack, um die `GamesTable`-Ressource und die zugehörigen Parameter und Ausgaben aus dem Stack zu löschen.

```
aws cloudformation update-stack --stack-name SourceStackName
```

5. Stellen Sie eine Liste der tatsächlich zu importierenden Ressourcen und ihrer eindeutigen Bezeichner im folgenden JSON-Zeichenfolgenformat zusammen. Weitere Informationen finden Sie unter Ressourcen-Identifikatoren.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifie
{"TableName":"Games"}}]
```

Alternativ können Sie die JSON-formatierten Parameter in einer Konfigurationsdatei angeben.

Um beispielsweise zu importieren`GamesTable`, können Sie eine *`ResourcesToImport.txt`* Datei erstellen, die die folgende Konfiguration enthält.

```
[
  {
      "ResourceType":"AWS::DynamoDB::Table",
      "LogicalResourceId":"GamesTable",
      "ResourceIdentifier": {
        "TableName":"Games"
      }
  }
]
```

6. Um einen Änderungssatz zu erstellen, verwenden Sie den folgenden create-change-set Befehl und ersetzen Sie den Platzhaltertext. Geben Sie für die Option `--change-set-type` einen

Wert von **IMPORT** an. Ersetzen Sie für `--resources-to-import` diese Option die JSON-Beispielzeichenfolge durch die tatsächliche JSON-Zeichenfolge, die Sie gerade erstellt haben.

```
aws cloudformation create-change-set \
    --stack-name TargetStackName --change-set-name ImportChangeSet \
    --change-set-type IMPORT \
    --template-body file://TemplateToImport.json \
    --resources-to-import
 "'[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdenti
{"TableName":"Games"}}]'"
```

> **Note**
>
> `--resources-to-import`unterstützt Inline-YAML nicht. Die Anforderungen für das Umgehen von Anführungszeichen in der JSON-Zeichenfolge variieren je nach Terminal. Weitere Informationen finden Sie im Benutzerhandbuch unter Verwenden von Anführungszeichen innerhalb von Zeichenketten.AWS Command Line Interface

Alternativ können Sie eine Datei-URL als Eingabe für die `--resources-to-import` Option verwenden, wie im folgenden Beispiel gezeigt.

```
--resources-to-import file://ResourcesToImport.txt
```

7. Überprüfen Sie den Änderungssatz, um sicherzustellen, dass die richtige Ressource in den Ziel-Stack importiert wird.

```
aws cloudformation describe-change-set \
    --change-set-name ImportChangeSet
```

8. Verwenden Sie den folgenden execute-change-set Befehl und ersetzen Sie den Platzhaltertext, um den Änderungssatz zu initiieren und die Ressource zu importieren. Alle Tags auf Stack-Ebene werden zu diesem Zeitpunkt auf importierte Ressourcen angewendet. Nach erfolgreichem Abschluss des Vorgangs (`IMPORT_COMPLETE`) wird die Ressource erfolgreich importiert.

```
aws cloudformation execute-change-set \
    --change-set-name ImportChangeSet --stack-name TargetStackName
```

> **Note**
>
> Es ist nicht erforderlich, die Abweichungserkennung für den Ziel-Stack nach diesem Importvorgang auszuführen, da die Ressource bereits von CloudFormation verwaltet wird.

Verschachteln eines vorhandenen Stacks

Verwenden Sie die `resource import`-Funktion, um einen vorhandenen Stack in einem anderen vorhandenen Stack zu verschachteln. Verschachtelte Stacks sind gängige Komponenten, die Sie in anderen Vorlagen deklarieren und referenzieren. Auf diese Weise können Sie das Kopieren und Einfügen derselben Konfigurationen in Ihre Vorlagen vermeiden und Stack-Aktualisierungen vereinfachen. Wenn Sie über eine Vorlage für eine gängige Komponente verfügen, können Sie die `AWS::CloudFormation::Stack`-Ressource verwenden, um aus einer anderen Vorlage auf diese Vorlage zu verweisen. Weitere Hinweise zu verschachtelten Stacks finden Sie unter Teilen Sie eine Vorlage mithilfe verschachtelter Stapel in wiederverwendbare Teile auf.

AWS CloudFormation unterstützt nur eine Verschachtelungsebene. `resource import` Dies bedeutet, dass Sie keinen Stack in einen untergeordneten Stack importieren oder einen Stack importieren können, der untergeordnete Stacks enthält.

Wenn Sie mit dem Importieren noch nicht vertraut sind, empfehlen wir Ihnen, zunächst die einführenden Informationen im Manuelles Importieren von AWS Ressourcen in einen CloudFormation Stack Thema zu lesen.

Validierung des Imports von verschachtelten Stacks

AWS CloudFormation Führt während eines Importvorgangs für verschachtelte Stacks die folgenden Validierungen durch.

- Die verschachtelte `AWS::CloudFormation::Stack`-Definition in der übergeordneten Stack-Vorlage entspricht der tatsächlichen geschachtelten Stack-Vorlage.
- Die Tags für die verschachtelte `AWS::CloudFormation::Stack`-Definition in der übergeordneten Stack-Vorlage entsprechen den Tags für die tatsächliche verschachtelte Stack-Ressource.

Verschachteln eines vorhandenen Stacks mithilfe der AWS Management Console

1. Fügen Sie die `AWS::CloudFormation::Stack` Ressource mit einem zur übergeordneten Stack-Vorlage hinzu `Retain` DeletionPolicy. Im folgenden Beispiel `MyNestedStack` ist die Vorlage für den übergeordneten Stapel das Ziel des Imports.

   JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        },
    "MyNestedStack" : {
      "Type" : "AWS::CloudFormation::Stack",
      "DeletionPolicy": "Retain",
      "Properties" : {
      "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template",
        "Parameters" : {
          "InstanceType" : "t1.micro",
          "KeyName" : "mykey"
        }
```

```
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  ServiceTable:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  MyNestedStack:
    Type: 'AWS::CloudFormation::Stack'
    DeletionPolicy: Retain
    Properties:
      TemplateURL: >-
        https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template
      Parameters:
        InstanceType: t1.micro
        KeyName: mykey
```

2. Öffnen Sie die AWS CloudFormation Konsole.
3. Wählen Sie auf der Seite Stacks mit ausgewähltem übergeordneten Stack Stack-Aktionen und dann Ressourcen in Stack importieren aus.
4. Auf der Seite Importübersicht finden Sie eine Liste der Dinge, die Sie während dieses Vorgangs angeben müssen. Wählen Sie anschließend Weiter.
5. Geben Sie auf der Seite Vorlage angeben die aktualisierte übergeordnete Vorlage mit einer der folgenden Methoden an und wählen Sie dann Weiter aus.

- Wählen Sie Amazon S3-URL aus und geben Sie dann die URL für Ihre Vorlage im Textfeld an.
- Wählen Sie Vorlagendatei hochladen aus und suchen Sie dann nach Ihrer Vorlage.

6. Identifizieren Sie auf der Seite Ressourcen identifizieren die `AWS::CloudFormation::Stack`-Ressource.

   a. Wählen Sie unter Bezeichnereigenschaft den Typ des Ressourcenbezeichners aus. Beispielsweise kann eine `AWS::CloudFormation::Stack`-Ressource mit der `StackId`-Eigenschaft identifiziert werden.

   b. Geben Sie unter Identifier-Wert den ARN des Stacks ein, den Sie importieren. Beispiel, *`arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10`*.

   c. Wählen Sie Weiter aus.

7. Ändern Sie auf der Seite Stack-Details angeben alle Parameter und wählen Sie dann Weiter aus. Dadurch wird automatisch ein Änderungssatz erstellt.

   > **Important**
   >
   > Der Importvorgang schlägt fehl, wenn Sie vorhandene Parameter ändern, die einen Erstellungs-, Aktualisierungs- oder Löschvorgang initiieren.

8. Vergewissern Sie sich auf der *`MyParentStack`* Seite „Überprüfen“, dass die richtige Ressource importiert wird, und wählen Sie dann Ressourcen importieren aus. Dadurch wird der im letzten Schritt erstellte Änderungssatz automatisch ausgeführt. Alle Tags auf Stack-Ebene werden zu diesem Zeitpunkt auf importierte Ressourcen angewendet.

9. Der Bereich Ereignisse der Seite Stack-Details für Ihren übergeordneten Stack wird angezeigt.

   > **Note**
   >
   > Es ist nicht erforderlich, die Abweichungserkennung für den übergeordneten Stack nach diesem Importvorgang auszuführen, da die `AWS::CloudFormation::Stack`-Ressource bereits von AWS CloudFormation verwaltet wurde.

Verschachteln eines vorhandenen Stacks mithilfe der AWS CLI

1. Fügen Sie die `AWS::CloudFormation::Stack` Ressource mit einem zur übergeordneten Stack-Vorlage hinzu `Retain` DeletionPolicy. Im folgenden Beispiel einer übergeordneten Vorlage ist `MyNestedStack` das Ziel des Imports.

   JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        },
    "MyNestedStack" : {
      "Type" : "AWS::CloudFormation::Stack",
      "DeletionPolicy": "Retain",
      "Properties" : {
      "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template",
        "Parameters" : {
          "InstanceType" : "t1.micro",
          "KeyName" : "mykey"
        }
```

```
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  ServiceTable:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  MyNestedStack:
    Type: 'AWS::CloudFormation::Stack'
    DeletionPolicy: Retain
    Properties:
      TemplateURL: >-
        https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template
      Parameters:
        InstanceType: t1.micro
        KeyName: mykey
```

2. Verfassen Sie eine JSON-Zeichenfolge, wie im folgenden Beispiel gezeigt, mit diesen Änderungen:

   - *MyNestedStack*Ersetzen Sie durch die logische ID der Zielressource, wie in der Vorlage angegeben.
   - *arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10*Ersetzen Sie durch den ARN des Stacks, den Sie importieren möchten.

```
[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack","Resource
{"StackId":"arn:aws:cloudformation:us-east-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10"}}]
```

Alternativ können Sie die Parameter in einer Konfigurationsdatei angeben.

Um beispielsweise zu importierenMyNestedStack, können Sie eine *ResourcesToImport.txt* Datei erstellen, die die folgende Konfiguration enthält.

JSON

```
[
  {
      "ResourceType":"AWS::CloudFormation::Stack",
      "LogicalResourceId":"MyNestedStack",
      "ResourceIdentifier": {
        "StackId":"arn:aws:cloudformation:us-west-2:12345678910:stack/
mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10"
      }
  }
]
```

YAML

```
ResourceType: 'AWS::CloudFormation::Stack'
  LogicalResourceId: MyNestedStack
  ResourceIdentifier:
    StackId: >-
      arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10
```

3. Um einen Änderungssatz zu erstellen, verwenden Sie den folgenden create-change-set Befehl und ersetzen Sie den Platzhaltertext. Geben Sie für die Option `--change-set-type` einen Wert von **`IMPORT`** an. Ersetzen Sie für `--resources-to-import` diese Option die JSON-Beispielzeichenfolge durch die tatsächliche JSON-Zeichenfolge, die Sie gerade erstellt haben.

```
aws cloudformation create-change-set \
    --stack-name MyParentStack --change-set-name ImportChangeSet \
    --change-set-type IMPORT \
```

```
    --template-body file://TemplateToImport.json \
    --resources-to-
import '[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack","
{"StackId":"arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10"}}]'
```

> **Note**
>
> `--resources-to-import`unterstützt Inline-YAML nicht. Die Anforderungen für das Umgehen von Anführungszeichen in der JSON-Zeichenfolge variieren je nach Terminal. Weitere Informationen finden Sie im Benutzerhandbuch unter Verwenden von Anführungszeichen innerhalb von Zeichenketten.AWS Command Line Interface

Alternativ können Sie eine Datei-URL als Eingabe für die `--resources-to-import` Option verwenden, wie im folgenden Beispiel gezeigt.

```
--resources-to-import file://ResourcesToImport.txt
```

Bei Erfolg gibt dieser Befehl die folgende Beispielausgabe zurück.

```
{
    "Id": "arn:aws:cloudformation:us-west-2:12345678910:changeSet/
ImportChangeSet/8ad75b3f-665f-46f6-a200-0b4727a9442e",
    "StackId": "arn:aws:cloudformation:us-west-2:12345678910:stack/
MyParentStack/4e345b70-1281-11ef-b027-027366d8e82b"
}
```

4. Überprüfen Sie den Änderungssatz, um sicherzustellen, dass der richtige Stack importiert wird.

```
aws cloudformation describe-change-set --change-set-name ImportChangeSet
```

5. Verwenden Sie den folgenden execute-change-set Befehl und ersetzen Sie den Platzhaltertext, um den Änderungssatz zu initiieren und den Stapel in den übergeordneten Quellstapel zu importieren. Alle Tags auf Stack-Ebene werden zu diesem Zeitpunkt auf importierte Ressourcen angewendet. Nach erfolgreichem Abschluss des Importvorgangs (`IMPORT_COMPLETE`) ist der Stack erfolgreich verschachtelt.

```
aws cloudformation execute-change-set --change-set-name ImportChangeSet
```

> **Note**
>
> Es ist nicht erforderlich, die Abweichungserkennung für den übergeordneten Stack nach diesem Importvorgang auszuführen, da die `AWS::CloudFormation::Stack`-Ressource bereits von AWS CloudFormation verwaltet wird.

Automatisches Importieren von AWS Ressourcen in einen CloudFormation Stack

Sie können jetzt benannte Ressourcen automatisch importieren, wenn Sie Stacks erstellen oder aktualisieren CloudFormation . Eine benannte Ressource ist eine Ressource mit einem benutzerdefinierten Namen. Weitere Informationen finden Sie unter Namenstyp in der CloudFormation Vorlagenreferenz.

Wenn Sie den automatischen Import starten, CloudFormation sucht es nach vorhandenen Ressourcen, die Ihrer Vorlage entsprechen, und importiert sie während der Bereitstellung. Für verschachtelte Stacks erstellen Sie den Änderungssatz aus dem Root-Stack.

Nachdem der Import abgeschlossen ist und bevor nachfolgende Stack-Vorgänge ausgeführt werden, empfehlen wir, die Abweichungserkennung für importierte Ressourcen auszuführen. Die Abweichungserkennung stellt sicher, dass die Vorlagenkonfiguration mit der tatsächlichen Konfiguration übereinstimmt. Weitere Informationen finden Sie unter Drift auf einem ganzen CloudFormation Stack erkennen.

Um eine Ressource zu importieren, müssen sie die folgenden Anforderungen erfüllen:

- Für die Ressource muss ein statischer benutzerdefinierter Name in Ihrer Vorlage definiert sein. Dynamische Namen (mit! Ref (oder andere Funktionen) werden derzeit nicht unterstützt.
- Die Ressource muss einen Wert `DeletionPolicy` von `Retain` oder haben`RetainExceptOnCreate`.
- Die Ressource darf nicht bereits zu einem anderen CloudFormation Stapel gehören.
- Der Ressourcentyp muss CloudFormation Importvorgänge unterstützen. Weitere Informationen finden Sie unter Unterstützung für Ressourcentypen.
- Die primäre ID für den Ressourcentyp muss in der Vorlage enthalten sein. Primäre Eigenschaften IDs mit schreibgeschützten Eigenschaften werden nicht unterstützt. Um herauszufinden, was die

primäre ID für einen Typ ist, suchen Sie im Ressourcenschema nach der `primaryIdentifier` Eigenschaft. Weitere Informationen zur Eigenschaft finden Sie unter PrimaryIdentifier.

Example Beispiel für einen automatischen Import

Im folgenden Beispiel wird ein Änderungssatz verwendet, `CreateChangeSet` um einen Stack zu erstellen, der auf der `my-stack` Grundlage einer Vorlagendatei aufgerufen wird`template.yaml`, und die entsprechenden Ressourcen werden automatisch importiert.

```
aws cloudformation create-change-set \
  --stack-name my-stack \
  --change-set-name CreateChangeSet \
  --change-set-type CREATE \
  --template-body file://template.yaml \
  --import-existing-resources
```

Fehlerbehebung

Wenn der automatische Import fehlschlägt, gehen Sie wie folgt vor, um das Problem zu beheben:

- Stellen Sie sicher, dass der Ressourcenname in Ihrer Vorlage genau mit dem Namen der Ressource übereinstimmt
- Stellen Sie sicher, dass die Ressource nicht bereits von einem anderen Stack verwaltet wird
- Stellen Sie sicher, dass der Ressourcentyp Importvorgänge unterstützt
- Stellen Sie sicher, dass Ihre Vorlage alle erforderlichen Eigenschaften für den Ressourcentyp enthält

Wiederherstellen eines Importvorgangs

Um einen Importvorgang wiederherzustellen, geben Sie eine `Retain`-Löschrichtlinie für die Ressource an, die Sie aus der Vorlage entfernen möchten, um sicherzustellen, dass sie erhalten bleibt, wenn Sie sie aus dem Stack löschen.

Wiederherstellen eines Importvorgangs mithilfe der AWS Management Console

1. Geben Sie a `Retain` DeletionPolicyfür die Ressourcen an, die Sie aus Ihrem Stack entfernen möchten. In der folgenden Beispielvorlage ist `GamesTable` das Ziel dieses Wiederherstellungsvorgangs.

Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
```

```
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        }
    }
}
```

2. Öffnen Sie die CloudFormation Konsole, um ein Stack-Update durchzuführen und die Löschrichtlinie anzuwenden.

   a. Wählen Sie auf der Seite Stacks mit ausgewähltem Stack Aktualisieren und dann Stack aktualisieren (Standard) aus.

   b. Wählen Sie unter Vorlage vorbereiten die Option Aktuelle Vorlage ersetzen aus.

   c. Geben Sie unter Vorlage angeben die aktualisierte Quellvorlage mit dem `DeletionPolicy`-Attribut für `GamesTable` ein und wählen Sie dann Weiter aus.

      - Wählen Sie Amazon S3-URL aus und geben Sie dann im Textfeld die URL zur aktualisierten Quellvorlage an.
      - Wählen Sie Vorlagendatei hochladen aus und suchen Sie dann nach der aktualisierten Quellvorlagendatei.

   d. Auf der Seite Stack-Details angeben sind keine Änderungen erforderlich. Wählen Sie Weiter aus.

   e. Auf der Seite Stack-Optionen konfigurieren sind keine Änderungen erforderlich. Wählen Sie Weiter aus.

   f. Überprüfen Sie auf der *`MyStack`* Seite „Überprüfen“ Ihre Änderungen. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie I acknowledge that this template may create IAM resources, um anzugeben, dass Sie IAM-Ressourcen in der Vorlage verwenden möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen. Aktualisieren Sie dann entweder Ihren Quell-Stack, indem Sie einen Änderungssatz erstellen, oder aktualisieren Sie Ihren Quell-Stack direkt.

3. Entfernen Sie die Ressource, zugehörige Parameter und Ausgaben aus der Stack-Vorlage. In diesem Beispiel sieht die Vorlage jetzt wie folgt aus.

Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        }
    }
}
```

4. Wiederholen Sie Schritt 2, um die Ressource (`GamesTable`) und die zugehörigen Parameter und Ausgaben aus dem Stack zu löschen.

Wiederherstellen eines Importvorgangs mithilfe der AWS CLI

1. Geben Sie a `Retain` DeletionPolicyfür die Ressourcen an, die Sie aus Ihrem Stack entfernen möchten. In der folgenden Beispielvorlage ist `GamesTable` das Ziel dieses Wiederherstellungsvorgangs.

Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
              }
           }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
```

```
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        }
    }
}
```

2. Aktualisieren Sie den Stack, um die Löschrichtlinie auf die Ressource anzuwenden.

```
aws cloudformation update-stack --stack-name MyStack
```

3. Entfernen Sie die Ressource, zugehörige Parameter und Ausgaben aus der Stack-Vorlage. In diesem Beispiel sieht die Vorlage jetzt wie folgt aus.

   Example JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable":{
           "Type":"AWS::DynamoDB::Table",
           "Properties":{
              "TableName":"Service",
              "AttributeDefinitions":[
                 {
                    "AttributeName":"key",
                    "AttributeType":"S"
                 }
              ],
              "KeySchema":[
                 {
                    "AttributeName":"key",
                    "KeyType":"HASH"
                 }
              ],
              "ProvisionedThroughput":{
                 "ReadCapacityUnits":5,
                 "WriteCapacityUnits":1
```

```
                }
            }
        }
    }
}
```

4. Aktualisieren Sie den Stack, um die Ressource (`GamesTable`) und die zugehörigen Parameter und Ausgaben aus dem Stack zu löschen.

```
aws cloudformation update-stack --stack-name MyStack
```

Stack-Refactoring

Stack-Refactoring vereinfacht die Reorganisation der Ressourcen in Ihren CloudFormation Stacks und behält gleichzeitig die vorhandenen Ressourceneigenschaften und -daten bei. Mit Stack-Refactoring können Sie Ressourcen zwischen Stacks verschieben, monolithische Stacks in kleinere Komponenten aufteilen oder mehrere Stacks zu einem zusammenfassen.

So funktioniert das Stack-Refactoring

Berücksichtigen Sie bei der Planung Ihres Stack-Refactorings die folgenden Schritte:

1. Beurteilen Sie Ihre aktuelle Infrastruktur: Überprüfen Sie Ihre vorhandenen CloudFormation Stacks und Ressourcen, um Möglichkeiten für das Stack-Refactoring zu identifizieren.
2. Planen Sie Ihr Refactoring: Definieren Sie, wie die Ressourcen organisiert werden sollen. Berücksichtigen Sie Ihre Abhängigkeiten, Benennungskonventionen und betrieblichen Beschränkungen. Diese können sich später auf die CloudFormation Validierung auswirken.

   Ermitteln Sie die Anzahl der Ziel-Stacks, in die Sie Ressourcen umstrukturieren werden. Sie können Ressourcen zwischen mindestens 2 Stapel und maximal 5 Stapel verschieben. Ressourcen können zwischen verschachtelten Stapeln verschoben werden.
3. Aktualisieren Sie Ihre Vorlagen: Passen Sie Ihre CloudFormation Vorlagen an die geplante Änderung an, z. B. das Verschieben von Ressourcendefinitionen zwischen Vorlagen. Sie können logisch IDs beim Refactoring umbenennen.
4. Erstellen Sie den Stack-Refactor: Geben Sie eine Liste der Stack-Namen und Vorlagen an, die Sie umgestalten möchten.

5. Prüfen Sie die Auswirkungen des Refactorings auf Ihre Infrastruktur und lösen Sie etwaige Konflikte: CloudFormation validiert die von Ihnen bereitgestellten Vorlagen und überprüft stackübergreifende Abhängigkeiten, Ressourcentypen mit Problemen bei der Tag-Aktualisierung und Konflikte mit logischen IDs von Ressourcen.

   Wenn die Validierung erfolgreich ist, CloudFormation wird eine Vorschau der Refactor-Aktionen generiert, die nach der Ausführung des Refactors ausgeführt werden.

   Wenn die Validierung fehlschlägt, können Sie es erneut versuchen, nachdem Sie die gefundenen Probleme behoben haben. Wenn Konflikte auftreten, müssen Sie eine logische ID-Zuordnung für die Ressource angeben, aus der Quelle und das Ziel der konfliktbehafteten Ressource hervorgehen.
6. Führen Sie den Refactor aus: Nachdem Sie bestätigt haben, dass die Änderungen mit der Art und Weise übereinstimmen, wie Sie Ihre Stacks umgestalten möchten, führen Sie den Stack-Refactor aus.
7. Überwachen: Sehen Sie sich den `ExecutionStatus` Status des Stack-Refaktors an, um sicherzustellen, dass er erfolgreich war.

Einschränkungen beim Stack-Refactoring

Beachten Sie bei der Planung Ihres Stack-Refactorings die folgenden Einschränkungen:

- Refactoring-Operationen erlauben keine Erstellung neuer Ressourcen, das Löschen von Ressourcen oder Änderungen an Ressourcenkonfigurationen.
- Sie können während eines Stack-Refactorings keine Parameter, Bedingungen oder Zuordnungen ändern oder neue hinzufügen. Eine mögliche Problemumgehung besteht darin, Ihren Stack zu aktualisieren, bevor Sie den Refactor durchführen.
- Sie können dieselbe Ressource nicht in mehrere Stacks umstrukturieren.
- Sie können eine Ressource nicht in einen neuen Stack umgestalten, wenn dieser Stack bestimmte Pseudo-Parameter verwendet (Beispiel:). `AWS::StackName`
- CloudFormation unterstützt keine leeren Stacks. Bevor Sie einen Stack-Refactor erstellen, der alle vorhandenen Ressourcen aus einem Stack entfernt, müssen Sie diesem Stack eine Ressource hinzufügen. Bei dieser Ressource kann es sich um eine einfache Ressource handeln, z. B. um einen `waitCondition` Ressourcentyp.
- Stack Refactor unterstützt keine Stacks, die Stack-Richtlinien enthalten, einschließlich Richtlinien, die Änderungen für Ressourcen ermöglichen.

- Die folgenden Ressourcen sind nicht verfügbar:
  - `AWS::ACMPCA::Certificate`
  - `AWS::ACMPCA::CertificateAuthority`
  - `AWS::ACMPCA::CertificateAuthorityActivation`
  - `AWS::ApiGateway::BasePathMapping`
  - `AWS::ApiGateway::Method`
  - `AWS::AppConfig::ConfigurationProfile`
  - `AWS::AppConfig::Deployment`
  - `AWS::AppConfig::Environment`
  - `AWS::AppConfig::Extension`
  - `AWS::AppConfig::ExtensionAssociation`
  - `AWS::AppStream::DirectoryConfig`
  - `AWS::AppStream::StackFleetAssociation`
  - `AWS::AppStream::StackUserAssociation`
  - `AWS::AppStream::User`
  - `AWS::BackupGateway::Hypervisor`
  - `AWS::CodePipeline::CustomActionType`
  - `AWS::Cognito::UserPoolRiskConfigurationAttachment`
  - `AWS::Cognito::UserPoolUICustomizationAttachment`
  - `AWS::Cognito::UserPoolUserToGroupAttachment`
  - `AWS::Config::ConfigRule`
  - `AWS::DataBrew::Dataset`
  - `AWS::DataBrew::Job`
  - `AWS::DataBrew::Project`
  - `AWS::DataBrew::Recipe`
  - `AWS::DataBrew::Ruleset`
  - `AWS::DataBrew::Schedule`
  - `AWS::DataZone::DataSource`
  - `AWS::DataZone::Environment`
  - `AWS::DataZone::EnvironmentBlueprintConfiguration`

- AWS::DataZone::EnvironmentProfile
- AWS::DataZone::Project
- AWS::DataZone::SubscriptionTarget
- AWS::DynamoDB::GlobalTable
- AWS::EC2::LaunchTemplate
- AWS::EC2::SpotFleet
- AWS::EC2::VolumeAttachment
- AWS::EC2::VPCDHCPOptionsAssociation
- AWS::ElasticBeanstalk::ConfigurationTemplate
- AWS::FIS::ExperimentTemplate
- AWS::Glue::Schema
- AWS::GuardDuty::IPSet
- AWS::GuardDuty::PublishingDestination
- AWS::GuardDuty::ThreatIntelSet
- AWS::ImageBuilder::Component
- AWS::IoTFleetWise::Campaign
- AWS::IoTWireless::WirelessDeviceImportTask
- AWS::Lambda::EventInvokeConfig
- AWS::Lex::BotVersion
- AWS::M2::Application
- AWS::Maester::DocumentType
- AWS::MediaTailor::Channel
- AWS::MSK::Configuration
- AWS::MSK::ServerlessCluster
- AWS::NeptuneGraph::PrivateGraphEndpoint
- AWS::Omics::AnnotationStore
- AWS::Omics::ReferenceStore
- AWS::Omics::SequenceStore
- AWS::OpenSearchServerless::Collection

- AWS::Panorama::PackageVersion

- AWS::PCAConnectorAD::Connector
- AWS::PCAConnectorAD::DirectoryRegistration
- AWS::PCAConnectorAD::Template
- AWS::PCAConnectorAD::TemplateGroupAccessControlEntry
- AWS::QuickSight::Theme
- AWS::RefactorSpaces::Environment
- AWS::RefactorSpaces::Route
- AWS::RefactorSpaces::Service
- AWS::RoboMaker::RobotApplication
- AWS::RoboMaker::SimulationApplication
- AWS::SageMaker::InferenceComponen
- AWS::ServiceCatalog::PortfolioPrincipalAssociation
- AWS::ServiceCatalog::PortfolioProductAssociation
- AWS::ServiceCatalog::PortfolioShare
- AWS::ServiceCatalog::TagOptionAssociation
- AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation
- AWS::ServiceCatalogAppRegistry::ResourceAssociation
- AWS::StepFunctions::StateMachineVersion
- AWS::Synthetics::Canary
- AWS::VoiceID::Domain
- AWS::WAFv2::IPSet
- AWS::WAFv2::RegexPatternSet
- AWS::WAFv2::RuleGroup
- AWS::WAFv2::WebACL

Refaktorierung eines Stacks mit dem AWS Command Line Interface

Die folgenden Befehle werden für das Stack-Refactoring verwendet:

- create-stack-refactor
- describe-stack-refactor

- execute-stack-refactor
- list-stack-refactors
- list-stack-refactor-actions

Im Folgenden finden Sie ein Beispiel dafür, wie Sie einen Stack mithilfe der AWS Command Line Interface (CLI) umgestalten können.

1. Zu Beginn benötigen Sie die CloudFormation Vorlage, die Sie umgestalten möchten. Der folgende Befehl ruft die Vorlage ab:

   ```
   aws cloudformation get-template --stack-name ExampleStack1
   ```

   Sobald Sie über die Vorlage verfügen, verwenden Sie die integrierte Entwicklungsumgebung (IDE) Ihrer Wahl, um sie so zu aktualisieren, dass sie die gewünschte Struktur und Ressourcenorganisation verwendet.

2. Erstellen Sie den Stack-Refactor mithilfe des `create-stack-refactor` Befehls und geben Sie den Stack-Namen und die Vorlagen für jeden Stack an, der am Refactoring beteiligt ist:

   ```
   aws cloudformation create-stack-refactor \
     --stack-definitions \
       StackName=MySns,TemplateBody@=file://afterSns.yaml \
       StackName=MyLambdaSubscription,TemplateBody@=file://afterLambda.yaml \
     --enable-stack-creation \
     --resource-mappings file://refactor.json
   ```

   Der `--resource-mappings` Parameter ist optional, aber er ist erforderlich, wenn bei der Vorlagenvalidierung ein Konflikt festgestellt wird. Zusätzlich zur Bereitstellung der `Destination` Stapel `Source` und müssen Sie auch die `LogicalResourceId` angeben. Im Folgenden sehen Sie ein Beispiel für eine `refactor.json`-Datei.

   ```
   [
       {
           "Source": {
               "StackName": "MySns",
               "LogicalResourceId": "MyFunction"
           },
           "Destination": {
               "StackName": "MyLambdaSubscription",
   ```

```
            "LogicalResourceId": "Function"
        }
    }
]
```

3. Nach der Erstellung des Refactors CloudFormation wird eine Vorschau der Refactor-Änderungen generiert, mit der Sie eine Vorschau der Auswirkungen der vorgeschlagenen Änderungen auf Ihre Infrastruktur sehen können. In diesem Beispiel ist das `StackRefactorId` Erstellte. `stack-refactor-1ab2-c34d-5ef6` Verwenden Sie den folgenden Befehl, um eine Vorschau der Änderungen anzuzeigen:

```
aws cloudformation list-stack-refactor-actions \
  --stack-refactor-id stack-refactor-1ab2-c34d-5ef6
```

4. Nachdem Sie Ihre Änderungen überprüft und bestätigt haben, verwenden Sie den `execute-stack-refactor` Befehl, um den Refactoring-Prozess abzuschließen:

```
aws cloudformation execute-stack-refactor \
  --stack-refactor-id stack-refactor-1ab2-c34d-5ef6
```

5. Nach der Ausführung `execute-stack-refactor` können Sie den Status des Vorgangs mit dem folgenden Befehl überwachen:

```
aws cloudformation describe-stack-refactor \
  --stack-refactor-id stack-refactor-1ab2-c34d-5ef6
```

   CloudFormation aktualisiert automatisch das `Status` Ende `ExecutionStatus` des Refactoring-Vorgangs.

Unterstützung für Ressourcentypen

In der folgenden Tabelle sind AWS Ressourcentypen aufgeführt, die derzeit Import-, Drifterkennungs- und IaC-Generatoroperationen (Infrastructure as Code) unterstützen. Jeder Ressourcentypname ist mit dem entsprechenden Referenzthema im AWS CloudFormation Vorlagenreferenzhandbuch verknüpft.

Eine Ressource, die den Ressourcenimport unterstützt, könnte auch den automatischen Import unterstützen. Weitere Informationen finden Sie unter AWS Ressourcen in einen CloudFormation Stack importieren.

Dies ist keine vollständige Liste von Ressourcen. AWS Wenn ein bestimmter Ressourcentyp nicht aufgeführt ist, bedeutet dies wahrscheinlich, dass er nicht über den AWS -Cloud-Control- API zugänglich ist. Weitere Informationen finden Sie im Cloud Control API-Benutzerhandbuch unter Ressourcentypen, die die Cloud Control API unterstützen. Jeder AWS Dienst bestimmt unabhängig, welche Ressourcentypen über die Cloud Control API zugänglich gemacht werden sollen.

CloudFormation unterstützt auch Import- und Drift-Erkennungsvorgänge für private Ressourcentypen, die bereitstellbar sind (solche mit dem Bereitstellungstyp entweder `FULLY_MUTABLE` oder`IMMUTABLE`). Um einen privaten Ressourcentyp zu importieren oder eine Drift-Erkennung für einen privaten Ressourcentyp durchzuführen, müssen Sie zunächst die Standardversion dieses Ressourcentyps in Ihrem Konto registrieren und sicherstellen, dass er bereitstellbar ist. Weitere Informationen finden Sie unter Verwenden Sie private Erweiterungen von Drittanbietern, die mit Ihnen geteilt wurden.

Beachten Sie, dass der IaC-Generator nur AWS Ressourcen unterstützt, die mit der Cloud Control API in Ihrer Region kompatibel sind.

Um programmgesteuert auf Informationen über öffentliche und private bereitstellbare Ressourcentypen zuzugreifen, können Sie den verwenden. AWS -Cloud-Control- API Weitere Informationen finden Sie im Cloud Control API-Benutzerhandbuch unter Feststellen, ob ein Ressourcentyp die Cloud Control API unterstützt.

Für die ersten Schritte mit Import, Drifterkennung oder IaC-Generator finden Sie hier einige nützliche Themen, die Sie sich ansehen sollten:

- AWS Ressourcen in einen CloudFormation Stack importieren
- Erkennen Sie mit Drift-Erkennung nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen
- Generieren Sie Vorlagen aus vorhandenen Ressourcen mit dem IaC-Generator

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ACMPCA::Certificate	Ja	Nein	Nein
AWS::ACMPCA::CertificateAuthority	Ja	Ja	Ja
AWS::ACMPCA::CertificateAuthorityActivation	Ja	Nein	Nein
AWS::ACMPCA::Permission	Ja	Ja	Nein
AWS::AIOps::InvestigationGroup	Ja	Ja	Nein
AWS::APS::ResourcePolicy	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::APS::RuleGroupsNamespace	Ja	Ja	Ja
AWS::APS::Scraper	Ja	Ja	Nein
AWS::APS::Workspace	Ja	Ja	Ja
AWS::ARCRegionSwitch::Plan	Ja	Ja	Nein
AWS::ARCZonalShift::AutoshiftObserverNotificationStatus	Ja	Ja	Nein
AWS::ARCZonalShift::ZonalAutoshiftConfiguration	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AccessAnalyzer::Analyzer	Ja	Ja	Ja
AWS::AmazonMQ::Configuration	Ja	Ja	Nein
AWS::Amplify::App	Ja	Ja	Ja
AWS::Amplify::Branch	Ja	Ja	Ja
AWS::Amplify::Domain	Ja	Ja	Ja
AWS::AmplifyUIBuilder::Component	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AmplifyUIBuilder::Form	Ja	Ja	Ja
AWS::AmplifyUIBuilder::Theme	Ja	Ja	Ja
AWS::ApiGateway::Account	Ja	Ja	Nein
AWS::ApiGateway::ApiKey	Ja	Ja	Ja
AWS::ApiGateway::Authorizer	Ja	Ja	Nein
AWS::ApiGateway::BasePathMapping	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ApiGateway::BasePathMappingV2	Ja	Ja	Nein
AWS::ApiGateway::ClientCertificate	Ja	Ja	Ja
AWS::ApiGateway::Deployment	Ja	Ja	Ja
AWS::ApiGateway::DocumentationPart	Ja	Ja	Nein
AWS::ApiGateway::DocumentationVersion	Ja	Ja	Ja
AWS::ApiGateway::DomainName	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ApiGateway::DomainNameAccessAssociation	Ja	Ja	Nein
AWS::ApiGateway::DomainNameV2	Ja	Ja	Nein
AWS::ApiGateway::GatewayResponse	Ja	Ja	Nein
AWS::ApiGateway::Method	Ja	Ja	Nein
AWS::ApiGateway::Model	Ja	Ja	Nein
AWS::ApiGateway::RequestValidator	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ApiGateway::Resource	Ja	Ja	Nein
AWS::ApiGateway::RestApi	Ja	Ja	Ja
AWS::ApiGateway::Stage	Ja	Nein	Ja
AWS::ApiGateway::UsagePlan	Ja	Ja	Ja
AWS::ApiGateway::UsagePlanKey	Ja	Nein	Ja
AWS::ApiGateway::VpcLink	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ApiGatewayV2::Api	Ja	Ja	Ja
AWS::ApiGatewayV2::ApiMapping	Ja	Ja	Ja
AWS::ApiGatewayV2::Authorizer	Ja	Ja	Nein
AWS::ApiGatewayV2::Deployment	Ja	Ja	Ja
AWS::ApiGatewayV2::DomainName	Ja	Ja	Ja
AWS::ApiGatewayV2::Integration	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ApiGatewayV2::IntegrationResponse	Ja	Ja	Nein
AWS::ApiGatewayV2::Model	Ja	Ja	Nein
AWS::ApiGatewayV2::Route	Ja	Ja	Nein
AWS::ApiGatewayV2::RouteResponse	Ja	Ja	Nein
AWS::ApiGatewayV2::RoutingRule	Ja	Ja	Nein
AWS::ApiGatewayV2::VpcLink	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AppConfig::Application	Ja	Ja	Ja
AWS::AppConfig::ConfigurationProfile	Ja	Ja	Nein
AWS::AppConfig::Deployment	Ja	Ja	Nein
AWS::AppConfig::DeploymentStrategy	Ja	Ja	Nein
AWS::AppConfig::Environment	Ja	Ja	Nein
AWS::AppConfig::Extension	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AppConfig::ExtensionAssociation	Ja	Ja	Nein
AWS::AppConfig::HostedConfigurationVersion	Ja	Ja	Nein
AWS::AppFlow::Connector	Ja	Ja	Ja
AWS::AppFlow::ConnectorProfile	Ja	Ja	Ja
AWS::AppFlow::Flow	Ja	Ja	Ja
AWS::AppIntegrations::Application	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AppIntegrations::DataIntegration	Ja	Ja	Ja
AWS::AppIntegrations::EventIntegration	Ja	Ja	Ja
AWS::AppRunner::AutoScalingConfiguration	Ja	Ja	Nein
AWS::AppRunner::ObservabilityConfiguration	Ja	Ja	Nein
AWS::AppRunner::Service	Ja	Ja	Nein
AWS::AppRunner::VpcConnector	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AppRunner::VpcIngressConnection	Ja	Ja	Nein
AWS::AppStream::AppBlock	Ja	Ja	Nein
AWS::AppStream::AppBlockBuilder	Ja	Ja	Nein
AWS::AppStream::Application	Ja	Ja	Nein
AWS::AppStream::ApplicationEntitlementAssociation	Ja	Ja	Nein
AWS::AppStream::ApplicationFleetAssociation	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AppStream::DirectoryConfig	Ja	Ja	Ja
AWS::AppStream::Entitlement	Ja	Ja	Nein
AWS::AppStream::ImageBuilder	Ja	Ja	Ja
AWS::AppSync::Api	Ja	Ja	Nein
AWS::AppSync::ChannelNamespace	Ja	Ja	Nein
AWS::AppSync::DataSource	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AppSync::DomainName	Ja	Ja	Ja
AWS::AppSync::DomainNameApiAssociation	Ja	Ja	Nein
AWS::AppSync::FunctionConfiguration	Ja	Ja	Nein
AWS::AppSync::GraphQLApi	Ja	Ja	Nein
AWS::AppSync::Resolver	Ja	Ja	Nein
AWS::AppSync::SourceApiAssociation	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AppTest::TestCase	Ja	Ja	Nein
AWS::ApplicationAutoScaling::ScalableTarget	Ja	Ja	Ja
AWS::ApplicationAutoScaling::ScalingPolicy	Ja	Nein	Nein
AWS::ApplicationInsights::Application	Ja	Ja	Nein
AWS::ApplicationSignals::Discovery	Ja	Ja	Nein
AWS::ApplicationSignals::ServiceLevelObjective	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Athena::CapacityReservation	Ja	Ja	Ja
AWS::Athena::DataCatalog	Ja	Ja	Ja
AWS::Athena::NamedQuery	Ja	Ja	Ja
AWS::Athena::PreparedStatement	Ja	Ja	Ja
AWS::Athena::WorkGroup	Ja	Ja	Ja
AWS::AuditManager::Assessment	Ja	Nein	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::AutoScaling::AutoScalingGroup	Ja	Nein	Ja
AWS::AutoScaling::LaunchConfiguration	Ja	Ja	Ja
AWS::AutoScaling::LifecycleHook	Ja	Ja	Ja
AWS::AutoScaling::ScalingPolicy	Ja	Ja	Ja
AWS::AutoScaling::ScheduledAction	Ja	Ja	Ja
AWS::AutoScaling::WarmPool	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::B2BI::Capability	Ja	Ja	Nein
AWS::B2BI::Partnership	Ja	Ja	Nein
AWS::B2BI::Profile	Ja	Ja	Nein
AWS::B2BI::Transformer	Ja	Ja	Nein
AWS::BCMDataExports::Export	Ja	Ja	Nein
AWS::Backup::BackupPlan	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Backup::BackupSelection	Ja	Nein	Ja
AWS::Backup::BackupVault	Ja	Ja	Nein
AWS::Backup::Framework	Ja	Ja	Ja
AWS::Backup::LogicallyAirGappedBackupVault	Ja	Ja	Nein
AWS::Backup::ReportPlan	Ja	Ja	Ja
AWS::Backup::RestoreTestingPlan	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Backup::RestoreTestingSelection	Ja	Ja	Nein
AWS::BackupGateway::Hypervisor	Ja	Ja	Nein
AWS::Batch::ComputeEnvironment	Ja	Ja	Ja
AWS::Batch::ConsumableResource	Ja	Ja	Nein
AWS::Batch::JobDefinition	Ja	Ja	Nein
AWS::Batch::JobQueue	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Batch::SchedulingPolicy	Ja	Ja	Ja
AWS::Batch::ServiceEnvironment	Ja	Ja	Nein
AWS::Bedrock::Agent	Ja	Ja	Ja
AWS::Bedrock::AgentAlias	Ja	Ja	Ja
AWS::Bedrock::ApplicationInferenceProfile	Ja	Ja	Ja
AWS::Bedrock::Blueprint	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Bedrock::DataAutomationProject	Ja	Ja	Ja
AWS::Bedrock::DataSource	Ja	Ja	Ja
AWS::Bedrock::Flow	Ja	Ja	Ja
AWS::Bedrock::FlowAlias	Ja	Ja	Ja
AWS::Bedrock::FlowVersion	Ja	Ja	Ja
AWS::Bedrock::Guardrail	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Bedrock::GuardrailVersion	Ja	Ja	Nein
AWS::Bedrock::IntelligentPromptRouter	Ja	Ja	Ja
AWS::Bedrock::KnowledgeBase	Ja	Nein	Ja
AWS::Bedrock::Prompt	Ja	Ja	Ja
AWS::Bedrock::PromptVersion	Ja	Ja	Ja
AWS::Billing::BillingView	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::BillingConductor::BillingGroup	Ja	Ja	Ja
AWS::BillingConductor::CustomLineItem	Ja	Ja	Ja
AWS::BillingConductor::PricingPlan	Ja	Ja	Ja
AWS::BillingConductor::PricingRule	Ja	Ja	Ja
AWS::Budgets::BudgetsAction	Ja	Ja	Ja
AWS::CE::AnomalyMonitor	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CE::AnomalySubscription	Ja	Ja	Nein
AWS::CE::CostCategory	Ja	Ja	Ja
AWS::CUR::ReportDefinition	Ja	Ja	Nein
AWS::Cassandra::Keyspace	Ja	Ja	Nein
AWS::Cassandra::Table	Ja	Ja	Ja
AWS::Cassandra::Type	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CertificateManager::Account	Ja	Ja	Nein
AWS::Chatbot::CustomAction	Ja	Ja	Nein
AWS::Chatbot::MicrosoftTeamsChannelConfiguration	Ja	Ja	Ja
AWS::Chatbot::SlackChannelConfiguration	Ja	Ja	Ja
AWS::CleanRooms::AnalysisTemplate	Ja	Ja	Ja
AWS::CleanRooms::Collaboration	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CleanRooms::ConfiguredTable	Ja	Ja	Ja
AWS::CleanRooms::ConfiguredTableAssociation	Ja	Ja	Ja
AWS::CleanRooms::IdMappingTable	Ja	Ja	Nein
AWS::CleanRooms::IdNamespaceAssociation	Ja	Ja	Nein
AWS::CleanRooms::Membership	Ja	Ja	Ja
AWS::CleanRooms::PrivacyBudgetTemplate	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CleanRoomsML::TrainingDataset	Ja	Ja	Nein
AWS::CloudFormation::GuardHook	Ja	Ja	Nein
AWS::CloudFormation::HookDefaultVersion	Ja	Ja	Ja
AWS::CloudFormation::HookTypeConfig	Ja	Ja	Ja
AWS::CloudFormation::HookVersion	Ja	Ja	Ja
AWS::CloudFormation::LambdaHook	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CloudFormation::ModuleDefaultVersion	Ja	Ja	Nein
AWS::CloudFormation::ModuleVersion	Ja	Ja	Nein
AWS::CloudFormation::PublicTypeVersion	Ja	Ja	Nein
AWS::CloudFormation::Publisher	Ja	Ja	Nein
AWS::CloudFormation::ResourceDefaultVersion	Ja	Ja	Nein
AWS::CloudFormation::ResourceVersion	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CloudFormation::Stack	Ja	Ja	Nein
AWS::CloudFormation::StackSet	Ja	Ja	Nein
AWS::CloudFormation::TypeActivation	Ja	Ja	Nein
AWS::CloudFront::AnycastIpList	Ja	Ja	Nein
AWS::CloudFront::CachePolicy	Ja	Ja	Ja
AWS::CloudFront::CloudFrontOriginAccessIdentity	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CloudFront::ConnectionGroup	Ja	Ja	Nein
AWS::CloudFront::ContinuousDeploymentPolicy	Ja	Ja	Ja
AWS::CloudFront::Distribution	Ja	Ja	Ja
AWS::CloudFront::DistributionTenant	Ja	Ja	Nein
AWS::CloudFront::Function	Ja	Ja	Nein
AWS::CloudFront::KeyGroup	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CloudFront::KeyValueStore	Ja	Ja	Nein
AWS::CloudFront::MonitoringSubscription	Ja	Ja	Nein
AWS::CloudFront::OriginAccessControl	Ja	Ja	Ja
AWS::CloudFront::OriginRequestPolicy	Ja	Ja	Ja
AWS::CloudFront::PublicKey	Ja	Ja	Ja
AWS::CloudFront::RealtimeLogConfig	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CloudFront::ResponseHeadersPolicy	Ja	Ja	Nein
AWS::CloudFront::VpcOrigin	Ja	Ja	Nein
AWS::CloudTrail::Channel	Ja	Ja	Nein
AWS::CloudTrail::Dashboard	Ja	Ja	Nein
AWS::CloudTrail::EventDataStore	Ja	Ja	Ja
AWS::CloudTrail::ResourcePolicy	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CloudTrail::Trail	Ja	Ja	Ja
AWS::CloudWatch::Alarm	Ja	Ja	Ja
AWS::CloudWatch::CompositeAlarm	Ja	Ja	Ja
AWS::CloudWatch::Dashboard	Ja	Ja	Nein
AWS::CloudWatch::MetricStream	Ja	Ja	Nein
AWS::CodeArtifact::Domain	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CodeArtifact::PackageGroup	Ja	Ja	Nein
AWS::CodeArtifact::Repository	Ja	Ja	Ja
AWS::CodeBuild::Fleet	Ja	Ja	Nein
AWS::CodeConnections::Connection	Ja	Ja	Nein
AWS::CodeDeploy::Application	Ja	Ja	Ja
AWS::CodeDeploy::DeploymentConfig	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CodeGuruProfiler::ProfilingGroup	Ja	Ja	Ja
AWS::CodeGuruReviewer::RepositoryAssociation	Ja	Ja	Nein
AWS::CodePipeline::CustomActionType	Ja	Ja	Nein
AWS::CodePipeline::Pipeline	Ja	Ja	Nein
AWS::CodePipeline::Webhook	Ja	Ja	Nein
AWS::CodeStarConnections::Connection	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CodeStarConnections::RepositoryLink	Ja	Ja	Nein
AWS::CodeStarConnections::SyncConfiguration	Ja	Ja	Nein
AWS::CodeStarNotifications::NotificationRule	Ja	Ja	Ja
AWS::Cognito::IdentityPool	Ja	Ja	Nein
AWS::Cognito::IdentityPoolPrincipalTag	Ja	Ja	Ja
AWS::Cognito::IdentityPoolRoleAttachment	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Cognito::LogDeliveryConfiguration	Ja	Nein	Nein
AWS::Cognito::ManagedLoginBranding	Ja	Ja	Nein
AWS::Cognito::UserPool	Ja	Nein	Nein
AWS::Cognito::UserPoolClient	Ja	Ja	Nein
AWS::Cognito::UserPoolDomain	Ja	Ja	Nein
AWS::Cognito::UserPoolGroup	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Cognito::UserPoolIdentityProvider	Ja	Ja	Nein
AWS::Cognito::UserPoolResourceServer	Ja	Ja	Nein
AWS::Cognito::UserPoolRiskConfigurationAttachment	Ja	Ja	Nein
AWS::Cognito::UserPoolUICustomizationAttachment	Ja	Ja	Nein
AWS::Cognito::UserPoolUser	Ja	Nein	Nein
AWS::Cognito::UserPoolUserToGroupAttachment	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Comprehend::DocumentClassifier	Ja	Ja	Ja
AWS::Comprehend::Flywheel	Ja	Ja	Ja
AWS::Config::AggregationAuthorization	Ja	Ja	Ja
AWS::Config::ConfigRule	Ja	Ja	Ja
AWS::Config::ConfigurationAggregator	Ja	Ja	Ja
AWS::Config::ConformancePack	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Config::OrganizationConformancePack	Ja	Ja	Ja
AWS::Config::StoredQuery	Ja	Ja	Ja
AWS::Connect::AgentStatus	Ja	Ja	Ja
AWS::Connect::ApprovedOrigin	Ja	Ja	Ja
AWS::Connect::ContactFlow	Ja	Ja	Ja
AWS::Connect::ContactFlowModule	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Connect::ContactFlowVersion	Ja	Ja	Ja
AWS::Connect::EmailAddress	Ja	Ja	Ja
AWS::Connect::EvaluationForm	Ja	Ja	Ja
AWS::Connect::HoursOfOperation	Ja	Ja	Ja
AWS::Connect::Instance	Ja	Ja	Ja
AWS::Connect::InstanceStorageConfig	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Connect::IntegrationAssociation	Ja	Ja	Ja
AWS::Connect::PhoneNumber	Ja	Ja	Ja
AWS::Connect::PredefinedAttribute	Ja	Ja	Ja
AWS::Connect::Prompt	Ja	Ja	Ja
AWS::Connect::Queue	Ja	Ja	Ja
AWS::Connect::QuickConnect	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Connect::RoutingProfile	Ja	Ja	Ja
AWS::Connect::Rule	Ja	Ja	Nein
AWS::Connect::SecurityKey	Ja	Ja	Ja
AWS::Connect::SecurityProfile	Ja	Ja	Ja
AWS::Connect::TaskTemplate	Ja	Ja	Ja
AWS::Connect::TrafficDistributionGroup	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Connect::User	Ja	Ja	Ja
AWS::Connect::UserHierarchyGroup	Ja	Ja	Ja
AWS::Connect::UserHierarchyStructure	Ja	Ja	Nein
AWS::Connect::View	Ja	Ja	Ja
AWS::Connect::ViewVersion	Ja	Ja	Ja
AWS::ConnectCampaigns::Campaign	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ConnectCampaignsV2::Campaign	Ja	Ja	Ja
AWS::ControlTower::EnabledBaseline	Ja	Ja	Nein
AWS::ControlTower::EnabledControl	Ja	Ja	Ja
AWS::ControlTower::LandingZone	Ja	Ja	Nein
AWS::CustomerProfiles::CalculatedAttributeDefinition	Ja	Ja	Ja
AWS::CustomerProfiles::Domain	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::CustomerProfiles::EventStream	Ja	Ja	Ja
AWS::CustomerProfiles::EventTrigger	Ja	Ja	Nein
AWS::CustomerProfiles::Integration	Ja	Ja	Ja
AWS::CustomerProfiles::ObjectType	Ja	Ja	Ja
AWS::CustomerProfiles::SegmentDefinition	Ja	Ja	Nein
AWS::DMS::DataMigration	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::DMS::DataProvider	Ja	Ja	Nein
AWS::DMS::InstanceProfile	Ja	Ja	Nein
AWS::DMS::MigrationProject	Ja	Ja	Nein
AWS::DMS::ReplicationConfig	Ja	Ja	Ja
AWS::DSQL::Cluster	Ja	Ja	Nein
AWS::DataBrew::Dataset	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::DataBrew::Job	Ja	Ja	Ja
AWS::DataBrew::Project	Ja	Ja	Ja
AWS::DataBrew::Recipe	Ja	Ja	Ja
AWS::DataBrew::Ruleset	Ja	Ja	Ja
AWS::DataBrew::Schedule	Ja	Ja	Ja
AWS::DataPipeline::Pipeline	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::DataSync::Agent	Ja	Ja	Ja
AWS::DataSync::LocationAzureBlob	Ja	Ja	Ja
AWS::DataSync::LocationEFS	Ja	Ja	Ja
AWS::DataSync::LocationFSxLustre	Ja	Ja	Ja
AWS::DataSync::LocationFSxONTAP	Ja	Ja	Ja
AWS::DataSync::LocationFSxOpenZFS	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::DataSync::LocationFSxWindows	Ja	Ja	Ja
AWS::DataSync::LocationHDFS	Ja	Ja	Ja
AWS::DataSync::LocationNFS	Ja	Ja	Ja
AWS::DataSync::LocationObjectStorage	Ja	Ja	Ja
AWS::DataSync::LocationS3	Ja	Ja	Ja
AWS::DataSync::LocationSMB	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::DataSync::Task	Ja	Ja	Ja
AWS::DataZone::Connection	Ja	Ja	Nein
AWS::DataZone::DataSource	Ja	Ja	Nein
AWS::DataZone::Domain	Ja	Ja	Nein
AWS::DataZone::DomainUnit	Ja	Ja	Nein
AWS::DataZone::Environment	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::DataZone::EnvironmentActions	Ja	Ja	Nein
AWS::DataZone::EnvironmentBlueprintConfiguration	Ja	Ja	Nein
AWS::DataZone::EnvironmentProfile	Ja	Ja	Nein
AWS::DataZone::GroupProfile	Ja	Ja	Nein
AWS::DataZone::Owner	Ja	Ja	Nein
AWS::DataZone::PolicyGrant	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::DataZone::Project	Ja	Ja	Nein
AWS::DataZone::ProjectMembership	Ja	Ja	Nein
AWS::DataZone::ProjectProfile	Ja	Ja	Nein
AWS::DataZone::SubscriptionTarget	Ja	Ja	Nein
AWS::DataZone::UserProfile	Ja	Ja	Nein
AWS::Deadline::Farm	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Deadline::Fleet	Ja	Ja	Nein
AWS::Deadline::LicenseEndpoint	Ja	Ja	Nein
AWS::Deadline::Limit	Ja	Ja	Nein
AWS::Deadline::MeteredProduct	Ja	Ja	Nein
AWS::Deadline::Monitor	Ja	Ja	Nein
AWS::Deadline::Queue	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Deadline::QueueEnvironment	Ja	Ja	Nein
AWS::Deadline::QueueFleetAssociation	Ja	Ja	Nein
AWS::Deadline::QueueLimitAssociation	Ja	Ja	Nein
AWS::Deadline::StorageProfile	Ja	Ja	Nein
AWS::Detective::Graph	Ja	Ja	Ja
AWS::Detective::MemberInvitation	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Detective::OrganizationAdmin	Ja	Ja	Ja
AWS::DevOpsGuru::LogAnomalyDetectionIntegration	Ja	Ja	Ja
AWS::DevOpsGuru::NotificationChannel	Ja	Ja	Ja
AWS::DevOpsGuru::ResourceCollection	Ja	Ja	Ja
AWS::DeviceFarm::DevicePool	Ja	Ja	Nein
AWS::DeviceFarm::InstanceProfile	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::DeviceFarm::NetworkProfile	Ja	Ja	Nein
AWS::DeviceFarm::Project	Ja	Ja	Nein
AWS::DeviceFarm::TestGridProject	Ja	Ja	Nein
AWS::DeviceFarm::VPCEConfiguration	Ja	Ja	Nein
AWS::DirectoryService::SimpleAD	Ja	Ja	Ja
AWS::DocDBElastic::Cluster	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::DynamoDB::GlobalTable	Ja	Ja	Nein
AWS::DynamoDB::Table	Ja	Ja	Ja
AWS::EC2::CapacityReservation	Ja	Ja	Ja
AWS::EC2::CapacityReservationFleet	Ja	Ja	Ja
AWS::EC2::CarrierGateway	Ja	Ja	Ja
AWS::EC2::CustomerGateway	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::DHCPOptions	Ja	Ja	Ja
AWS::EC2::EC2Fleet	Ja	Ja	Nein
AWS::EC2::EIP	Ja	Ja	Ja
AWS::EC2::EIPAssociation	Ja	Nein	Ja
AWS::EC2::EgressOnlyInternetGateway	Ja	Ja	Ja
AWS::EC2::EnclaveCertificateIamRoleAssociation	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::FlowLog	Ja	Ja	Ja
AWS::EC2::GatewayRouteTableAssociation	Ja	Ja	Nein
AWS::EC2::Host	Ja	Ja	Ja
AWS::EC2::IPAM	Ja	Ja	Ja
AWS::EC2::IPAMAllocation	Ja	Ja	Ja
AWS::EC2::IPAMPool	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::IPAMPoolCidr	Ja	Ja	Ja
AWS::EC2::IPAMResourceDiscovery	Ja	Ja	Ja
AWS::EC2::IPAMResourceDiscoveryAssociation	Ja	Ja	Ja
AWS::EC2::IPAMScope	Ja	Ja	Ja
AWS::EC2::Instance	Ja	Ja	Ja
AWS::EC2::InstanceConnectEndpoint	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::InternetGateway	Ja	Ja	Ja
AWS::EC2::IpPoolRouteTableAssociation	Ja	Ja	Nein
AWS::EC2::KeyPair	Ja	Ja	Ja
AWS::EC2::LaunchTemplate	Ja	Ja	Nein
AWS::EC2::LocalGatewayRoute	Ja	Ja	Ja
AWS::EC2::LocalGatewayRouteTable	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::LocalGatewayRouteTableVPCAssociation	Ja	Ja	Ja
AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation	Ja	Ja	Ja
AWS::EC2::NatGateway	Ja	Ja	Ja
AWS::EC2::NetworkAcl	Ja	Ja	Ja
AWS::EC2::NetworkInsightsAccessScope	Ja	Ja	Nein
AWS::EC2::NetworkInsightsAccessScopeAnalysis	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::NetworkInsightsAnalysis	Ja	Ja	Nein
AWS::EC2::NetworkInsightsPath	Ja	Ja	Ja
AWS::EC2::NetworkInterface	Ja	Ja	Ja
AWS::EC2::NetworkInterfaceAttachment	Ja	Ja	Ja
AWS::EC2::NetworkPerformanceMetricSubscription	Ja	Ja	Ja
AWS::EC2::PlacementGroup	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::PrefixList	Ja	Ja	Nein
AWS::EC2::Route	Ja	Ja	Ja
AWS::EC2::RouteServer	Ja	Ja	Nein
AWS::EC2::RouteServerAssociation	Ja	Ja	Nein
AWS::EC2::RouteServerEndpoint	Ja	Ja	Nein
AWS::EC2::RouteServerPeer	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::RouteServerPropagation	Ja	Ja	Nein
AWS::EC2::RouteTable	Ja	Ja	Ja
AWS::EC2::SecurityGroup	Ja	Ja	Ja
AWS::EC2::SecurityGroupEgress	Ja	Ja	Nein
AWS::EC2::SecurityGroupIngress	Ja	Ja	Nein
AWS::EC2::SecurityGroupVpcAssociation	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::SnapshotBlockPublicAccess	Ja	Ja	Nein
AWS::EC2::SpotFleet	Ja	Ja	Nein
AWS::EC2::Subnet	Ja	Ja	Ja
AWS::EC2::SubnetCidrBlock	Ja	Ja	Ja
AWS::EC2::SubnetNetworkAclAssociation	Ja	Ja	Ja
AWS::EC2::SubnetRouteTableAssociation	Ja	Nein	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::TrafficMirrorFilter	Ja	Ja	Nein
AWS::EC2::TrafficMirrorFilterRule	Ja	Ja	Nein
AWS::EC2::TrafficMirrorSession	Ja	Ja	Nein
AWS::EC2::TrafficMirrorTarget	Ja	Ja	Nein
AWS::EC2::TransitGateway	Ja	Ja	Ja
AWS::EC2::TransitGatewayAttachment	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::TransitGatewayConnect	Ja	Ja	Ja
AWS::EC2::TransitGatewayConnectPeer	Ja	Ja	Nein
AWS::EC2::TransitGatewayMulticastDomain	Ja	Ja	Ja
AWS::EC2::TransitGatewayMulticastDomainAssociation	Ja	Ja	Ja
AWS::EC2::TransitGatewayMulticastGroupMember	Ja	Ja	Ja
AWS::EC2::TransitGatewayMulticastGroupSource	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::TransitGatewayPeeringAttachment	Ja	Ja	Ja
AWS::EC2::TransitGatewayRoute	Ja	Ja	Nein
AWS::EC2::TransitGatewayRouteTable	Ja	Ja	Ja
AWS::EC2::TransitGatewayRouteTableAssociation	Ja	Ja	Nein
AWS::EC2::TransitGatewayRouteTablePropagation	Ja	Ja	Nein
AWS::EC2::TransitGatewayVpcAttachment	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::VPC	Ja	Ja	Ja
AWS::EC2::VPCBlockPublicAccessExclusion	Ja	Ja	Nein
AWS::EC2::VPCBlockPublicAccessOptions	Ja	Ja	Nein
AWS::EC2::VPCCidrBlock	Ja	Ja	Nein
AWS::EC2::VPCDHCPOptionsAssociation	Ja	Ja	Ja
AWS::EC2::VPCEndpoint	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::VPCEndpointConnectionNotification	Ja	Ja	Ja
AWS::EC2::VPCEndpointService	Ja	Ja	Ja
AWS::EC2::VPCEndpointServicePermissions	Ja	Ja	Nein
AWS::EC2::VPCGatewayAttachment	Ja	Ja	Nein
AWS::EC2::VPCPeeringConnection	Ja	Ja	Ja
AWS::EC2::VPNConnection	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::VPNConnectionRoute	Ja	Ja	Ja
AWS::EC2::VPNGateway	Ja	Ja	Ja
AWS::EC2::VerifiedAccessEndpoint	Ja	Ja	Ja
AWS::EC2::VerifiedAccessGroup	Ja	Ja	Ja
AWS::EC2::VerifiedAccessInstance	Ja	Ja	Ja
AWS::EC2::VerifiedAccessTrustProvider	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EC2::Volume	Ja	Ja	Ja
AWS::EC2::VolumeAttachment	Ja	Ja	Ja
AWS::ECR::PublicRepository	Ja	Ja	Ja
AWS::ECR::PullThroughCacheRule	Ja	Ja	Ja
AWS::ECR::RegistryPolicy	Ja	Ja	Ja
AWS::ECR::RegistryScanningConfiguration	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ECR::ReplicationConfiguration	Ja	Ja	Ja
AWS::ECR::Repository	Ja	Ja	Ja
AWS::ECR::RepositoryCreationTemplate	Ja	Ja	Nein
AWS::ECS::CapacityProvider	Ja	Ja	Nein
AWS::ECS::Cluster	Ja	Nein	Ja
AWS::ECS::ClusterCapacityProviderAssociations	Ja	Nein	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ECS::PrimaryTaskSet	Ja	Ja	Nein
AWS::ECS::Service	Ja	Ja	Ja
AWS::ECS::TaskDefinition	Ja	Ja	Ja
AWS::ECS::TaskSet	Ja	Ja	Nein
AWS::EFS::AccessPoint	Ja	Ja	Ja
AWS::EFS::FileSystem	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EFS::MountTarget	Ja	Ja	Ja
AWS::EKS::AccessEntry	Ja	Ja	Nein
AWS::EKS::Addon	Ja	Ja	Ja
AWS::EKS::Cluster	Ja	Nein	Ja
AWS::EKS::FargateProfile	Ja	Ja	Ja
AWS::EKS::IdentityProviderConfig	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EKS::Nodegroup	Ja	Nein	Ja
AWS::EKS::PodIdentityAssociation	Ja	Ja	Nein
AWS::EMR::SecurityConfiguration	Ja	Ja	Ja
AWS::EMR::Step	Ja	Nein	Nein
AWS::EMR::Studio	Ja	Ja	Ja
AWS::EMR::StudioSessionMapping	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EMR::WALWorkspace	Ja	Ja	Nein
AWS::EMRContainers::VirtualCluster	Ja	Ja	Nein
AWS::EMRServerless::Application	Ja	Ja	Ja
AWS::EVS::Environment	Ja	Ja	Nein
AWS::ElastiCache::GlobalReplicationGroup	Ja	Ja	Nein
AWS::ElastiCache::ParameterGroup	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ElastiCache::ServerlessCache	Ja	Ja	Nein
AWS::ElastiCache::SubnetGroup	Ja	Ja	Ja
AWS::ElastiCache::User	Ja	Ja	Ja
AWS::ElastiCache::UserGroup	Ja	Ja	Ja
AWS::ElasticBeanstalk::Application	Ja	Ja	Ja
AWS::ElasticBeanstalk::ApplicationVersion	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ElasticBeanstalk::ConfigurationTemplate	Ja	Ja	Ja
AWS::ElasticBeanstalk::Environment	Ja	Nein	Nein
AWS::ElasticLoadBalancingV2::Listener	Ja	Ja	Ja
AWS::ElasticLoadBalancingV2::ListenerRule	Ja	Nein	Ja
AWS::ElasticLoadBalancingV2::LoadBalancer	Ja	Ja	Ja
AWS::ElasticLoadBalancingV2::TargetGroup	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ElasticLoadBalancingV2::TrustStore	Ja	Ja	Nein
AWS::ElasticLoadBalancingV2::TrustStoreRevocation	Ja	Ja	Nein
AWS::EntityResolution::IdMappingWorkflow	Ja	Ja	Ja
AWS::EntityResolution::IdNamespace	Ja	Ja	Nein
AWS::EntityResolution::MatchingWorkflow	Ja	Ja	Ja
AWS::EntityResolution::PolicyStatement	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::EntityResolution::SchemaMapping	Ja	Ja	Ja
AWS::EventSchemas::Discoverer	Ja	Ja	Nein
AWS::EventSchemas::Registry	Ja	Ja	Nein
AWS::EventSchemas::RegistryPolicy	Ja	Ja	Nein
AWS::EventSchemas::Schema	Ja	Ja	Nein
AWS::Events::ApiDestination	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Events::Archive	Ja	Ja	Ja
AWS::Events::Connection	Ja	Ja	Nein
AWS::Events::Endpoint	Ja	Ja	Ja
AWS::Events::EventBus	Ja	Ja	Ja
AWS::Events::Rule	Ja	Ja	Ja
AWS::Evidently::Experiment	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Evidently::Feature	Ja	Nein	Nein
AWS::Evidently::Launch	Ja	Ja	Nein
AWS::Evidently::Project	Ja	Ja	Nein
AWS::Evidently::Segment	Ja	Ja	Nein
AWS::FIS::ExperimentTemplate	Ja	Ja	Ja
AWS::FIS::TargetAccountConfiguration	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::FMS::NotificationChannel	Ja	Ja	Nein
AWS::FMS::Policy	Ja	Nein	Nein
AWS::FMS::ResourceSet	Ja	Ja	Nein
AWS::FSx::DataRepositoryAssociation	Ja	Ja	Ja
AWS::FSx::S3AccessPointAttachment	Ja	Ja	Nein
AWS::FinSpace::Environment	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Forecast::Dataset	Ja	Ja	Nein
AWS::Forecast::DatasetGroup	Ja	Ja	Nein
AWS::FraudDetector::Detector	Ja	Ja	Ja
AWS::FraudDetector::EntityType	Ja	Ja	Ja
AWS::FraudDetector::EventType	Ja	Ja	Ja
AWS::FraudDetector::Label	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::FraudDetector::List	Ja	Ja	Ja
AWS::FraudDetector::Outcome	Ja	Ja	Ja
AWS::FraudDetector::Variable	Ja	Ja	Ja
AWS::GameLift::Alias	Ja	Ja	Ja
AWS::GameLift::Build	Ja	Ja	Ja
AWS::GameLift::ContainerFleet	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::GameLift::ContainerGroupDefinition	Ja	Ja	Nein
AWS::GameLift::Fleet	Ja	Ja	Ja
AWS::GameLift::GameServerGroup	Ja	Ja	Nein
AWS::GameLift::GameSessionQueue	Ja	Ja	Nein
AWS::GameLift::Location	Ja	Ja	Nein
AWS::GameLift::MatchmakingConfiguration	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::GameLift::MatchmakingRuleSet	Ja	Ja	Nein
AWS::GameLift::Script	Ja	Ja	Nein
AWS::GameLiftStreams::Application	Ja	Ja	Nein
AWS::GameLiftStreams::StreamGroup	Ja	Ja	Nein
AWS::GlobalAccelerator::Accelerator	Ja	Ja	Ja
AWS::GlobalAccelerator::CrossAccountAttachment	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::GlobalAccelerator::EndpointGroup	Ja	Ja	Ja
AWS::GlobalAccelerator::Listener	Ja	Ja	Ja
AWS::Glue::Crawler	Ja	Ja	Nein
AWS::Glue::Database	Ja	Ja	Nein
AWS::Glue::Job	Ja	Ja	Nein
AWS::Glue::Registry	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Glue::Schema	Ja	Ja	Ja
AWS::Glue::SchemaVersion	Ja	Ja	Ja
AWS::Glue::SchemaVersionMetadata	Ja	Ja	Ja
AWS::Glue::Trigger	Ja	Ja	Nein
AWS::Glue::UsageProfile	Ja	Ja	Nein
AWS::Grafana::Workspace	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::GreengrassV2::ComponentVersion	Ja	Ja	Nein
AWS::GreengrassV2::Deployment	Ja	Ja	Ja
AWS::GroundStation::Config	Ja	Ja	Ja
AWS::GroundStation::DataflowEndpointGroup	Ja	Ja	Ja
AWS::GroundStation::MissionProfile	Ja	Ja	Ja
AWS::GuardDuty::Detector	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::GuardDuty::Filter	Ja	Ja	Nein
AWS::GuardDuty::IPSet	Ja	Ja	Nein
AWS::GuardDuty::MalwareProtectionPlan	Ja	Ja	Nein
AWS::GuardDuty::Master	Ja	Ja	Nein
AWS::GuardDuty::Member	Ja	Ja	Nein
AWS::GuardDuty::PublishingDestination	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::GuardDuty::ThreatEntitySet	Ja	Ja	Nein
AWS::GuardDuty::ThreatIntelSet	Ja	Ja	Nein
AWS::GuardDuty::TrustedEntitySet	Ja	Ja	Nein
AWS::HealthImaging::Datastore	Ja	Ja	Ja
AWS::HealthLake::FHIRDatastore	Ja	Ja	Ja
AWS::IAM::Group	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IAM::GroupPolicy	Ja	Ja	Nein
AWS::IAM::InstanceProfile	Ja	Ja	Ja
AWS::IAM::ManagedPolicy	Ja	Ja	Ja
AWS::IAM::OIDCProvider	Ja	Ja	Ja
AWS::IAM::Role	Ja	Ja	Ja
AWS::IAM::RolePolicy	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IAM::SAMLProvider	Ja	Ja	Ja
AWS::IAM::ServerCertificate	Ja	Ja	Ja
AWS::IAM::ServiceLinkedRole	Ja	Ja	Nein
AWS::IAM::User	Ja	Ja	Ja
AWS::IAM::UserPolicy	Ja	Ja	Nein
AWS::IAM::VirtualMFADevice	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IVS::Channel	Ja	Ja	Ja
AWS::IVS::EncoderConfiguration	Ja	Ja	Nein
AWS::IVS::IngestConfiguration	Ja	Ja	Nein
AWS::IVS::PlaybackKeyPair	Ja	Ja	Ja
AWS::IVS::PlaybackRestrictionPolicy	Ja	Ja	Nein
AWS::IVS::PublicKey	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IVS::RecordingConfiguration	Ja	Ja	Ja
AWS::IVS::Stage	Ja	Ja	Nein
AWS::IVS::StorageConfiguration	Ja	Ja	Nein
AWS::IVS::StreamKey	Ja	Ja	Nein
AWS::IVSChat::LoggingConfiguration	Ja	Ja	Ja
AWS::IVSChat::Room	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IdentityStore::Group	Ja	Ja	Ja
AWS::IdentityStore::GroupMembership	Ja	Ja	Ja
AWS::ImageBuilder::Component	Ja	Ja	Ja
AWS::ImageBuilder::ContainerRecipe	Ja	Ja	Nein
AWS::ImageBuilder::DistributionConfiguration	Ja	Ja	Ja
AWS::ImageBuilder::Image	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ImageBuilder::ImagePipeline	Ja	Ja	Ja
AWS::ImageBuilder::ImageRecipe	Ja	Ja	Ja
AWS::ImageBuilder::InfrastructureConfiguration	Ja	Ja	Ja
AWS::ImageBuilder::LifecyclePolicy	Ja	Ja	Nein
AWS::ImageBuilder::Workflow	Ja	Ja	Nein
AWS::Inspector::AssessmentTarget	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Inspector::AssessmentTemplate	Ja	Ja	Ja
AWS::Inspector::ResourceGroup	Ja	Ja	Nein
AWS::InspectorV2::CisScanConfiguration	Ja	Ja	Nein
AWS::InspectorV2::CodeSecurityIntegration	Ja	Ja	Nein
AWS::InspectorV2::CodeSecurityScanConfiguration	Ja	Ja	Nein
AWS::InspectorV2::Filter	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::InternetMonitor::Monitor	Ja	Ja	Ja
AWS::Invoicing::InvoiceUnit	Ja	Ja	Nein
AWS::IoT::AccountAuditConfiguration	Ja	Ja	Ja
AWS::IoT::Authorizer	Ja	Ja	Ja
AWS::IoT::BillingGroup	Ja	Ja	Ja
AWS::IoT::CACertificate	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoT::Certificate	Ja	Ja	Ja
AWS::IoT::CertificateProvider	Ja	Ja	Nein
AWS::IoT::Command	Ja	Ja	Nein
AWS::IoT::CustomMetric	Ja	Ja	Ja
AWS::IoT::Dimension	Ja	Ja	Ja
AWS::IoT::DomainConfiguration	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoT::EncryptionConfiguration	Ja	Ja	Nein
AWS::IoT::FleetMetric	Ja	Ja	Nein
AWS::IoT::JobTemplate	Ja	Ja	Nein
AWS::IoT::Logging	Ja	Ja	Ja
AWS::IoT::MitigationAction	Ja	Ja	Ja
AWS::IoT::Policy	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoT::ProvisioningTemplate	Ja	Ja	Ja
AWS::IoT::ResourceSpecificLogging	Ja	Ja	Ja
AWS::IoT::RoleAlias	Ja	Ja	Ja
AWS::IoT::ScheduledAudit	Ja	Ja	Ja
AWS::IoT::SecurityProfile	Ja	Ja	Ja
AWS::IoT::SoftwarePackage	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoT::SoftwarePackageVersion	Ja	Ja	Nein
AWS::IoT::Thing	Ja	Ja	Ja
AWS::IoT::ThingGroup	Ja	Ja	Ja
AWS::IoT::ThingType	Ja	Ja	Nein
AWS::IoT::TopicRule	Ja	Ja	Ja
AWS::IoT::TopicRuleDestination	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoTAnalytics::Channel	Ja	Ja	Ja
AWS::IoTAnalytics::Dataset	Ja	Ja	Ja
AWS::IoTAnalytics::Datastore	Ja	Ja	Ja
AWS::IoTAnalytics::Pipeline	Ja	Ja	Ja
AWS::IoTCoreDeviceAdvisor::SuiteDefinition	Ja	Ja	Ja
AWS::IoTEvents::AlarmModel	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoTEvents::DetectorModel	Ja	Ja	Ja
AWS::IoTEvents::Input	Ja	Ja	Ja
AWS::IoTFleetHub::Application	Ja	Ja	Ja
AWS::IoTFleetWise::Campaign	Ja	Ja	Ja
AWS::IoTFleetWise::DecoderManifest	Ja	Ja	Ja
AWS::IoTFleetWise::Fleet	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoTFleetWise::ModelManifest	Ja	Ja	Ja
AWS::IoTFleetWise::SignalCatalog	Ja	Ja	Ja
AWS::IoTFleetWise::StateTemplate	Ja	Ja	Nein
AWS::IoTFleetWise::Vehicle	Ja	Ja	Ja
AWS::IoTSiteWise::AccessPolicy	Ja	Ja	Nein
AWS::IoTSiteWise::Asset	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoTSiteWise::AssetModel	Ja	Ja	Ja
AWS::IoTSiteWise::ComputationModel	Ja	Ja	Nein
AWS::IoTSiteWise::Dashboard	Ja	Ja	Nein
AWS::IoTSiteWise::Dataset	Ja	Ja	Nein
AWS::IoTSiteWise::Gateway	Ja	Ja	Ja
AWS::IoTSiteWise::Portal	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoTSiteWise::Project	Ja	Ja	Nein
AWS::IoTTwinMaker::ComponentType	Ja	Ja	Ja
AWS::IoTTwinMaker::Entity	Ja	Ja	Ja
AWS::IoTTwinMaker::Scene	Ja	Ja	Ja
AWS::IoTTwinMaker::SyncJob	Ja	Ja	Ja
AWS::IoTTwinMaker::Workspace	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoTWireless::Destination	Ja	Ja	Ja
AWS::IoTWireless::DeviceProfile	Ja	Ja	Ja
AWS::IoTWireless::FuotaTask	Ja	Ja	Ja
AWS::IoTWireless::MulticastGroup	Ja	Ja	Ja
AWS::IoTWireless::NetworkAnalyzerConfiguration	Ja	Ja	Ja
AWS::IoTWireless::PartnerAccount	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::IoTWireless::ServiceProfile	Ja	Ja	Ja
AWS::IoTWireless::TaskDefinition	Ja	Ja	Ja
AWS::IoTWireless::WirelessDevice	Ja	Ja	Ja
AWS::IoTWireless::WirelessDeviceImportTask	Ja	Ja	Ja
AWS::IoTWireless::WirelessGateway	Ja	Ja	Ja
AWS::KMS::Alias	Ja	Nein	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::KMS::Key	Ja	Nein	Ja
AWS::KMS::ReplicaKey	Ja	Nein	Ja
AWS::KafkaConnect::Connector	Ja	Nein	Nein
AWS::KafkaConnect::CustomPlugin	Ja	Ja	Nein
AWS::KafkaConnect::WorkerConfiguration	Ja	Ja	Nein
AWS::Kendra::DataSource	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Kendra::Faq	Ja	Ja	Ja
AWS::Kendra::Index	Ja	Ja	Ja
AWS::KendraRanking::ExecutionPlan	Ja	Ja	Ja
AWS::Kinesis::ResourcePolicy	Ja	Ja	Nein
AWS::Kinesis::Stream	Ja	Ja	Ja
AWS::Kinesis::StreamConsumer	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::KinesisAnalyticsV2::Application	Ja	Ja	Nein
AWS::KinesisFirehose::DeliveryStream	Ja	Nein	Nein
AWS::KinesisVideo::SignalingChannel	Ja	Ja	Nein
AWS::KinesisVideo::Stream	Ja	Ja	Nein
AWS::LakeFormation::DataCellsFilter	Ja	Ja	Nein
AWS::LakeFormation::PrincipalPermissions	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::LakeFormation::Tag	Ja	Ja	Nein
AWS::LakeFormation::TagAssociation	Ja	Ja	Nein
AWS::Lambda::Alias	Ja	Ja	Nein
AWS::Lambda::CodeSigningConfig	Ja	Ja	Ja
AWS::Lambda::EventInvokeConfig	Ja	Ja	Nein
AWS::Lambda::EventSourceMapping	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Lambda::Function	Ja	Ja	Ja
AWS::Lambda::LayerVersion	Ja	Ja	Nein
AWS::Lambda::LayerVersionPermission	Ja	Ja	Ja
AWS::Lambda::Permission	Ja	Ja	Ja
AWS::Lambda::Url	Ja	Ja	Ja
AWS::Lambda::Version	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::LaunchWizard::Deployment	Ja	Ja	Nein
AWS::Lex::Bot	Ja	Ja	Nein
AWS::Lex::BotAlias	Ja	Ja	Nein
AWS::Lex::BotVersion	Ja	Ja	Nein
AWS::Lex::ResourcePolicy	Ja	Ja	Nein
AWS::LicenseManager::Grant	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::LicenseManager::License	Ja	Ja	Nein
AWS::Lightsail::Alarm	Ja	Ja	Ja
AWS::Lightsail::Bucket	Ja	Ja	Ja
AWS::Lightsail::Certificate	Ja	Ja	Ja
AWS::Lightsail::Container	Ja	Ja	Ja
AWS::Lightsail::Database	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Lightsail::Disk	Ja	Ja	Ja
AWS::Lightsail::Distribution	Ja	Ja	Ja
AWS::Lightsail::Domain	Ja	Ja	Nein
AWS::Lightsail::Instance	Ja	Ja	Ja
AWS::Lightsail::InstanceSnapshot	Ja	Ja	Nein
AWS::Lightsail::LoadBalancer	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Lightsail::LoadBalancerTlsCertificate	Ja	Ja	Ja
AWS::Lightsail::StaticIp	Ja	Ja	Ja
AWS::Location::APIKey	Ja	Ja	Nein
AWS::Location::GeofenceCollection	Ja	Ja	Ja
AWS::Location::Map	Ja	Ja	Ja
AWS::Location::PlaceIndex	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Location::RouteCalculator	Ja	Ja	Ja
AWS::Location::Tracker	Ja	Ja	Ja
AWS::Location::TrackerConsumer	Ja	Ja	Ja
AWS::Logs::AccountPolicy	Ja	Ja	Ja
AWS::Logs::Delivery	Ja	Ja	Nein
AWS::Logs::DeliveryDestination	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Logs::DeliverySource	Ja	Ja	Nein
AWS::Logs::Destination	Ja	Ja	Ja
AWS::Logs::Integration	Ja	Ja	Nein
AWS::Logs::LogAnomalyDetector	Ja	Ja	Nein
AWS::Logs::LogGroup	Ja	Ja	Ja
AWS::Logs::LogStream	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Logs::MetricFilter	Ja	Ja	Ja
AWS::Logs::QueryDefinition	Ja	Ja	Nein
AWS::Logs::ResourcePolicy	Ja	Ja	Ja
AWS::Logs::SubscriptionFilter	Ja	Ja	Ja
AWS::Logs::Transformer	Ja	Ja	Nein
AWS::LookoutEquipment::InferenceScheduler	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::LookoutMetrics::Alert	Ja	Ja	Nein
AWS::LookoutMetrics::AnomalyDetector	Ja	Ja	Nein
AWS::LookoutVision::Project	Ja	Ja	Ja
AWS::M2::Application	Ja	Ja	Nein
AWS::M2::Deployment	Ja	Ja	Nein
AWS::M2::Environment	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::MPA::ApprovalTeam	Ja	Ja	Nein
AWS::MPA::IdentitySource	Ja	Ja	Nein
AWS::MSK::BatchScramSecret	Ja	Ja	Ja
AWS::MSK::Cluster	Ja	Ja	Ja
AWS::MSK::ClusterPolicy	Ja	Ja	Ja
AWS::MSK::Configuration	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::MSK::Replicator	Ja	Ja	Ja
AWS::MSK::ServerlessCluster	Ja	Ja	Ja
AWS::MSK::VpcConnection	Ja	Ja	Ja
AWS::MWAA::Environment	Ja	Ja	Nein
AWS::Macie::AllowList	Ja	Ja	Nein
AWS::Macie::CustomDataIdentifier	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Macie::FindingsFilter	Ja	Ja	Nein
AWS::Macie::Session	Ja	Ja	Ja
AWS::ManagedBlockchain::Accessor	Ja	Ja	Ja
AWS::MediaConnect::Bridge	Ja	Ja	Ja
AWS::MediaConnect::BridgeOutput	Ja	Ja	Nein
AWS::MediaConnect::BridgeSource	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::MediaConnect::Flow	Ja	Ja	Nein
AWS::MediaConnect::FlowEntitlement	Ja	Ja	Nein
AWS::MediaConnect::FlowOutput	Ja	Ja	Nein
AWS::MediaConnect::FlowSource	Ja	Ja	Nein
AWS::MediaConnect::FlowVpcInterface	Ja	Ja	Ja
AWS::MediaConnect::Gateway	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::MediaLive::ChannelPlacementGroup	Ja	Ja	Nein
AWS::MediaLive::CloudWatchAlarmTemplate	Ja	Ja	Nein
AWS::MediaLive::CloudWatchAlarmTemplateGroup	Ja	Ja	Nein
AWS::MediaLive::Cluster	Ja	Ja	Nein
AWS::MediaLive::EventBridgeRuleTemplate	Ja	Ja	Nein
AWS::MediaLive::EventBridgeRuleTemplateGroup	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::MediaLive::Multiplex	Ja	Ja	Nein
AWS::MediaLive::Multiplexprogram	Ja	Ja	Nein
AWS::MediaLive::Network	Ja	Ja	Nein
AWS::MediaLive::SdiSource	Ja	Ja	Nein
AWS::MediaLive::SignalMap	Ja	Ja	Nein
AWS::MediaPackage::Asset	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::MediaPackage::Channel	Ja	Ja	Ja
AWS::MediaPackage::OriginEndpoint	Ja	Ja	Ja
AWS::MediaPackage::PackagingConfiguration	Ja	Ja	Ja
AWS::MediaPackage::PackagingGroup	Ja	Ja	Ja
AWS::MediaPackageV2::Channel	Ja	Ja	Ja
AWS::MediaPackageV2::ChannelGroup	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::MediaPackageV2::ChannelPolicy	Ja	Ja	Nein
AWS::MediaPackageV2::OriginEndpoint	Ja	Ja	Ja
AWS::MediaPackageV2::OriginEndpointPolicy	Ja	Ja	Nein
AWS::MediaTailor::Channel	Ja	Ja	Nein
AWS::MediaTailor::ChannelPolicy	Ja	Ja	Nein
AWS::MediaTailor::LiveSource	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::MediaTailor::PlaybackConfiguration	Ja	Ja	Nein
AWS::MediaTailor::SourceLocation	Ja	Ja	Ja
AWS::MediaTailor::VodSource	Ja	Ja	Ja
AWS::MemoryDB::ACL	Ja	Ja	Ja
AWS::MemoryDB::Cluster	Ja	Ja	Ja
AWS::MemoryDB::MultiRegionCluster	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::MemoryDB::ParameterGroup	Ja	Ja	Ja
AWS::MemoryDB::SubnetGroup	Ja	Ja	Ja
AWS::MemoryDB::User	Ja	Ja	Ja
AWS::Neptune::DBCluster	Ja	Ja	Ja
AWS::Neptune::DBClusterParameterGroup	Ja	Ja	Nein
AWS::Neptune::DBInstance	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Neptune::DBParameterGroup	Ja	Ja	Nein
AWS::Neptune::DBSubnetGroup	Ja	Ja	Nein
AWS::NeptuneGraph::Graph	Ja	Ja	Nein
AWS::NeptuneGraph::PrivateGraphEndpoint	Ja	Ja	Nein
AWS::NetworkFirewall::Firewall	Ja	Ja	Ja
AWS::NetworkFirewall::FirewallPolicy	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::NetworkFirewall::LoggingConfiguration	Ja	Ja	Nein
AWS::NetworkFirewall::RuleGroup	Ja	Ja	Ja
AWS::NetworkFirewall::TLSInspectionConfiguration	Ja	Ja	Nein
AWS::NetworkFirewall::VpcEndpointAssociation	Ja	Ja	Nein
AWS::NetworkManager::ConnectAttachment	Ja	Ja	Ja
AWS::NetworkManager::ConnectPeer	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::NetworkManager::CoreNetwork	Ja	Ja	Ja
AWS::NetworkManager::CustomerGatewayAssociation	Ja	Ja	Ja
AWS::NetworkManager::Device	Ja	Ja	Ja
AWS::NetworkManager::DirectConnectGatewayAttachment	Ja	Ja	Nein
AWS::NetworkManager::GlobalNetwork	Ja	Ja	Ja
AWS::NetworkManager::Link	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::NetworkManager::LinkAssociation	Ja	Ja	Ja
AWS::NetworkManager::Site	Ja	Ja	Ja
AWS::NetworkManager::SiteToSiteVpnAttachment	Ja	Ja	Ja
AWS::NetworkManager::TransitGatewayPeering	Ja	Ja	Ja
AWS::NetworkManager::TransitGatewayRegistration	Ja	Ja	Ja
AWS::NetworkManager::TransitGatewayRouteTableAttachment	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::NetworkManager::VpcAttachment	Ja	Ja	Ja
AWS::Notifications::ChannelAssociation	Ja	Ja	Nein
AWS::Notifications::EventRule	Ja	Ja	Nein
AWS::Notifications::ManagedNotificationAccountContactAssociation	Ja	Ja	Nein
AWS::Notifications::ManagedNotificationAdditionalChannelAssociation	Ja	Ja	Nein
AWS::Notifications::NotificationConfiguration	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Notifications::NotificationHub	Ja	Ja	Nein
AWS::NotificationsContacts::EmailContact	Ja	Ja	Nein
AWS::ODB::CloudAutonomousVmCluster	Ja	Ja	Nein
AWS::ODB::CloudExadataInfrastructure	Ja	Ja	Nein
AWS::ODB::CloudVmCluster	Ja	Ja	Nein
AWS::ODB::OdbNetwork	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::OSIS::Pipeline	Ja	Ja	Ja
AWS::Oam::Link	Ja	Ja	Ja
AWS::Oam::Sink	Ja	Ja	Ja
AWS::ObservabilityAdmin::OrganizationTelemetryRule	Ja	Ja	Nein
AWS::ObservabilityAdmin::TelemetryRule	Ja	Ja	Nein
AWS::Omics::AnnotationStore	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Omics::ReferenceStore	Ja	Ja	Ja
AWS::Omics::RunGroup	Ja	Ja	Ja
AWS::Omics::SequenceStore	Ja	Ja	Ja
AWS::Omics::VariantStore	Ja	Ja	Ja
AWS::Omics::Workflow	Ja	Ja	Ja
AWS::Omics::WorkflowVersion	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::OpenSearchServerless::AccessPolicy	Ja	Ja	Ja
AWS::OpenSearchServerless::Collection	Ja	Ja	Nein
AWS::OpenSearchServerless::Index	Ja	Ja	Nein
AWS::OpenSearchServerless::LifecyclePolicy	Ja	Ja	Nein
AWS::OpenSearchServerless::SecurityConfig	Ja	Ja	Ja
AWS::OpenSearchServerless::SecurityPolicy	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::OpenSearchServerless::VpcEndpoint	Ja	Ja	Ja
AWS::OpenSearchService::Application	Ja	Ja	Nein
AWS::OpenSearchService::Domain	Ja	Ja	Nein
AWS::Organizations::Account	Ja	Ja	Ja
AWS::Organizations::Organization	Ja	Ja	Ja
AWS::Organizations::OrganizationalUnit	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Organizations::Policy	Ja	Ja	Ja
AWS::Organizations::ResourcePolicy	Ja	Ja	Ja
AWS::PCAConnectorAD::Connector	Ja	Ja	Ja
AWS::PCAConnectorAD::DirectoryRegistration	Ja	Ja	Ja
AWS::PCAConnectorAD::ServicePrincipalName	Ja	Ja	Ja
AWS::PCAConnectorAD::Template	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::PCAConnectorAD::TemplateGroupAccessControlEntry	Ja	Ja	Ja
AWS::PCAConnectorSCEP::Challenge	Ja	Ja	Nein
AWS::PCAConnectorSCEP::Connector	Ja	Ja	Nein
AWS::PCS::Cluster	Ja	Ja	Nein
AWS::PCS::ComputeNodeGroup	Ja	Ja	Nein
AWS::PCS::Queue	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Panorama::ApplicationInstance	Ja	Ja	Ja
AWS::Panorama::Package	Ja	Ja	Ja
AWS::Panorama::PackageVersion	Ja	Ja	Nein
AWS::PaymentCryptography::Alias	Ja	Ja	Nein
AWS::PaymentCryptography::Key	Ja	Ja	Nein
AWS::Personalize::Dataset	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Personalize::DatasetGroup	Ja	Ja	Ja
AWS::Personalize::Schema	Ja	Ja	Ja
AWS::Personalize::Solution	Ja	Ja	Nein
AWS::Pinpoint::InAppTemplate	Ja	Ja	Ja
AWS::Pipes::Pipe	Ja	Ja	Ja
AWS::Proton::EnvironmentAccountConnection	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Proton::EnvironmentTemplate	Ja	Ja	Ja
AWS::Proton::ServiceTemplate	Ja	Ja	Ja
AWS::QBusiness::Application	Ja	Ja	Nein
AWS::QBusiness::DataAccessor	Ja	Ja	Nein
AWS::QBusiness::DataSource	Ja	Ja	Nein
AWS::QBusiness::Index	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::QBusiness::Permission	Ja	Ja	Nein
AWS::QBusiness::Plugin	Ja	Ja	Nein
AWS::QBusiness::Retriever	Ja	Ja	Nein
AWS::QBusiness::WebExperience	Ja	Ja	Nein
AWS::QLDB::Stream	Ja	Ja	Ja
AWS::QuickSight::Analysis	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::QuickSight::CustomPermissions	Ja	Ja	Nein
AWS::QuickSight::Dashboard	Ja	Ja	Nein
AWS::QuickSight::DataSet	Ja	Ja	Ja
AWS::QuickSight::DataSource	Ja	Ja	Ja
AWS::QuickSight::Folder	Ja	Ja	Nein
AWS::QuickSight::RefreshSchedule	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::QuickSight::Template	Ja	Ja	Nein
AWS::QuickSight::Theme	Ja	Ja	Nein
AWS::QuickSight::Topic	Ja	Ja	Ja
AWS::QuickSight::VPCConnection	Ja	Ja	Ja
AWS::RAM::Permission	Ja	Ja	Nein
AWS::RAM::ResourceShare	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::RDS::CustomDBEngineVersion	Ja	Ja	Nein
AWS::RDS::DBCluster	Ja	Ja	Ja
AWS::RDS::DBClusterParameterGroup	Ja	Ja	Nein
AWS::RDS::DBInstance	Ja	Nein	Ja
AWS::RDS::DBParameterGroup	Ja	Ja	Nein
AWS::RDS::DBProxy	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::RDS::DBProxyEndpoint	Ja	Ja	Ja
AWS::RDS::DBProxyTargetGroup	Ja	Ja	Ja
AWS::RDS::DBShardGroup	Ja	Ja	Nein
AWS::RDS::DBSubnetGroup	Ja	Ja	Ja
AWS::RDS::EventSubscription	Ja	Ja	Ja
AWS::RDS::GlobalCluster	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::RDS::Integration	Ja	Ja	Nein
AWS::RDS::OptionGroup	Ja	Ja	Ja
AWS::RUM::AppMonitor	Ja	Ja	Ja
AWS::Rbin::Rule	Ja	Ja	Nein
AWS::Redshift::Cluster	Ja	Ja	Ja
AWS::Redshift::ClusterParameterGroup	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Redshift::ClusterSubnetGroup	Ja	Ja	Nein
AWS::Redshift::EndpointAccess	Ja	Ja	Ja
AWS::Redshift::EndpointAuthorization	Ja	Ja	Ja
AWS::Redshift::EventSubscription	Ja	Ja	Nein
AWS::Redshift::Integration	Ja	Ja	Nein
AWS::Redshift::ScheduledAction	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::RedshiftServerless::Namespace	Ja	Ja	Nein
AWS::RedshiftServerless::Snapshot	Ja	Ja	Nein
AWS::RedshiftServerless::Workgroup	Ja	Ja	Nein
AWS::RefactorSpaces::Application	Ja	Ja	Ja
AWS::RefactorSpaces::Environment	Ja	Ja	Ja
AWS::RefactorSpaces::Route	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::RefactorSpaces::Service	Ja	Ja	Ja
AWS::Rekognition::Collection	Ja	Ja	Ja
AWS::Rekognition::Project	Ja	Ja	Ja
AWS::Rekognition::StreamProcessor	Ja	Ja	Ja
AWS::ResilienceHub::App	Ja	Ja	Ja
AWS::ResilienceHub::ResiliencyPolicy	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ResourceExplorer2::DefaultViewAssociation	Ja	Ja	Nein
AWS::ResourceExplorer2::Index	Ja	Ja	Ja
AWS::ResourceExplorer2::View	Ja	Ja	Ja
AWS::ResourceGroups::Group	Ja	Ja	Ja
AWS::ResourceGroups::TagSyncTask	Ja	Ja	Nein
AWS::RoboMaker::Fleet	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::RoboMaker::Robot	Ja	Ja	Nein
AWS::RoboMaker::RobotApplication	Ja	Ja	Nein
AWS::RoboMaker::RobotApplicationVersion	Ja	Ja	Nein
AWS::RoboMaker::SimulationApplication	Ja	Ja	Nein
AWS::RoboMaker::SimulationApplicationVersion	Ja	Ja	Nein
AWS::RolesAnywhere::CRL	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::RolesAnywhere::Profile	Ja	Ja	Ja
AWS::RolesAnywhere::TrustAnchor	Ja	Ja	Nein
AWS::Route53::CidrCollection	Ja	Ja	Ja
AWS::Route53::DNSSEC	Ja	Ja	Ja
AWS::Route53::HealthCheck	Ja	Ja	Ja
AWS::Route53::HostedZone	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Route53::KeySigningKey	Ja	Ja	Ja
AWS::Route53Profiles::Profile	Ja	Ja	Nein
AWS::Route53Profiles::ProfileAssociation	Ja	Ja	Nein
AWS::Route53Profiles::ProfileResourceAssociation	Ja	Ja	Nein
AWS::Route53RecoveryControl::Cluster	Ja	Ja	Ja
AWS::Route53RecoveryControl::ControlPanel	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Route53RecoveryControl::RoutingControl	Ja	Ja	Ja
AWS::Route53RecoveryControl::SafetyRule	Ja	Ja	Ja
AWS::Route53RecoveryReadiness::Cell	Ja	Ja	Ja
AWS::Route53RecoveryReadiness::ReadinessCheck	Ja	Ja	Ja
AWS::Route53RecoveryReadiness::RecoveryGroup	Ja	Ja	Ja
AWS::Route53RecoveryReadiness::ResourceSet	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Route53Resolver::FirewallDomainList	Ja	Ja	Nein
AWS::Route53Resolver::FirewallRuleGroup	Ja	Ja	Ja
AWS::Route53Resolver::FirewallRuleGroupAssociation	Ja	Ja	Ja
AWS::Route53Resolver::OutpostResolver	Ja	Ja	Ja
AWS::Route53Resolver::ResolverConfig	Ja	Ja	Ja
AWS::Route53Resolver::ResolverDNSSECConfig	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Route53Resolver::ResolverEndpoint	Ja	Ja	Nein
AWS::Route53Resolver::ResolverQueryLoggingConfig	Ja	Ja	Ja
AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation	Ja	Ja	Ja
AWS::Route53Resolver::ResolverRule	Ja	Ja	Nein
AWS::Route53Resolver::ResolverRuleAssociation	Ja	Ja	Ja
AWS::S3::AccessGrant	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::S3::AccessGrantsInstance	Ja	Ja	Nein
AWS::S3::AccessGrantsLocation	Ja	Ja	Nein
AWS::S3::AccessPoint	Ja	Ja	Ja
AWS::S3::Bucket	Ja	Ja	Ja
AWS::S3::BucketPolicy	Ja	Nein	Ja
AWS::S3::MultiRegionAccessPoint	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::S3::MultiRegionAccessPointPolicy	Ja	Nein	Nein
AWS::S3::StorageLens	Ja	Ja	Ja
AWS::S3::StorageLensGroup	Ja	Ja	Nein
AWS::S3Express::AccessPoint	Ja	Ja	Nein
AWS::S3Express::BucketPolicy	Ja	Ja	Nein
AWS::S3Express::DirectoryBucket	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::S3ObjectLambda::AccessPoint	Ja	Ja	Ja
AWS::S3ObjectLambda::AccessPointPolicy	Ja	Ja	Nein
AWS::S3Outposts::AccessPoint	Ja	Ja	Ja
AWS::S3Outposts::Bucket	Ja	Ja	Ja
AWS::S3Outposts::BucketPolicy	Ja	Ja	Nein
AWS::S3Outposts::Endpoint	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::S3Tables::Namespace	Ja	Ja	Nein
AWS::S3Tables::Table	Ja	Ja	Nein
AWS::S3Tables::TableBucket	Ja	Ja	Nein
AWS::S3Tables::TableBucketPolicy	Ja	Ja	Nein
AWS::S3Tables::TablePolicy	Ja	Ja	Nein
AWS::SES::ConfigurationSet	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SES::ConfigurationSetEventDestination	Ja	Ja	Nein
AWS::SES::ContactList	Ja	Ja	Nein
AWS::SES::DedicatedIpPool	Ja	Ja	Ja
AWS::SES::EmailIdentity	Ja	Ja	Ja
AWS::SES::MailManagerAddonInstance	Ja	Ja	Nein
AWS::SES::MailManagerAddonSubscription	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SES::MailManagerAddressList	Ja	Ja	Nein
AWS::SES::MailManagerArchive	Ja	Ja	Nein
AWS::SES::MailManagerIngressPoint	Ja	Ja	Nein
AWS::SES::MailManagerRelay	Ja	Ja	Nein
AWS::SES::MailManagerRuleSet	Ja	Ja	Nein
AWS::SES::MailManagerTrafficPolicy	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SES::Template	Ja	Ja	Nein
AWS::SES::VdmAttributes	Ja	Ja	Nein
AWS::SNS::Subscription	Ja	Ja	Nein
AWS::SNS::Topic	Ja	Ja	Ja
AWS::SNS::TopicInlinePolicy	Ja	Ja	Nein
AWS::SQS::Queue	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SQS::QueueInlinePolicy	Ja	Ja	Nein
AWS::SSM::Association	Ja	Ja	Ja
AWS::SSM::Document	Ja	Nein	Nein
AWS::SSM::Parameter	Ja	Ja	Nein
AWS::SSM::PatchBaseline	Ja	Ja	Nein
AWS::SSM::ResourceDataSync	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SSM::ResourcePolicy	Ja	Ja	Ja
AWS::SSMContacts::Contact	Ja	Ja	Ja
AWS::SSMContacts::ContactChannel	Ja	Ja	Ja
AWS::SSMContacts::Plan	Ja	Ja	Nein
AWS::SSMContacts::Rotation	Ja	Ja	Nein
AWS::SSMGuiConnect::Preferences	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SSMIncidents::ReplicationSet	Ja	Ja	Ja
AWS::SSMIncidents::ResponsePlan	Ja	Ja	Ja
AWS::SSMQuickSetup::ConfigurationManager	Ja	Ja	Nein
AWS::SSO::Application	Ja	Ja	Nein
AWS::SSO::ApplicationAssignment	Ja	Ja	Nein
AWS::SSO::Assignment	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SSO::Instance	Ja	Ja	Nein
AWS::SSO::InstanceAccessControlAttributeConfiguration	Ja	Ja	Ja
AWS::SageMaker::App	Ja	Ja	Nein
AWS::SageMaker::AppImageConfig	Ja	Ja	Nein
AWS::SageMaker::Cluster	Ja	Ja	Nein
AWS::SageMaker::DataQualityJobDefinition	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SageMaker::Device	Ja	Ja	Nein
AWS::SageMaker::DeviceFleet	Ja	Ja	Nein
AWS::SageMaker::Domain	Ja	Ja	Nein
AWS::SageMaker::Endpoint	Nein	Ja	Nein
AWS::SageMaker::FeatureGroup	Ja	Ja	Ja
AWS::SageMaker::Image	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SageMaker::ImageVersion	Ja	Ja	Ja
AWS::SageMaker::InferenceComponent	Ja	Ja	Nein
AWS::SageMaker::InferenceExperiment	Ja	Ja	Ja
AWS::SageMaker::MlflowTrackingServer	Ja	Ja	Nein
AWS::SageMaker::ModelBiasJobDefinition	Ja	Ja	Nein
AWS::SageMaker::ModelCard	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SageMaker::ModelExplainabilityJobDefinition	Ja	Ja	Nein
AWS::SageMaker::ModelPackage	Ja	Ja	Ja
AWS::SageMaker::ModelPackageGroup	Ja	Ja	Ja
AWS::SageMaker::ModelQualityJobDefinition	Ja	Ja	Nein
AWS::SageMaker::MonitoringSchedule	Ja	Ja	Nein
AWS::SageMaker::PartnerApp	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SageMaker::Pipeline	Ja	Ja	Ja
AWS::SageMaker::ProcessingJob	Ja	Ja	Nein
AWS::SageMaker::Project	Ja	Ja	Ja
AWS::SageMaker::Space	Ja	Ja	Nein
AWS::SageMaker::StudioLifecycleConfig	Ja	Ja	Nein
AWS::SageMaker::UserProfile	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Scheduler::Schedule	Ja	Ja	Ja
AWS::Scheduler::ScheduleGroup	Ja	Ja	Nein
AWS::SecretsManager::ResourcePolicy	Ja	Nein	Nein
AWS::SecretsManager::RotationSchedule	Ja	Ja	Nein
AWS::SecretsManager::Secret	Ja	Ja	Ja
AWS::SecretsManager::SecretTargetAttachment	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SecurityHub::AggregatorV2	Ja	Ja	Nein
AWS::SecurityHub::AutomationRule	Ja	Ja	Nein
AWS::SecurityHub::AutomationRuleV2	Ja	Ja	Nein
AWS::SecurityHub::ConfigurationPolicy	Ja	Ja	Nein
AWS::SecurityHub::DelegatedAdmin	Ja	Ja	Nein
AWS::SecurityHub::FindingAggregator	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SecurityHub::Hub	Ja	Ja	Nein
AWS::SecurityHub::HubV2	Ja	Ja	Nein
AWS::SecurityHub::Insight	Ja	Ja	Nein
AWS::SecurityHub::OrganizationConfiguration	Ja	Ja	Nein
AWS::SecurityHub::PolicyAssociation	Ja	Ja	Nein
AWS::SecurityHub::ProductSubscription	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SecurityHub::SecurityControl	Ja	Ja	Nein
AWS::SecurityHub::Standard	Ja	Ja	Nein
AWS::SecurityLake::AwsLogSource	Ja	Ja	Nein
AWS::SecurityLake::DataLake	Ja	Ja	Nein
AWS::SecurityLake::Subscriber	Ja	Ja	Nein
AWS::SecurityLake::SubscriberNotification	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ServiceCatalog::CloudFormationProvisionedProduct	Ja	Nein	Nein
AWS::ServiceCatalog::ServiceAction	Ja	Ja	Ja
AWS::ServiceCatalog::ServiceActionAssociation	Ja	Ja	Nein
AWS::ServiceCatalogAppRegistry::Application	Ja	Ja	Ja
AWS::ServiceCatalogAppRegistry::AttributeGroup	Ja	Ja	Ja
AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::ServiceCatalogAppRegistry::ResourceAssociation	Ja	Nein	Nein
AWS::Shield::DRTAccess	Ja	Ja	Nein
AWS::Shield::ProactiveEngagement	Ja	Ja	Nein
AWS::Shield::Protection	Ja	Ja	Nein
AWS::Shield::ProtectionGroup	Ja	Ja	Nein
AWS::Signer::ProfilePermission	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Signer::SigningProfile	Ja	Ja	Ja
AWS::SimSpaceWeaver::Simulation	Ja	Ja	Ja
AWS::StepFunctions::Activity	Ja	Ja	Nein
AWS::StepFunctions::StateMachine	Ja	Ja	Ja
AWS::StepFunctions::StateMachineAlias	Ja	Ja	Ja
AWS::StepFunctions::StateMachineVersion	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::SupportApp::AccountAlias	Ja	Ja	Ja
AWS::SupportApp::SlackChannelConfiguration	Ja	Ja	Ja
AWS::SupportApp::SlackWorkspaceConfiguration	Ja	Ja	Ja
AWS::Synthetics::Canary	Ja	Ja	Ja
AWS::Synthetics::Group	Ja	Ja	Ja
AWS::SystemsManagerSAP::Application	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Timestream::Database	Ja	Ja	Ja
AWS::Timestream::InfluxDBInstance	Ja	Ja	Nein
AWS::Timestream::ScheduledQuery	Ja	Ja	Nein
AWS::Timestream::Table	Ja	Ja	Ja
AWS::Transfer::Agreement	Ja	Ja	Ja
AWS::Transfer::Certificate	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Transfer::Connector	Ja	Ja	Ja
AWS::Transfer::Profile	Ja	Ja	Ja
AWS::Transfer::Server	Ja	Ja	Nein
AWS::Transfer::User	Ja	Ja	Nein
AWS::Transfer::WebApp	Ja	Ja	Nein
AWS::Transfer::Workflow	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::VerifiedPermissions::IdentitySource	Ja	Ja	Ja
AWS::VerifiedPermissions::Policy	Ja	Ja	Ja
AWS::VerifiedPermissions::PolicyStore	Ja	Ja	Ja
AWS::VerifiedPermissions::PolicyTemplate	Ja	Ja	Ja
AWS::VoiceID::Domain	Ja	Ja	Nein
AWS::VpcLattice::AccessLogSubscription	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::VpcLattice::AuthPolicy	Ja	Ja	Nein
AWS::VpcLattice::Listener	Ja	Ja	Ja
AWS::VpcLattice::ResourceConfiguration	Ja	Ja	Nein
AWS::VpcLattice::ResourceGateway	Ja	Ja	Nein
AWS::VpcLattice::ResourcePolicy	Ja	Ja	Nein
AWS::VpcLattice::Rule	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::VpcLattice::Service	Ja	Ja	Ja
AWS::VpcLattice::ServiceNetwork	Ja	Ja	Ja
AWS::VpcLattice::ServiceNetworkResourceAssociation	Ja	Ja	Nein
AWS::VpcLattice::ServiceNetworkServiceAssociation	Ja	Ja	Ja
AWS::VpcLattice::ServiceNetworkVpcAssociation	Ja	Ja	Ja
AWS::VpcLattice::TargetGroup	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::WAFv2::IPSet	Ja	Ja	Ja
AWS::WAFv2::LoggingConfiguration	Ja	Ja	Ja
AWS::WAFv2::RegexPatternSet	Ja	Ja	Ja
AWS::WAFv2::RuleGroup	Ja	Ja	Ja
AWS::WAFv2::WebACL	Ja	Ja	Ja
AWS::WAFv2::WebACLAssociation	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Wisdom::AIAgent	Ja	Ja	Nein
AWS::Wisdom::AIAgentVersion	Ja	Ja	Nein
AWS::Wisdom::AIGuardrail	Ja	Ja	Nein
AWS::Wisdom::AIGuardrailVersion	Ja	Ja	Nein
AWS::Wisdom::AIPrompt	Ja	Ja	Nein
AWS::Wisdom::AIPromptVersion	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::Wisdom::Assistant	Ja	Ja	Ja
AWS::Wisdom::AssistantAssociation	Ja	Ja	Ja
AWS::Wisdom::KnowledgeBase	Ja	Ja	Ja
AWS::Wisdom::MessageTemplate	Ja	Ja	Nein
AWS::Wisdom::MessageTemplateVersion	Ja	Ja	Nein
AWS::Wisdom::QuickResponse	Ja	Ja	Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::WorkSpaces::ConnectionAlias	Ja	Ja	Nein
AWS::WorkSpaces::WorkspacesPool	Ja	Ja	Nein
AWS::WorkSpacesThinClient::Environment	Ja	Ja	Nein
AWS::WorkSpacesWeb::BrowserSettings	Ja	Ja	Ja
AWS::WorkSpacesWeb::DataProtectionSettings	Ja	Ja	Nein
AWS::WorkSpacesWeb::IdentityProvider	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::WorkSpacesWeb::IpAccessSettings	Ja	Ja	Ja
AWS::WorkSpacesWeb::NetworkSettings	Ja	Ja	Ja
AWS::WorkSpacesWeb::Portal	Ja	Ja	Ja
AWS::WorkSpacesWeb::SessionLogger	Ja	Ja	Nein
AWS::WorkSpacesWeb::TrustStore	Ja	Ja	Ja
AWS::WorkSpacesWeb::UserAccessLoggingSettings	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::WorkSpacesWeb::UserSettings	Ja	Ja	Ja
AWS::WorkspacesInstances::Volume	Ja	Ja	Nein
AWS::WorkspacesInstances::VolumeAssociation	Ja	Ja	Nein
AWS::WorkspacesInstances::WorkspaceInstance	Ja	Ja	Nein
AWS::XRay::Group	Ja	Ja	Ja
AWS::XRay::ResourcePolicy	Ja	Ja	Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
AWS::XRay::SamplingRule	Ja	Ja	Nein
AWS::XRay::TransactionSearchConfig	Ja	Ja	Nein

Verwenden Sie Schnellerstellungslinks, um Stapel zu erstellen CloudFormation

Schnellerstellungslinks bieten eine optimierte Methode, um CloudFormation Stacks direkt von der Konsole aus zu starten. URLs CloudFormation Indem Sie die Vorlagen-URL, den Stacknamen und die Vorlagenparameter als URL-Abfrageparameter angeben, können Sie eine einzelne Seite „Stack erstellen“ vorab füllen und so die Stack-Erstellung beschleunigen. Dies vereinfacht das Erstellen von Stacks, da sowohl die Anzahl der Assistentenseiten als auch die Menge der erforderlichen Benutzereingaben reduziert werden. Außerdem wird die Wiederverwendung von Vorlagen optimiert, da Sie mehrere Vorlagen erstellen können URLs , die unterschiedliche Werte für dieselbe Vorlage angeben.

URL-Format

Der Link zur Schnellerstellung folgt diesem URL-Format:

```
https://region-code.console.aws.amazon.com/cloudformation/home?region=region-code#/
stacks/create/review
   ?templateURL=TemplateURL
   &stackName=StackName
   &param_parameterName=parameterValue
```

CloudFormation unterstützt die folgenden URL-Abfrageparameter:

Vorlagen-URL

Erforderlich Der `templateURL` Parameter gibt die URL der Stack-Vorlage an, die sich in einem Amazon S3 S3-Bucket befindet. Um Zugriffsprobleme mit einer vorsignierten S3-URL zu vermeiden, stellen Sie sicher, dass Sie die URL URL-kodieren.

Unterstützte S3-URL-Formate:

- `https://s3.region-code.amazonaws.com/bucket-name/template-name`
- `https://bucket-name.s3.region-code.amazonaws.com/template-name`
- `https://s3-region-code.amazonaws.com/bucket-name/template-name`(Legacy-Format)

Stack name

Optional. Verwenden Sie den `stackName` Parameter, um den Namen des zu erstellenden CloudFormation Stacks anzugeben. 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.

Vorlagenparameter

Optional. Verwenden Sie für Parameter in der Stack-Vorlage, die kein `NoEcho` Parametertyp sind, das Format `param_parameterName` in der URL-Abfragezeichenfolge. Der URL-Parameter muss das `param_`-Präfix enthalten und das Parameternamensegment muss genau mit dem Parameternamen in der Vorlage übereinstimmen. Beispiel: `param_DBName`.

CloudFormation ignoriert Parameter, die in der Vorlage nicht vorhanden sind, und alle Parameter, deren `NoEcho` Eigenschaft auf `true` Typen gesetzt ist (normalerweise Benutzernamen und Kennwörter). URL-Parameter überschreiben in der Vorlage angegebene Standardwerte. Sie können so viele Parameter wie nötig angeben.

> **Important**
>
> Anstatt vertrauliche Informationen direkt in Ihre CloudFormation Vorlagen einzubetten, empfehlen wir Ihnen, dynamische Parameter in der Stack-Vorlage zu verwenden, um auf vertrauliche Informationen zu verweisen, die außerhalb von gespeichert und verwaltet werden CloudFormation, z. B. im AWS Systems Manager Parameterspeicher oder. AWS Secrets Manager

Weitere Informationen finden Sie in den bewährten Methoden zu Keine Anmeldeinformationen in Ihre Vorlagen einbetten.

Für alle Abfrageparameternamen wird die Groß- und Kleinschreibung berücksichtigt. Benutzer können diese Werte in der Konsole überschreiben, bevor der Stack erstellt wird.

Beispiel

Das folgende Beispiel basiert auf der WordPress einfachen Einzelinstanz-Beispielvorlage. Die Abfragezeichenfolge enthält die erforderlichen `templateURL` Parameter und die Parameter `stackName`, `DBName`, `InstanceType` und `KeyName`.

Die folgende URL enthält der Übersichtlichkeit halber Zeilenumbrüche.

```
https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/
create/review
   ?templateURL=https://s3.us-east-2.amazonaws.com/cloudformation-templates-us-east-2/
WordPress_Single_Instance.template
   &stackName=MyWPBlog
   &param_DBName=mywpblog
   &param_InstanceType=t2.medium
```

Der folgende URL enthält dieselben Parameter wie im vorherigen Beispiel, aber die Zeilenumbrüche wurden entfernt. Dies ist das tatsächliche URL-Format.

```
https://us-east-2.console.aws.amazon.com/cloudformation/home?
region=us-east-2#/stacks/create/review?templateURL=https://
s3.us-east-2.amazonaws.com/cloudformation-templates-us-east-2/
WordPress_Single_Instance.template&stackName=MyWPBlog&param_DBName=mywpblog&param_InstanceType=
```

Erstellen eines Stacks mithilfe eines Schnellerstellungslinks

Wenn Sie einen Link zur Schnellerstellung öffnen, werden Sie zur Konsole weitergeleitet. CloudFormation In der Konsole wird direkt die Seite Quick Create Stack geöffnet, auf der die angegebenen Werte automatisch für die Parameter verwendet werden.

So erstellen Sie einen Stack mithilfe eines Schnellerstellungslinks (Konsole)

1. Vergewissern Sie sich auf der Seite Quick Create Stack für Vorlage, Vorlagen-URL, dass die Vorlagen-URL korrekt ist.
2. Erweitern Sie den Abschnitt Vorlage anzeigen, um die Vorlage zu überprüfen.
3. Überprüfen Sie unter Stackname den vorab ausgefüllten Stacknamen.
4. Sehen Sie sich den Abschnitt Parameter an. Stellen Sie sicher, dass die vorab ausgefüllten Parameterwerte korrekt sind. Geben Sie alle erforderlichen Parameter ein, die nicht in der URL angegeben wurden. Ändern Sie bei Bedarf alle vorab ausgefüllten Werte.
5. Als Nächstes können Sie die folgenden Einstellungen konfigurieren:

   - Tags — Organisieren Sie Ressourcen mit Schlüssel-Wert-Paaren.
   - Berechtigungen — Wählen Sie die IAM-Servicerolle für Stack-Operationen aus.
   - Optionen für Stack-Fehler — Wählen Sie, ob Sie ein Rollback (Standard) durchführen oder Ressourcen beibehalten möchten.
   - Stack-Richtlinie — Steuern Sie die Berechtigungen zur Aktualisierung von Ressourcen.
   - Rollback-Konfiguration — Konfigurieren Sie ein CloudWatch alarmbasiertes Rollback.
   - Benachrichtigungsoptionen — Richten Sie Amazon SNS SNS-Benachrichtigungen für Stack-Ereignisse ein.
   - Optionen für die Stack-Erstellung — Definieren Sie die maximale Zeit für die Stack-Erstellung und aktivieren Sie den Kündigungsschutz, um versehentliche Löschungen zu verhindern.

   Weitere Informationen finden Sie unter Konfigurieren Sie die Stack-Optionen.
6. Füllen Sie für Capabilities alle erforderlichen Bestätigungen aus. Wenn Ihre Vorlage beispielsweise IAM-Ressourcen enthält, wählen Sie Ich bestätige, dass diese Vorlage IAM-Ressourcen erstellen kann aus, um anzugeben, dass Sie IAM-Ressourcen in der Vorlage verwenden möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.
7. (Optional) Sie können einen Änderungssatz erstellen, um eine Vorschau der Konfiguration des Stacks anzuzeigen, bevor Sie ihn erstellen. Wählen Sie Änderungssatz erstellen und folgen Sie den Anweisungen. Weitere Informationen finden Sie unter Sehen Sie sich eine Vorschau der Konfiguration Ihres Stacks an.

8. Wenn Sie bereit sind, wählen Sie Stack erstellen aus, um den Stack zu starten, und überwachen Sie dann den Fortschritt der Stack-Erstellung auf der Registerkarte Ereignisse. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

Beispiele für CloudFormation Stack-Operationsbefehle für AWS CLI und PowerShell

Die folgenden Befehlszeilenbeispiele zeigen, wie einzelne CloudFormation Aktionen mit dem AWS CLI und PowerShell ausgeführt werden. Diese Beispiele beinhalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie cloudformationin der AWS CLI Befehlsreferenz.

Die Beispiele in diesem Handbuch verwenden die Konvention eines umgekehrten Schrägstrichs (\), um anzuzeigen, dass eine lange Befehlszeile in der nächsten Zeile fortgesetzt wird.

Themen

- Brechen Sie ein Stack-Update ab
- Rollback einer Aktualisierung fortsetzen
- Erstellen eines Stacks
- Erstellen Sie einen Stapel, der Transformationen enthält
- Löschen eines Stacks
- Beschreiben Sie Stack-Ereignisse
- Beschreiben Sie eine Stack-Ressource
- Beschreiben Sie Stack-Ressourcen
- Beschreiben Sie Stapel
- Holen Sie sich eine Vorlage
- Stack-Ressourcen auflisten
- Stapel auflisten
- Einen Stack aktualisieren
- Bestätigen Sie Ihre Vorlage
- Laden Sie lokale Artefakte in einen S3-Bucket hoch mit dem AWS CLI

Brechen Sie ein Stack-Update ab

Verwenden Sie den cancel-update-stackBefehlsbefehl, um ein Stack-Update abzubrechen. Weitere Informationen finden Sie unter Ein Stack-Update abbrechen.

CLI

AWS CLI

Um ein laufendes Stack-Update abzubrechen

Mit dem folgenden `cancel-update-stack` Befehl wird ein Stack-Update auf dem `myteststack` Stack abgebrochen:

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- Einzelheiten zur API finden Sie CancelUpdateStackin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Bricht ein Update auf dem angegebenen Stack ab.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter CancelUpdateStack AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Bricht ein Update auf dem angegebenen Stack ab.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter CancelUpdateStack AWS -Tools für PowerShellCmdlet-Referenz (V5).

Rollback einer Aktualisierung fortsetzen

Verwenden Sie den continue-update-rollbackBefehl, um mit dem Rollback eines Updates fortzufahren. Weitere Informationen finden Sie unter Rollback einer Aktualisierung fortsetzen.

CLI

AWS CLI

Um erneut zu versuchen, ein Update rückgängig zu machen

Im folgenden `continue-update-rollback` Beispiel wird ein Rollback-Vorgang nach einem zuvor fehlgeschlagenen Stack-Update wieder aufgenommen.

```
aws cloudformation continue-update-rollback \
    --stack-name my-stack
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- Einzelheiten zur API finden Sie unter ContinueUpdateRollback AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Setzt das Rollback des benannten Stacks fort, der sich im Status 'UPDATE_ROLLBACK_FAILED' befinden sollte. Wenn das fortgesetzte Rollback erfolgreich ist, wechselt der Stack in den Status „UPDATE_ROLLBACK_COMPLETE“.

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter Cmdlet-Referenz (V4). ContinueUpdateRollbackAWS - Tools für PowerShell

Tools für V5 PowerShell

Beispiel 1: Setzt das Rollback des benannten Stacks fort, der sich im Status 'UPDATE_ROLLBACK_FAILED' befinden sollte. Wenn das fortgesetzte Rollback erfolgreich ist, wechselt der Stack in den Status „UPDATE_ROLLBACK_COMPLETE“.

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter Cmdlet-Referenz (V5). ContinueUpdateRollbackAWS - Tools für PowerShell

Erstellen eines Stacks

Verwenden Sie den create-stackBefehl, um einen Stack zu erstellen. Sie müssen den Stack-Namen, den Speicherort einer gültigen Vorlage und alle Eingabeparameter angeben. Bei den Namen der Parameterschlüssel wird zwischen Groß- und Kleinschreibung unterschieden. Wenn Sie den Namen eines Parameterschlüssels falsch eingeben, wird der Stapel CloudFormation nicht erstellt und es wird gemeldet, dass die Vorlage diesen Parameter nicht enthält.

Die folgenden Beispiele zeigen, wie ein neuer Stack mit dem angegebenen Namen, der Vorlage und den Eingabeparametern erstellt wird.

CLI

AWS CLI

Um einen AWS CloudFormation Stapel zu erstellen

Der folgende `create-stacks` Befehl erstellt `myteststack` mithilfe der `sampletemplate.json` Vorlage einen Stack mit dem Namen:

```
aws cloudformation create-stack --stack-name myteststack --template-body file://
sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
 ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

Ausgabe:

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

Weitere Informationen finden Sie unter Stacks im AWS CloudFormation Benutzerhandbuch.

- Einzelheiten zur API finden Sie CreateStackin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Erzeugt einen neuen Stack mit dem angegebenen Namen. Die Vorlage wird anhand des bereitgestellten Inhalts mit Anpassungsparametern analysiert ('PK1' und 'PK2' stehen für die Namen der im Vorlageninhalt deklarierten Parameter, 'PV1' und 'PV2' stehen für die Werte für diese Parameter. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und 'ParameterKey' angegeben werden. ParameterValue Wenn die Erstellung des Stacks fehlschlägt, wird er nicht zurückgesetzt.

```
New-CFNStack -StackName "myStack" `
             -TemplateBody "{TEMPLATE CONTENT HERE}" `
             -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
 @{ ParameterKey="PK2"; ParameterValue="PV2" }) `
             -DisableRollback $true
```

Beispiel 2: Erzeugt einen neuen Stack mit dem angegebenen Namen. Die Vorlage wird anhand des bereitgestellten Inhalts mit Anpassungsparametern analysiert ('PK1' und 'PK2' stehen für die Namen der im Vorlageninhalt deklarierten Parameter, 'PV1' und 'PV2' stehen für die Werte für diese Parameter. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und 'ParameterKey' angegeben werden. ParameterValue Wenn die Erstellung des Stacks fehlschlägt, wird er zurückgesetzt.

```
$p1 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p1.ParameterKey = "PK1"
$p1.ParameterValue = "PV1"

$p2 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p2.ParameterKey = "PK2"
$p2.ParameterValue = "PV2"

New-CFNStack -StackName "myStack" `
             -TemplateBody "{TEMPLATE CONTENT HERE}" `
             -Parameter @( $p1, $p2 ) `
             -OnFailure "ROLLBACK"
```

Beispiel 3: Erzeugt einen neuen Stack mit dem angegebenen Namen. Die Vorlage wird von der Amazon S3 S3-URL mit Anpassungsparametern abgerufen ('PK1' steht für den Namen eines Parameters, der im Inhalt der Vorlage deklariert ist, PV1 '' steht für den Wert für den

Parameter. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von " und ParameterKey 'ParameterValue' angegeben werden. Wenn die Erstellung des Stacks fehlschlägt, wird er zurückgesetzt (genauso wie bei der Angabe von - DisableRollback $false).

```
New-CFNStack -StackName "myStack" `
             -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
             -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Beispiel 4: Erzeugt einen neuen Stack mit dem angegebenen Namen. Die Vorlage wird von der Amazon S3 S3-URL mit Anpassungsparametern abgerufen ('PK1' steht für den Namen eines Parameters, der im Inhalt der Vorlage deklariert ist, PV1 " steht für den Wert für den Parameter. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von " und ParameterKey 'ParameterValue' angegeben werden. Wenn die Erstellung des Stacks fehlschlägt, wird er zurückgesetzt (genauso wie bei der Angabe von - DisableRollback $false). Die angegebene Benachrichtigung AENs empfängt veröffentlichte Ereignisse im Zusammenhang mit dem Stack.

```
New-CFNStack -StackName "myStack" `
             -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
             -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
             -NotificationARN @( "arn1", "arn2" )
```

- Einzelheiten zur API finden Sie unter CreateStack AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Erzeugt einen neuen Stack mit dem angegebenen Namen. Die Vorlage wird anhand des bereitgestellten Inhalts mit Anpassungsparametern analysiert ('PK1' und 'PK2' stehen für die Namen der im Vorlageninhalt deklarierten Parameter, 'PV1' und 'PV2' stehen für die Werte für diese Parameter. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von " und 'ParameterKey' angegeben werden. ParameterValue Wenn die Erstellung des Stacks fehlschlägt, wird er nicht zurückgesetzt.

```
New-CFNStack -StackName "myStack" `
             -TemplateBody "{TEMPLATE CONTENT HERE}" `
             -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
 @{ ParameterKey="PK2"; ParameterValue="PV2" }) `
```

```
             -DisableRollback $true
```

Beispiel 2: Erzeugt einen neuen Stack mit dem angegebenen Namen. Die Vorlage wird anhand des bereitgestellten Inhalts mit Anpassungsparametern analysiert ('PK1' und 'PK2' stehen für die Namen der im Vorlageninhalt deklarierten Parameter, 'PV1' und 'PV2' stehen für die Werte für diese Parameter. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und 'ParameterKey' angegeben werden. ParameterValue Wenn die Erstellung des Stacks fehlschlägt, wird er zurückgesetzt.

```
$p1 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p1.ParameterKey = "PK1"
$p1.ParameterValue = "PV1"

$p2 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p2.ParameterKey = "PK2"
$p2.ParameterValue = "PV2"

New-CFNStack -StackName "myStack" `
             -TemplateBody "{TEMPLATE CONTENT HERE}" `
             -Parameter @( $p1, $p2 ) `
             -OnFailure "ROLLBACK"
```

Beispiel 3: Erzeugt einen neuen Stack mit dem angegebenen Namen. Die Vorlage wird von der Amazon S3 S3-URL mit Anpassungsparametern abgerufen ('PK1' steht für den Namen eines Parameters, der im Inhalt der Vorlage deklariert ist, PV1 '' steht für den Wert für den Parameter. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden. Wenn die Erstellung des Stacks fehlschlägt, wird er zurückgesetzt (genauso wie bei der Angabe von - DisableRollback $false).

```
New-CFNStack -StackName "myStack" `
             -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
             -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Beispiel 4: Erzeugt einen neuen Stack mit dem angegebenen Namen. Die Vorlage wird von der Amazon S3 S3-URL mit Anpassungsparametern abgerufen ('PK1' steht für den Namen eines Parameters, der im Inhalt der Vorlage deklariert ist, PV1 '' steht für den Wert für den Parameter. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden. Wenn die Erstellung des Stacks fehlschlägt, wird er zurückgesetzt (genauso wie bei der Angabe von - DisableRollback

$false). Die angegebene Benachrichtigung AENs empfängt veröffentlichte Ereignisse im Zusammenhang mit dem Stack.

```
New-CFNStack -StackName "myStack" `
             -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
             -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
             -NotificationARN @( "arn1", "arn2" )
```

- Einzelheiten zur API finden Sie unter CreateStack AWS -Tools für PowerShellCmdlet-Referenz (V5).

> **Note**
>
> Sie können die AWS CLI `--template-url` Option verwenden, um einen Speicherort für die Vorlagendatei in Amazon S3 oder anzugeben AWS Systems Manager.
> Für Amazon S3 muss die URL mit beginnen`https://`. Statische S3-Websites URLs werden nicht unterstützt.
>
> ```
> --template-url https://s3.region-code.amazonaws.com/bucket-name/template-name
> ```
>
> Verwenden AWS Systems Manager Sie für das folgende Format:
>
> ```
> --template-url "ssm-doc://arn:aws:ssm:region-code:account-id:document/document-name"
> ```

Erstellen Sie einen Stapel, der Transformationen enthält

Verwenden Sie den deployBefehl, um einen Stapel zu erstellen, der Transformationen enthält. Wenn Sie einen Stapel aus einer Vorlage erstellen, die Transformationen enthält, müssen Sie einen Änderungssatz verwenden. Der `deploy` Befehl kombiniert zwei Schritte (Erstellen eines Änderungssatzes und dessen Ausführung) in einem einzigen Befehl.

AWS CLI

Der folgende `deploy` Befehl erstellt einen Stack mit dem angegebenen Namen, der Vorlage und den Eingabeparametern.

```
aws cloudformation deploy --stack-name myteststack \
  --template /path_to_template/my-template.json \
  --parameter-overrides Key1=Value1 Key2=Value2
```

Löschen eines Stacks

Verwenden Sie den delete-stackBefehl, um einen Stapel zu löschen. Weitere Informationen finden Sie unter Löschen eines Stacks.

CLI

AWS CLI

Um einen Stapel zu löschen

Im folgenden `delete-stack` Beispiel wird der angegebene Stapel gelöscht.

```
aws cloudformation delete-stack \
    --stack-name my-stack
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- Einzelheiten zur API finden Sie DeleteStackin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Löscht den angegebenen Stack.

```
Remove-CFNStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter DeleteStack AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Löscht den angegebenen Stapel.

```
Remove-CFNStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter DeleteStack AWS -Tools für PowerShellCmdlet-Referenz (V5).

Wenn das Löschen fehlschlägt und ein `DELETE_FAILED` Status zurückgegeben wird, können Sie den Stapel mithilfe der Option gewaltsam löschen. `--deletion-mode` Dies sind die folgenden Werte, die mit verwendet werden können`deletion-mode`:

- `STANDARD`: Löscht den Stapel normal. Dies ist der Standard-Löschmodus.
- `FORCE_DELETE_STACK`: Löscht den Stapel und überspringt alle Ressourcen, die nicht gelöscht werden können.

AWS CLI

Die folgende `delete-stack` Befehlszeile löscht den *`myteststack`* Stack unter Verwendung des `FORCE_DELETE_STACK` Werts mit dem Parameter: `deletion-mode`

```
aws cloudformation delete-stack --stack-name myteststack \
    --deletion-mode FORCE_DELETE_STACK
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Nach der Verwendung können Sie den `list-stack-resources` Befehl verwenden`FORCE_DELETE_STACK`, um die Ressourcen aufzulisten, die beim Löschen des Stacks übersprungen wurden. Für die zurückbehaltenen Ressourcen wird der Status DELETE_SKIPPED angezeigt. Weitere Informationen finden Sie unter Stack-Ressourcen auflisten.

Beschreiben Sie Stack-Ereignisse

Verwenden Sie den describe-stack-eventsBefehl, um Stack-Ereignisse zu beschreiben. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

CLI

AWS CLI

Um Stack-Ereignisse zu beschreiben

Im folgenden `describe-stack-events` Beispiel werden die 2 neuesten Ereignisse für den angegebenen Stack angezeigt.

```
aws cloudformation describe-stack-events \
    --stack-name my-stack \
    --max-items 2

{
    "StackEvents": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "EventId": "4e1516d0-e4d6-xmpl-b94f-0a51958a168c",
            "StackName": "my-stack",
            "LogicalResourceId": "my-stack",
            "PhysicalResourceId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2019-10-02T05:34:29.556Z",
            "ResourceStatus": "UPDATE_COMPLETE"
        },
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "EventId": "4dd3c810-e4d6-xmpl-bade-0aaf8b31ab7a",
            "StackName": "my-stack",
            "LogicalResourceId": "my-stack",
            "PhysicalResourceId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2019-10-02T05:34:29.127Z",
            "ResourceStatus": "UPDATE_COMPLETE_CLEANUP_IN_PROGRESS"
        }
    ],

 "NextToken": "eyJOZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

- Einzelheiten zur API finden Sie DescribeStackEventsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt alle stapelbezogenen Ereignisse für den angegebenen Stack zurück.

```
Get-CFNStackEvent -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter DescribeStackEvents AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt alle stapelbezogenen Ereignisse für den angegebenen Stack zurück.

```
Get-CFNStackEvent -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter DescribeStackEvents AWS -Tools für PowerShellCmdlet-Referenz (V5).

Beschreiben Sie eine Stack-Ressource

CLI

AWS CLI

Um Informationen über eine Stack-Ressource zu erhalten

Im folgenden `describe-stack-resource` Beispiel werden Details für die Ressource angezeigt, die `MyFunction` im angegebenen Stack benannt ist.

```
aws cloudformation describe-stack-resource \
    --stack-name MyStack \
    --logical-resource-id MyFunction
```

Ausgabe:

```
{
    "StackResourceDetail": {
        "StackName": "MyStack",
        "StackId": "arn:aws:cloudformation:us-east-2:123456789012:stack/MyStack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
        "LogicalResourceId": "MyFunction",
        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
        "ResourceType": "AWS::Lambda::Function",
        "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
        "ResourceStatus": "UPDATE_COMPLETE",
```

```
        "Metadata": "{}",
        "DriftInformation": {
            "StackResourceDriftStatus": "IN_SYNC"
        }
    }
}
```

- Einzelheiten zur API finden Sie DescribeStackResourceunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt die Beschreibung einer Ressource zurück, die in der Vorlage identifiziert wurde, die dem angegebenen Stack durch die logische ID DBInstance „My“ zugeordnet ist.

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Einzelheiten zur API finden Sie unter DescribeStackResource AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt die Beschreibung einer Ressource zurück, die in der Vorlage identifiziert wurde, die dem angegebenen Stack durch die logische ID DBInstance „My“ zugeordnet ist.

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Einzelheiten zur API finden Sie unter DescribeStackResource AWS -Tools für PowerShellCmdlet-Referenz (V5).

Beschreiben Sie Stack-Ressourcen

CLI

AWS CLI

Um Informationen über eine Stack-Ressource zu erhalten

Im folgenden `describe-stack-resources` Beispiel werden Details zu den Ressourcen im angegebenen Stack angezeigt.

```
aws cloudformation describe-stack-resources \
    --stack-name my-stack
```

Ausgabe:

```
{
    "StackResources": [
        {
            "StackName": "my-stack",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "bucket",
            "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
            "ResourceType": "AWS::S3::Bucket",
            "Timestamp": "2019-10-02T04:34:11.345Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "StackName": "my-stack",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "function",
            "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
            "ResourceType": "AWS::Lambda::Function",
            "Timestamp": "2019-10-02T05:34:27.989Z",
            "ResourceStatus": "UPDATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "StackName": "my-stack",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "functionRole",
            "PhysicalResourceId": "my-functionRole-HIZXMPLEOM9E",
            "ResourceType": "AWS::IAM::Role",
            "Timestamp": "2019-10-02T04:34:06.350Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
```

```
                "StackResourceDriftStatus": "IN_SYNC"
            }
        }
    ]
}
```

- Einzelheiten zur API finden Sie DescribeStackResourcesunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt die AWS Ressourcenbeschreibungen für bis zu 100 Ressourcen zurück, die dem angegebenen Stack zugeordnet sind. Um Details zu allen Ressourcen zu erhalten, die einem Stack zugeordnet sind, verwenden Sie Get- CFNStackResourceSummary, das auch manuelles Paging der Ergebnisse unterstützt.

```
Get-CFNStackResourceList -StackName "myStack"
```

Beispiel 2: Gibt die Beschreibung der EC2 Amazon-Instance zurück, die in der Vorlage identifiziert wurde, die dem angegebenen Stack durch die logische ID „Ec2Instance“ zugeordnet ist.

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Beispiel 3: Gibt die Beschreibung von bis zu 100 Ressourcen zurück, die dem Stack zugeordnet sind, der eine EC2 Amazon-Instance enthält, die durch die Instance-ID „i-123456“ identifiziert wird. Um Details zu allen Ressourcen zu erhalten, die einem Stack zugeordnet sind, verwenden Sie Get- CFNStackResourceSummary, das auch manuelles Paging der Ergebnisse unterstützt.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Beispiel 4: Gibt die Beschreibung der EC2 Amazon-Instance zurück, die durch die logische ID „Ec2Instance“ in der Vorlage für einen Stack identifiziert wird. Der Stack wird anhand der physischen Ressourcen-ID einer darin enthaltenen Ressource identifiziert, in diesem Fall auch einer EC2 Amazon-Instance mit der Instance-ID „i-123456“. Abhängig vom Inhalt der Vorlage könnte auch eine andere physische Ressource verwendet werden, um den Stapel zu identifizieren, z. B. ein Amazon S3 S3-Bucket.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId
 "Ec2Instance"
```

- Einzelheiten zur API finden Sie unter DescribeStackResources AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt die AWS Ressourcenbeschreibungen für bis zu 100 Ressourcen zurück, die dem angegebenen Stack zugeordnet sind. Um Details zu allen Ressourcen zu erhalten, die einem Stack zugeordnet sind, verwenden Sie Get- CFNStackResourceSummary, das auch manuelles Paging der Ergebnisse unterstützt.

```
Get-CFNStackResourceList -StackName "myStack"
```

Beispiel 2: Gibt die Beschreibung der EC2 Amazon-Instance zurück, die in der Vorlage identifiziert wurde, die dem angegebenen Stack durch die logische ID „Ec2Instance“ zugeordnet ist.

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Beispiel 3: Gibt die Beschreibung von bis zu 100 Ressourcen zurück, die dem Stack zugeordnet sind, der eine EC2 Amazon-Instance enthält, die durch die Instance-ID „i-123456“ identifiziert wird. Um Details zu allen Ressourcen zu erhalten, die einem Stack zugeordnet sind, verwenden Sie Get- CFNStackResourceSummary, das auch manuelles Paging der Ergebnisse unterstützt.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Beispiel 4: Gibt die Beschreibung der EC2 Amazon-Instance zurück, die durch die logische ID „Ec2Instance“ in der Vorlage für einen Stack identifiziert wird. Der Stack wird anhand der physischen Ressourcen-ID einer darin enthaltenen Ressource identifiziert, in diesem Fall auch einer EC2 Amazon-Instance mit der Instance-ID „i-123456“. Abhängig vom Inhalt der Vorlage könnte auch eine andere physische Ressource verwendet werden, um den Stapel zu identifizieren, z. B. ein Amazon S3 S3-Bucket.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId
 "Ec2Instance"
```

- Einzelheiten zur API finden Sie unter DescribeStackResources AWS -Tools für PowerShellCmdlet-Referenz (V5).

Beschreiben Sie Stapel

CLI

AWS CLI

Um Stapel zu beschreiben AWS CloudFormation

Der folgende `describe-stacks` Befehl zeigt zusammenfassende Informationen für den `myteststack` Stack:

```
aws cloudformation describe-stacks --stack-name myteststack
```

Ausgabe:

```
{
    "Stacks":  [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if
 you create a stack from this template.",
            "Tags": [],
            "Outputs": [
                {
                    "Description": "Name of S3 bucket to hold website content",
                    "OutputKey": "BucketName",
                    "OutputValue": "myteststack-s3bucket-jssofi1zie2w"
                }
            ],
            "StackStatusReason": null,
            "CreationTime": "2013-08-23T01:02:15.422Z",
            "Capabilities": [],
            "StackName": "myteststack",
            "StackStatus": "CREATE_COMPLETE",
            "DisableRollback": false
        }
```

```
    ]
}
```

Weitere Informationen finden Sie unter Stacks im AWS CloudFormation Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeStacksin der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
)

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
 CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
 structured format.
func (actor CloudFormationActions) GetOutputs(ctx context.Context, stackName
 string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(ctx,
 &cloudformation.DescribeStacksInput{
  StackName: aws.String(stackName),
 })
 if err != nil || len(output.Stacks) == 0 {
```

```
  log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
 stackName, err)
 }
 stackOutputs := StackOutputs{}
 for _, out := range output.Stacks[0].Outputs {
  stackOutputs[*out.OutputKey] = *out.OutputValue
 }
 return stackOutputs
}
```

- Einzelheiten zur API finden Sie DescribeStacksin der AWS SDK für Go API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt eine Sammlung von Stack-Instanzen zurück, die alle Stacks des Benutzers beschreiben.

```
Get-CFNStack
```

Beispiel 2: Gibt eine Stack-Instanz zurück, die den angegebenen Stack beschreibt

```
Get-CFNStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter DescribeStacks AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt eine Sammlung von Stack-Instanzen zurück, die alle Stacks des Benutzers beschreiben.

```
Get-CFNStack
```

Beispiel 2: Gibt eine Stack-Instanz zurück, die den angegebenen Stack beschreibt

```
Get-CFNStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter DescribeStacks AWS -Tools für PowerShellCmdlet-Referenz (V5).

Standardmäßig gibt der `describe-stacks` Befehl Parameterwerte zurück. Um zu verhindern, dass sensible Parameterwerte wie Kennwörter zurückgegeben werden, sollten Sie `TRUE` in Ihre CloudFormation Vorlagen eine `NoEcho` Eigenschaft aufnehmen, die auf gesetzt ist.

> **Important**
>
> Durch die Verwendung des `NoEcho`-Attributs werden keine Informationen maskiert, die im Folgenden gespeichert sind:
>
> - Der `Metadata` Vorlagenbereich. CloudFormation transformiert, modifiziert oder redigiert keine Informationen, die Sie in den `Metadata` Abschnitt aufnehmen. Weitere Informationen finden Sie unter Metadaten.
> - Der `Outputs`-Vorlagenabschnitt. Weitere Informationen finden Sie unter Ausgaben:
> - Das `Metadata`-Attribut einer Ressourcendefinition. Weitere Informationen finden Sie unter `Metadata`-Attribut.
>
> Es wird dringend empfohlen, diese Mechanismen nicht zu verwenden, um vertrauliche Informationen wie Passwörter oder Geheimnisse einzugeben.

> **Important**
>
> Anstatt vertrauliche Informationen direkt in Ihre CloudFormation Vorlagen einzubetten, empfehlen wir Ihnen, dynamische Parameter in der Stack-Vorlage zu verwenden, um auf vertrauliche Informationen zu verweisen, die außerhalb von gespeichert und verwaltet werden CloudFormation, z. B. im AWS Systems Manager Parameterspeicher oder. AWS Secrets Manager
> Weitere Informationen finden Sie in den bewährten Methoden zu Keine Anmeldeinformationen in Ihre Vorlagen einbetten.

Holen Sie sich eine Vorlage

CLI

AWS CLI

Um den Vorlagentext für einen AWS CloudFormation Stapel anzuzeigen

Der folgende `get-template` Befehl zeigt die Vorlage für den `myteststack` Stack:

```
aws cloudformation get-template --stack-name myteststack
```

Ausgabe:

```
{
    "TemplateBody": {
        "AWSTemplateFormatVersion": "2010-09-09",
        "Outputs": {
            "BucketName": {
                "Description": "Name of S3 bucket to hold website content",
                "Value": {
                    "Ref": "S3Bucket"
                }
            }
        },
        "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if
 you create a stack from this template.",
        "Resources": {
            "S3Bucket": {
                "Type": "AWS::S3::Bucket",
                "Properties": {
                    "AccessControl": "PublicRead"
                }
            }
        }
    }
}
```

- Einzelheiten zur API finden Sie GetTemplatein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt die Vorlage zurück, die dem angegebenen Stack zugeordnet ist.

```
Get-CFNTemplate -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter GetTemplate AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt die Vorlage zurück, die dem angegebenen Stack zugeordnet ist.

```
Get-CFNTemplate -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter GetTemplate AWS -Tools für PowerShellCmdlet-Referenz (V5).

Stack-Ressourcen auflisten

CLI

AWS CLI

Um Ressourcen in einem Stapel aufzulisten

Der folgende Befehl zeigt die Liste der Ressourcen im angegebenen Stack an.

```
aws cloudformation list-stack-resources \
    --stack-name my-stack
```

Ausgabe:

```
{
    "StackResourceSummaries": [
        {
            "LogicalResourceId": "bucket",
            "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
            "ResourceType": "AWS::S3::Bucket",
```

```
            "LastUpdatedTimestamp": "2019-10-02T04:34:11.345Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "LogicalResourceId": "function",
            "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
            "ResourceType": "AWS::Lambda::Function",
            "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
            "ResourceStatus": "UPDATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "LogicalResourceId": "functionRole",
            "PhysicalResourceId": "my-functionRole-HIZXMPLEOM9E",
            "ResourceType": "AWS::IAM::Role",
            "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        }
    ]
}
```

- Einzelheiten zur API finden Sie ListStackResourcesin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt Beschreibungen aller Ressourcen zurück, die dem angegebenen Stack zugeordnet sind.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter ListStackResources AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt Beschreibungen aller Ressourcen zurück, die dem angegebenen Stack zugeordnet sind.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter ListStackResources AWS -Tools für PowerShellCmdlet-Referenz (V5).

Stapel auflisten

Verwenden Sie den list-stacksBefehl, um Stapel aufzulisten. Um nur Stapel mit den angegebenen Statuscodes aufzulisten, fügen Sie die `--stack-status-filter` Option hinzu. Sie können einen oder mehrere Stack-Statuscodes für die `--stack-status-filter` Option angeben. Weitere Informationen finden Sie unter Stack-Statuscodes.

CLI

AWS CLI

Um AWS CloudFormation Stapel aufzulisten

Der folgende `list-stacks` Befehl zeigt eine Zusammenfassung aller Stapel mit dem Status: `CREATE_COMPLETE`

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

Ausgabe:

```
[
    {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
        "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket:
 Sample template showing how to create a publicly accessible S3 bucket.
 **WARNING** This template creates an S3 bucket. You will be billed for the AWS
 resources used if you create a stack from this template.",
        "StackStatusReason": null,
        "CreationTime": "2013-08-26T03:27:10.190Z",
```

```
        "StackName": "myteststack",
        "StackStatus": "CREATE_COMPLETE"
    }
]
```

- Einzelheiten zur API finden Sie ListStacksin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt zusammenfassende Informationen für alle Stapel zurück.

```
Get-CFNStackSummary
```

Beispiel 2: Gibt zusammenfassende Informationen für alle Stapel zurück, die gerade erstellt werden.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Beispiel 3: Gibt zusammenfassende Informationen für alle Stapel zurück, die gerade erstellt oder aktualisiert werden.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",
 "UPDATE_IN_PROGRESS")
```

- Einzelheiten zur API finden Sie unter ListStacks AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt zusammenfassende Informationen für alle Stapel zurück.

```
Get-CFNStackSummary
```

Beispiel 2: Gibt zusammenfassende Informationen für alle Stapel zurück, die gerade erstellt werden.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Beispiel 3: Gibt zusammenfassende Informationen für alle Stapel zurück, die gerade erstellt oder aktualisiert werden.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",
 "UPDATE_IN_PROGRESS")
```

- Einzelheiten zur API finden Sie unter ListStacks AWS -Tools für PowerShellCmdlet-Referenz (V5).

Einen Stack aktualisieren

Verwenden Sie den Befehl update-stack, um einen Stack direkt zu aktualisieren. Sie geben den Stack und Parameterwerte und Funktionen an, die Sie aktualisieren möchten, sowie, falls Sie eine aktualisierte Vorlage verwenden möchten, den Namen der Vorlage. Weitere Informationen finden Sie unter Stacks direkt aktualisieren.

CLI

AWS CLI

Um Stacks zu aktualisieren AWS CloudFormation

Der folgende `update-stack` Befehl aktualisiert die Vorlage und die Eingabeparameter für den `mystack` Stack:

```
aws cloudformation update-stack --stack-name mystack --
template-url https://s3.amazonaws.com/sample/updated.template --
parameters ParameterKey=KeyPairName,ParameterValue=SampleKeyPair
 ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

Der folgende `update-stack` Befehl aktualisiert nur den `SubnetIDs` Parameterwert für den `mystack` Stack. Wenn Sie keinen Parameterwert angeben, wird der in der Vorlage angegebene Standardwert verwendet:

```
aws cloudformation update-stack --stack-name mystack --
template-url https://s3.amazonaws.com/sample/updated.template
 --parameters ParameterKey=KeyPairName,UsePreviousValue=true
 ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,UpdatedSampleSubnetID2
```

Mit dem folgenden `update-stack` Befehl werden dem Stack zwei Themen für `mystack` Stack-Benachrichtigungen hinzugefügt:

```
aws cloudformation update-stack --stack-name mystack --use-previous-template --
notification-arns "arn:aws:sns:use-east-1:123456789012:mytopic1" "arn:aws:sns:us-
east-1:123456789012:mytopic2"
```

Weitere Informationen finden Sie unter AWS CloudFormation Stack-Updates im AWS CloudFormation Benutzerhandbuch.

- Einzelheiten zur API finden Sie UpdateStackin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Aktualisiert den Stack 'MyStack' mit den angegebenen Vorlagen- und Anpassungsparametern. 'PK1' steht für den Namen eines in der Vorlage deklarierten Parameters und 'PV1' für seinen Wert. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden.

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Beispiel 2: Aktualisiert den Stack 'MyStack' mit den angegebenen Vorlagen- und Anpassungsparametern. 'PK1' und 'PK2' stehen für die Namen der in der Vorlage deklarierten Parameter, '' und PV1 'PV2' stehen für ihre angeforderten Werte. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden.

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
 @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Beispiel 3: Aktualisiert den Stack 'MyStack' mit den angegebenen Vorlagen- und Anpassungsparametern. 'PK1' steht für den Namen eines in der Vorlage deklarierten Parameters und 'PV2' für seinen Wert. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden.

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Beispiel 4: Aktualisiert den Stack 'MyStack' mit der angegebenen Vorlage, die von Amazon S3 abgerufen wurde, und den Anpassungsparametern. 'PK1' und 'PK2' stehen für die Namen der in der Vorlage deklarierten Parameter, '' und PV1 'PV2' stehen für ihre angeforderten Werte. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden.

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
 @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Beispiel 5: Aktualisiert den Stack 'MyStack', von dem in diesem Beispiel angenommen wird, dass er IAM-Ressourcen enthält, mit der angegebenen Vorlage, die von Amazon S3 abgerufen wurde, und den Anpassungsparametern. 'PK1' und 'PK2' stehen für die Namen der in der Vorlage deklarierten Parameter, '' und PV1 '' stehen für ihre angeforderten Werte. PV2 Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden. Bei Stacks, die IAM-Ressourcen enthalten, müssen Sie den Capability-Parameter „CAPABILITY_IAM“ angeben. Andernfalls schlägt das Update mit einem Fehler '' fehl. InsufficientCapabilities

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
 @{ ParameterKey="PK2"; ParameterValue="PV2" } ) `
                -Capabilities "CAPABILITY_IAM"
```

- Einzelheiten zur API finden Sie unter UpdateStackCmdlet-Referenz (V4).AWS -Tools für PowerShell

Tools für V5 PowerShell

Beispiel 1: Aktualisiert den Stack 'MyStack' mit den angegebenen Vorlagen- und Anpassungsparametern. 'PK1' steht für den Namen eines in der Vorlage deklarierten Parameters und 'PV1' für seinen Wert. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden.

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Beispiel 2: Aktualisiert den Stack 'MyStack' mit den angegebenen Vorlagen- und Anpassungsparametern. 'PK1' und 'PK2' stehen für die Namen der in der Vorlage deklarierten Parameter, '' und PV1 'PV2' stehen für ihre angeforderten Werte. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden.

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
 @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Beispiel 3: Aktualisiert den Stack 'MyStack' mit den angegebenen Vorlagen- und Anpassungsparametern. 'PK1' steht für den Namen eines in der Vorlage deklarierten Parameters und 'PV2' für seinen Wert. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden.

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Beispiel 4: Aktualisiert den Stack 'MyStack' mit der angegebenen Vorlage, die von Amazon S3 abgerufen wurde, und den Anpassungsparametern. 'PK1' und 'PK2' stehen für die Namen der in der Vorlage deklarierten Parameter, '' und PV1 'PV2' stehen für ihre angeforderten Werte. Die Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden.

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
 @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Beispiel 5: Aktualisiert den Stack 'MyStack', von dem in diesem Beispiel angenommen wird, dass er IAM-Ressourcen enthält, mit der angegebenen Vorlage, die von Amazon S3 abgerufen wurde, und den Anpassungsparametern. 'PK1' und 'PK2' stehen für die Namen der in der Vorlage deklarierten Parameter, '' und PV1 '' stehen für ihre angeforderten Werte. PV2 Die

Anpassungsparameter können auch mit 'Key' und 'Value' anstelle von '' und ParameterKey 'ParameterValue' angegeben werden. Bei Stacks, die IAM-Ressourcen enthalten, müssen Sie den Capability-Parameter „CAPABILITY_IAM“ angeben. Andernfalls schlägt das Update mit einem Fehler '' fehl. InsufficientCapabilities

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
 @{ ParameterKey="PK2"; ParameterValue="PV2" } ) `
                -Capabilities "CAPABILITY_IAM"
```

- Einzelheiten zur API finden Sie unter UpdateStackCmdlet-Referenz (V5).AWS -Tools für PowerShell

> **Note**
>
> Um alle Benachrichtigungen zu entfernen, geben Sie für die Option `[]` for an. `--notification-arns`

Bestätigen Sie Ihre Vorlage

Verwenden Sie den validate-templateBefehl, um Ihre Vorlagendatei auf Syntaxfehler zu überprüfen.

Prüft während der Validierung CloudFormation zunächst, ob es sich bei der Vorlage um ein gültiges JSON handelt. Wenn dies nicht der Fall ist, wird von CloudFormation geprüft, ob die Vorlage gültiger YAML-Code ist. Wenn beide Prüfungen fehlschlagen, wird ein Fehler bei der Vorlagenvalidierung CloudFormation zurückgegeben.

CLI

AWS CLI

Um eine AWS CloudFormation Vorlage zu validieren

Der folgende `validate-template` Befehl validiert die `sampletemplate.json` Vorlage:

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

Ausgabe:

```
{
    "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template
 showing how to create a publicly accessible S3 bucket. **WARNING** This template
 creates an S3 bucket. You will be billed for the AWS resources used if you
 create a stack from this template.",
    "Parameters": [],
    "Capabilities": []
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter Arbeiten mit AWS CloudFormation Vorlagen.

- Einzelheiten zur API finden Sie ValidateTemplatein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Validiert den angegebenen Vorlageninhalt. In der Ausgabe werden die Funktionen, die Beschreibung und die Parameter der Vorlage detailliert beschrieben.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Beispiel 2: Validiert die angegebene Vorlage, auf die über eine Amazon S3 S3-URL zugegriffen wurde. In der Ausgabe werden die Funktionen, die Beschreibung und die Parameter der Vorlage detailliert beschrieben.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template
```

- Einzelheiten zur API finden Sie unter ValidateTemplate AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Validiert den angegebenen Vorlageninhalt. In der Ausgabe werden die Funktionen, die Beschreibung und die Parameter der Vorlage detailliert beschrieben.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Beispiel 2: Validiert die angegebene Vorlage, auf die über eine Amazon S3 S3-URL zugegriffen wurde. In der Ausgabe werden die Funktionen, die Beschreibung und die Parameter der Vorlage detailliert beschrieben.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template
```

- Einzelheiten zur API finden Sie unter ValidateTemplate AWS -Tools für PowerShellCmdlet-Referenz (V5).

Im Folgenden finden Sie ein Beispiel für eine Antwort, die zu einem Validierungsfehler führt.

```
{
    "ResponseMetadata": {
        "RequestId": "4ae33ec0-1988-11e3-818b-e15a6df955cd"
    },
    "Errors": [
        {
            "Message": "Template format error: JSON not well-formed. (line 11, column
 8)",
            "Code": "ValidationError",
            "Type": "Sender"
        }
    ],
    "Capabilities": [],
    "Parameters": []
}
A client error (ValidationError) occurred: Template format error: JSON not well-formed.
 (line 11, column 8)
```

> **Note**
>
> Der `validate-template`-Befehl wurde entwickelt, um nur die Syntax Ihrer Vorlage zu überprüfen. Er stellt nicht sicher, dass die Eigenschaftswerte, die Sie für eine Ressource angegeben haben, für diese Ressource gültig sind. Er bestimmt auch nicht die Anzahl der Ressourcen, die bei der Erstellung des Stacks vorhanden sein werden.
> Um die Betriebsgültigkeit zu überprüfen, müssen Sie versuchen, den Stack zu erstellen. Es gibt weder eine Sandbox noch einen Testbereich für AWS CloudFormation Stacks, sodass Ihnen die Ressourcen, die Sie beim Testen erstellen, in Rechnung gestellt werden.

Laden Sie lokale Artefakte in einen S3-Bucket hoch mit dem AWS CLI

Sie können den verwenden AWS CLI , um lokale Artefakte, auf die eine CloudFormation Vorlage verweist, in einen Amazon S3 S3-Bucket hochzuladen. Lokale Artefakte sind Dateien, auf die Sie in Ihrer Vorlage verweisen. Anstatt Dateien manuell in einen S3-Bucket hochzuladen und dann ihre Speicherorte zu Ihrer Vorlage hinzuzufügen, können Sie lokale Artefakte in Ihrer Vorlage angeben und sie mit dem packageBefehl schnell hochladen.

Ein lokales Artefakt ist ein Pfad zu einer Datei oder einem Ordner, die bzw. der mit dem package-Befehl in Amazon S3 hochgeladen wird. Ein Artefakt kann beispielsweise ein lokaler Pfad zum Quellcode Ihrer AWS Lambda Funktion oder die OpenAPI-Datei einer Amazon API Gateway Gateway-REST-API sein.

Wenn Sie den package Befehl verwenden:

- Wenn Sie eine Datei angeben, lädt der Befehl sie direkt in den S3-Bucket hoch.
- Wenn Sie einen Ordner angeben, erstellt der Befehl eine `.zip` Datei für den Ordner und lädt die `.zip` Datei dann hoch.
- Wenn Sie keinen Pfad angeben, erstellt der Befehl eine `.zip` Datei für das Arbeitsverzeichnis und lädt sie hoch.

Sie können einen absoluten oder relativen Pfad angeben, wobei der relative Pfad relativ zum Speicherort Ihrer Vorlage ist.

Nach dem Hochladen der Artefakte gibt der Befehl eine Kopie Ihrer Vorlage zurück, wobei Verweise auf lokale Artefakte durch den S3-Speicherort ersetzt werden, in den der Befehl die Artefakte hochgeladen hat. Sie können dann die zurückgegebene Vorlage verwenden, um einen Stack zu erstellen oder zu aktualisieren.

> **Note**
>
> Sie können lokale Artefakte nur für Ressourceneigenschaften verwenden, die der package-Befehl unterstützt. Weitere Informationen zu diesem Befehl und eine Liste der unterstützten Ressourceneigenschaften finden Sie in der packageDokumentation in der AWS CLI Befehlsreferenz.

Voraussetzungen

Bevor Sie beginnen, müssen Sie über einen vorhandenen Amazon S3 S3-Bucket verfügen.

Eine Vorlage mit lokalen Artefakten verpacken und bereitstellen

Die folgende Vorlage gibt das lokale Artefakt für den Quellcode einer Lambda-Funktion an. Der Quellcode wird im `/home/user/code/lambdafunction` Ordner gespeichert.

Originalvorlage

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Transform": "AWS::Serverless-2016-10-31",
  "Resources": {
    "MyFunction": {
      "Type": "AWS::Serverless::Function",
      "Properties": {
        "Handler": "index.handler",
        "Runtime": "nodejs18.x",
        "CodeUri": "/home/user/code/lambdafunction"
      }
    }
  }
}
```

Der folgende packageBefehl erstellt eine `.zip` Datei mit dem Quellcode-Ordner der Funktion und lädt sie in das Stammverzeichnis des angegebenen Buckets hoch.

```
aws cloudformation package \
  --s3-bucket amzn-s3-demo-bucket \
  --template /path_to_template/template.json \
  --output-template-file packaged-template.json \
  --output json
```

Der Befehl generiert eine neue Vorlage in dem von `--output-template-file` angegebenen Pfad. Es ersetzt die Artefaktreferenz durch den Amazon S3 S3-Standort, wie unten gezeigt. Die `.zip` Datei wird anhand der MD5 Prüfsumme des Ordnerinhalts benannt und nicht anhand des Ordnernamens selbst.

Resultierende Vorlage

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Transform": "AWS::Serverless-2016-10-31",
  "Resources": {
    "MyFunction": {
      "Type": "AWS::Serverless::Function",
      "Properties": {
        "Handler": "index.handler",
        "Runtime": "nodejs18.x",
        "CodeUri": "s3://amzn-s3-demo-bucket/md5 checksum"
      }
    }
  }
}
```

Nachdem Sie die Artefakte Ihrer Vorlage gepackt haben, stellen Sie die verarbeitete Vorlage mithilfe des deployBefehls bereit.

```
aws cloudformation deploy \
  --template-file packaged-template.json \
  --stack-name stack-name
```

Wenn Sie Vorlagen bereitstellen, die größer als 51.200 Byte sind, verwenden Sie den deploy Befehl mit der `--s3-bucket` Option, Ihre Vorlage auf S3 hochzuladen, wie im folgenden Beispiel gezeigt.

```
aws cloudformation deploy \
  --template-file packaged-template.json \
  --stack-name stack-name \
  --s3-bucket amzn-s3-demo-bucket
```

> **Note**
>
> Ein weiteres Beispiel für die Verwendung des package Befehls zum Hochladen lokaler Artefakte finden Sie unter. Teilen Sie eine Vorlage mithilfe verschachtelter Stapel in wiederverwendbare Teile auf

Verwaltung von Stacks über Konten und Regionen hinweg mit StackSets

AWS CloudFormation StackSets erweitert die Funktionen von Stacks, indem es Ihnen ermöglicht, Stacks für mehrere Konten und AWS-Regionen mit einem einzigen Vorgang zu erstellen, zu aktualisieren oder zu löschen. Mithilfe eines Administratorkontos definieren und verwalten Sie eine CloudFormation Vorlage und verwenden die Vorlage als Grundlage für die Bereitstellung von Stacks für ausgewählte Zielkonten auf bestimmten Zielkonten. AWS-Regionen

Dieser Abschnitt hilft Ihnen bei den ersten Schritten StackSets und beantwortet häufig gestellte Fragen zur Arbeit mit der StackSet Erstellung, Aktualisierung und Löschung sowie zur Problembehebung.

Themen

- StackSets Konzepte
- Voraussetzungen für die Verwendung AWS CloudFormation StackSets
- Erste Schritte mit StackSets der Verwendung einer Beispielvorlage
- AWS CloudFormation StackSets Mit selbstverwalteten Berechtigungen erstellen
- AWS CloudFormation StackSets Mit vom Service verwalteten Berechtigungen erstellen
- Automatische Bereitstellungen für StackSets in aktivieren oder deaktivieren AWS Organizations
- Aktualisiere AWS CloudFormation StackSets
- Stapel hinzufügen zu CloudFormation StackSets
- Überschreiben Sie Parameterwerte auf Stacks in Ihrem CloudFormation StackSet
- Stapel löschen von AWS CloudFormation StackSets
- Löschen AWS CloudFormation StackSets
- Vermeiden Sie fehlgeschlagene StackSets Bereitstellungen mithilfe von Target Account Gates
- Wählen Sie den Parallelitätsmodus für AWS CloudFormation StackSets
- Drifterkennung durchführen am CloudFormation StackSets
- Stapel importieren in AWS CloudFormation StackSets
- Bewährte Methoden für die Verwendung AWS CloudFormation StackSets
- AWS CloudFormation StackSets Beispielvorlagen

- Problembehebung AWS CloudFormation StackSets

StackSets Konzepte

Wenn Sie verwenden StackSets, arbeiten Sie mit StackSetsStack-Instances und Stacks.

Themen

- Administrator- und Ziel-Konten
- AWS CloudFormation StackSets
- Berechtigungsmodelle für StackSets
- Stack-Instances
- StackSet Operationen
- StackSet Operationsoptionen
- Tags
- StackSets Statuscodes
- Stack-Instance-Statuscodes

Administrator- und Ziel-Konten

Ein Administratorkonto ist das, AWS-Konto in dem Sie erstellen StackSets. StackSets Bei vom Dienst verwalteten Berechtigungen ist das Administratorkonto entweder das Verwaltungskonto der Organisation oder ein delegiertes Administratorkonto. Sie können ein verwalten, StackSet indem Sie sich bei dem AWS Administratorkonto anmelden, mit dem das erstellt wurde. StackSet

Ein Zielkonto ist das Konto, für das Sie einen oder mehrere Stacks in Ihrem StackSet erstellen, aktualisieren oder löschen. Bevor Sie einen verwenden können StackSet , um Stacks in einem Zielkonto zu erstellen, müssen Sie eine Vertrauensbeziehung zwischen dem Administrator und den Zielkonten einrichten.

AWS CloudFormation StackSets

A StackSetdient als Container für mehrere Stacks, die in bestimmten AWS-Konten Regionen bereitgestellt werden. Jeder Stapel basiert auf derselben CloudFormation Vorlage, aber Sie können einzelne Stapel mithilfe von Parametern anpassen.

Nachdem Sie eine definiert haben StackSet, können Sie Stacks in den Zielkonten erstellen, aktualisieren oder löschen, und zwar nach Ihren AWS-Regionen Wünschen. Wenn Sie Stacks erstellen, aktualisieren und löschen, können Sie auch Betriebseinstellungen festlegen. Geben Sie beispielsweise die Reihenfolge der Regionen an, in denen Sie den Vorgang ausführen möchten, den Schwellenwert für die Ausfalltoleranz vor dem Stack-Vorgang und die Anzahl der Konten, die gleichzeitig Stack-Vorgänge ausführen.

A StackSet ist eine regionale Ressource. Wenn Sie eine StackSet in einer Region erstellen AWS-Region, können Sie sie nur sehen oder ändern, wenn Sie diese Region aufrufen.

Berechtigungsmodelle für StackSets

Sie können entweder selbstverwaltete Berechtigungen oder dienstverwaltete Berechtigungen StackSets verwenden.

Mit selbstverwalteten Berechtigungen erstellen Sie die IAM-Rollen, die für die StackSets konto- und regionsübergreifende Bereitstellung erforderlich sind. Diese Rollen sind erforderlich, um eine vertrauenswürdige Beziehung zwischen dem Konto, von dem StackSet aus Sie die Daten verwalten, und dem Konto, für das Sie Stack-Instances bereitstellen, herzustellen. Wenn Sie dieses Berechtigungsmodell verwenden, StackSets können Sie die Bereitstellung für alle Benutzer AWS-Konto durchführen, für die Sie die Berechtigung zum Erstellen einer IAM-Rolle haben.

Mit den Berechtigungen serviceverwaltet können Sie Stack-Instances auf Konten verteilen, die von AWS Organizations verwaltet werden. Mit diesem Berechtigungsmodell müssen Sie nicht die erforderlichen IAM-Rollen erstellen. Die IAM-Rollen werden in Ihrem Namen StackSets erstellt. Mit diesem Modell können Sie auch automatische Bereitstellungen für Konten aktivieren, die Ihrer Organisation in der Zukunft hinzugefügt werden.

AWS Organizations lässt sich in Ihre Umgebung integrieren CloudFormation und unterstützt Sie bei der zentralen Verwaltung und Steuerung Ihrer Umgebung, während Sie Ihre AWS Ressourcen skalieren und erweitern.

- Verwaltungskonto – das Konto, das Sie zur Erstellung der Organisation verwenden. Weitere Informationen finden Sie unter Terminologie und Konzepte für AWS Organizations.
- Delegierter Administrator — Ein kompatibler AWS Dienst kann ein AWS Mitgliedskonto in der Organisation als Administrator für die Konten der Organisation in diesem Dienst registrieren. Weitere Informationen finden Sie unter AWS Dienste, die Sie zusammen AWS Organizations verwenden können.

Weitere Informationen zum Erstellen und Verwalten StackSets mit vom Dienst verwalteten Berechtigungen finden Sie in den folgenden Themen:

- Aktivieren Sie den vertrauenswürdigen Zugriff für StackSets mit AWS Organizations
- Registrieren Sie ein delegiertes Administrator-Mitgliedskonto
- AWS CloudFormation StackSets Mit vom Service verwalteten Berechtigungen erstellen

Stack-Instances

Eine Stack-Instance ist eine Referenz auf einen Stack in einem Zielkonto innerhalb einer Region. Eine Stack-Instance kann ohne Stack existieren. Wenn der Stack beispielsweise aus irgendeinem Grund nicht erstellt werden konnte, zeigt die Stack-Instance den Grund für den Fehler bei der Stack-Erstellung an. Eine Stack-Instanz ist nur einer StackSet zugeordnet.

Die folgende Abbildung zeigt die logischen Beziehungen zwischen StackSets Stack-Operationen und Stacks. Wenn Sie a aktualisieren StackSet, werden alle zugehörigen Stack-Instances in allen Konten und Regionen aktualisiert.

StackSet Operationen

Sie können die folgenden Operationen an ausführen StackSets.

Erstellen StackSet

Das Erstellen eines neuen StackSet umfasst die Angabe einer CloudFormation Vorlage, die Sie zum Erstellen von Stacks verwenden möchten, die Angabe der Zielkonten, in denen Sie Stacks erstellen möchten, und die Identifizierung der Zielkonten, AWS-Regionen in denen Sie Stacks in Ihren Zielkonten bereitstellen möchten. A StackSet gewährleistet die konsistente Bereitstellung derselben Stack-Ressourcen mit denselben Einstellungen für alle angegebenen Zielkonten in den von Ihnen ausgewählten Regionen.

Aktualisieren StackSet

Wenn Sie eine aktualisieren StackSet, übertragen Sie die Änderungen auf die Stapel in Ihrem StackSet. Sie können a StackSet auf eine der folgenden Arten aktualisieren. Ihre Vorlagenaktualisierungen wirken sich immer auf alle Stapel aus. Sie können die Vorlage nicht selektiv für einige Stapel im aktualisieren StackSet, für andere jedoch nicht.

- Ändern Sie bestehende Einstellungen in der Vorlage oder fügen Sie neue Ressourcen hinzu, z. B. das Aktualisieren von Parametereinstellungen für einen bestimmten Service oder das Hinzufügen neuer EC2 Amazon-Instances.
- Ersetzen der Vorlage durch eine andere Vorlage.
- Hinzufügen von Stacks in bestehenden oder zusätzlichen Zielkonten, über bestehende oder zusätzliche Regionen hinweg.

Stacks löschen

Wenn Sie Stacks löschen, entfernen Sie einen Stack und alle zugehörigen Ressourcen aus den von Ihnen angegebenen Zielkonten innerhalb der von Ihnen angegebenen Regionen. Sie können Stacks wie folgt löschen.

- Stacks von einigen Zielkonten löschen, während andere Stacks in anderen Zielkonten weiterhin ausgeführt werden.
- Löschen von Stacks aus einigen Regionen, während Stacks in anderen Regionen weiterlaufen.
- Löschen Sie Stapel aus Ihren StackSet, speichern Sie sie jedoch, damit sie unabhängig von Ihren weiterhin ausgeführt werden können, StackSet indem Sie die Option Stacks beibehalten wählen. Sie können die gespeicherten Stapel dann außerhalb Ihres eigenen Speichers verwalten. StackSet CloudFormation
- Löschen Sie alle Stapel in Ihrem StackSet, um das Löschen Ihrer gesamten Stapel vorzubereiten. StackSet

Löschen StackSet

Sie können Ihre StackSet nur löschen, wenn keine Stack-Instances darin enthalten sind.

StackSet Operationsoptionen

Die in diesem Abschnitt beschriebenen Optionen helfen dabei, die Dauer und Anzahl der Fehler zu kontrollieren, die für erfolgreiche StackSet Operationen zulässig sind, und verhindern, dass Sie Stack-Ressourcen verlieren.

Maximale Anzahl paralleler Konten

Diese Einstellung steht in den Workflows für das Erstellen, Aktualisieren und Löschen zur Verfügung. Sie gestattet Ihnen, die maximale Anzahl oder den Prozentsatz der Zielkonten anzugeben, in dem ein Vorgang ausgeführt wird. Eine niedrigere Anzahl oder ein niedrigerer Prozentsatz bedeuten, dass ein Vorgang in weniger Zielkonten ausgeführt wird. Die Vorgänge

werden jeweils in einer Region in der Reihenfolge durchgeführt, die im Feld Deployment order (Bereitstellungsreihenfolge) angegeben ist. Wenn Sie z. B. Stacks auf 10 Zielkonten innerhalb von zwei Regionen bereitstellen und Maximum concurrent accounts (Maximale gleichzeitige Konten) auf 50 und By percentage (Prozentual) einstellen, werden Stacks auf fünf Konten in der ersten Region und dann auf die zweiten fünf Konten innerhalb der ersten Region bereitgestellt, bevor in der nächsten Region mit der Bereitstellung an die ersten fünf Zielkonten begonnen wird.

Wenn Sie „Nach Prozent“ wählen und der angegebene Prozentsatz nicht einer ganzen Anzahl Ihrer angegebenen Konten entspricht, wird CloudFormation abgerundet. Wenn Sie beispielsweise Stacks für 10 Zielkonten bereitstellen und die maximale Anzahl gleichzeitiger Konten auf 25 und nach Prozent festlegen, wird von der gleichzeitigen Bereitstellung von 2,5 Stacks (was nicht möglich wäre) auf die gleichzeitige Bereitstellung von zwei Stacks CloudFormation abgerundet.

Beachten Sie, dass Ihnen diese Einstellung ermöglicht, das Maximum für Operationen anzugeben. Bei großen Bereitstellungen kann unter Umständen die tatsächliche Anzahl von gleichzeitig bedienen Konten aufgrund einer Service-Ablehnung kleiner ist.

Die maximale Anzahl gleichzeitiger Konten kann je nach Parallelitätsmodus vom Wert der Fehlertoleranz abhängen. Wenn Ihr Parallelitätsmodus auf Strikte Fehlertoleranz eingestellt ist, kann die maximale Anzahl gleichzeitiger Konten höchstens um eins höher sein als die Einstellung Fehlertoleranz.

Parallelitätsmodus

Mit dieser Einstellung, die in Workflows zum Erstellen, Aktualisieren und Löschen verfügbar ist, können Sie auswählen, wie sich die Parallelitätsebene bei Vorgängen verhält. StackSet Weitere Informationen finden Sie unter Wählen Sie den Parallelitätsmodus für AWS CloudFormation StackSets.

Fehlertoleranz

Mit dieser Einstellung, die in Workflows zum Erstellen, Aktualisieren und Löschen verfügbar ist, können Sie pro Region die maximale Anzahl oder den Prozentsatz von Stack-Operationsfehlern angeben, die auftreten können. Bei Überschreitung dieser Einstellung wird ein Vorgang automatisch CloudFormation beendet. Eine niedrigere Anzahl bzw. ein geringerer Prozentsatz bedeutet, dass der Vorgang auf weniger Stacks ausgeführt wird, aber dass Sie schneller mit der Fehlersuche für fehlgeschlagene Vorgänge beginnen können. Wenn Sie z.B. 10 Stacks in 10 Zielkonten innerhalb von drei Regionen aktualisieren, bedeutet die Einstellung von Failure tolerance (Ausfalltoleranz) auf 20 und By percentage (Prozentual), dass maximal zwei Stack-Aktualisierungen in einer Region fehlschlagen können, damit der Vorgang fortgesetzt

werden kann. Wenn ein dritter Stack in derselben Region ausfällt, CloudFormation wird der Vorgang beendet. Wenn ein Stack in der ersten Region nicht aktualisiert werden konnte, wird der Aktualisierungsvorgang in dieser Region fortgesetzt und geht dann zur nächsten Region über. Wenn zwei Stapel in der zweiten Region nicht aktualisiert werden können, erreicht die Fehlertoleranz 20%. Wenn ein dritter Stack in der Region ausfällt, wird der Aktualisierungsvorgang CloudFormation gestoppt und es wird nicht zu nachfolgenden Regionen übergegangen.

Wenn Sie „Nach Prozent“ wählen und der angegebene Prozentsatz nicht einer ganzen Anzahl Ihrer Stapel in jeder Region entspricht, wird CloudFormation abgerundet. Wenn Sie beispielsweise Stacks für 10 Zielkonten in drei Regionen bereitstellen und die Fehlertoleranz auf 25 und „Nach Prozent“ festlegen, wird von einer Fehlertoleranz von 2,5 Stacks (was nicht möglich wäre) auf eine Fehlertoleranz von zwei Stacks pro Region CloudFormation abgerundet.

Stacks beibehalten

Mit dieser Einstellung, die in Workflows zum Löschen von Stacks verfügbar ist, können Sie Stapel und ihre Ressourcen auch nach dem Entfernen von Stacks aus einem am Laufen halten. StackSet Wenn Sie Stacks beibehalten, lässt AWS CloudFormation die Stacks in einzelnen Konten und Regionen intakt. Stacks trennen sich von dem StackSet, schonen aber den Stapel und seine Ressourcen. Nachdem ein Vorgang zum Löschen von Stacks abgeschlossen ist, verwalten Sie die gespeicherten Stacks in dem Zielkonto (nicht im Administratorkonto) CloudFormation, das die Stacks erstellt hat.

Region-Parallelität

Mit dieser Einstellung, die in Workflows zum Erstellen, Aktualisieren und Löschen verfügbar ist, können Sie auswählen, wie die StackSets Bereitstellung in Regionen erfolgen soll.

Sequentiell — Bereitstellen StackSets des Vorgangs in jeweils einer Region, wie im Feld „Region Deployment Order“ angegeben, sofern die Bereitstellungsfehler einer Region eine angegebene Fehlertoleranz nicht überschreiten. Die sequenzielle Bereitstellung ist die Standardauswahl.

Parallel — Bereitstellen von StackSets Vorgängen in allen angegebenen Regionen gleichzeitig, sofern die Bereitstellungsfehler einer Region eine angegebene Fehlertoleranz nicht überschreiten.

Tags

Sie können bei StackSet Erstellungs- und Aktualisierungsvorgängen Tags hinzufügen, indem Sie Schlüssel- und Wertepaare angeben. Tags sind nützlich, um StackSet Ressourcen für die Abrechnung und Kostenzuweisung zu sortieren und zu filtern. Weitere Informationen zur Verwendung

von Stichwörtern finden Sie im AWSAWS Fakturierung und Kostenmanagement Benutzerhandbuch unter Organisieren und Nachverfolgen von AWS Kosten mithilfe von Kostenverrechnungs-Tags. Nachdem Sie das Schlüssel-Wert-Paar festgelegt haben, wählen Sie +, um das Tag zu speichern. Sie können Tags löschen, die Sie nicht mehr verwenden, indem Sie das rote X rechts von einem Tag auswählen.

Tags, die Sie anwenden, um sie StackSets auf alle Stapel und auf die Ressourcen anzuwenden, die durch Ihre Stapel erstellt werden. Sie können Tags auch nur auf Stack-Ebene in hinzufügen CloudFormation, aber diese Tags werden möglicherweise nicht angezeigt. StackSets

Es werden zwar StackSets keine systemdefinierten Tags hinzugefügt, Sie sollten jedoch die Schlüsselnamen von Tags nicht mit der Zeichenfolge beginnen. `aws:`

StackSets Statuscodes

AWS CloudFormation StackSets generiert Statuscodes für StackSet Operationen.

In der folgenden Tabelle werden Statuscodes für StackSet Operationen beschrieben.

`RUNNING`

Die Operation ist derzeit in Bearbeitung.

`SUCCEEDED`

Die Operation wurde abgeschlossen, ohne die Fehlertoleranz für die Operation zu überschreiten.

`FAILED`

Die Anzahl der Stacks, auf denen der Vorgang nicht fertiggestellt werden konnte, hat die benutzerdefinierte Fehlertoleranz überschritten. Der Fehlertoleranz-Wert, den Sie für einen Vorgang festgelegt haben, gilt während Stack-Erstellungs- und Aktualisierungs-Vorgängen für jede Region. Wenn die Anzahl der fehlgeschlagenen Stacks innerhalb einer Region die Fehlertoleranz überschreitet, wird der Status des Vorgangs in der Region auf `FAILED` gesetzt. Der Status des gesamten Vorgangs ist ebenfalls auf `FAILED` festgelegt und der Vorgang wird in allen verbleibenden Regionen CloudFormation abgebrochen.

`QUEUED`

[`Service-managed permissions`] Bei automatischen Bereitstellungen, die eine Abfolge von Vorgängen erfordern, wird der Vorgang in die Warteschlange gestellt, damit er ausgeführt werden kann. Zum Beispiel:

- Wenn ein Konto von einer Organisationseinheit (OU) in eine andere verschoben wird0U2, wird eine automatische Bereitstellung ausgelöst. 0U1 StackSets führt einen Löschvorgang aus, um die Stack-Instance aus dem 0U1 Zielkonto in der Zielregion zu entfernen, und stellt einen Erstellungsvorgang in die Warteschlange, um dem 0U2 Zielkonto in der Zielregion eine Stack-Instance hinzuzufügen.
- Das Hinzufügen eines Kontos `AccountA` zu einer Organisationseinheit löst eine automatische Bereitstellung aus. StackSets führt einen Erstellungsvorgang aus, um eine Stack-Instance `AccountA` in der Zielregion hinzuzufügen. Wenn Sie der Organisationseinheit ein weiteres Konto `AccountB` hinzufügen, während dieser Erstellungsvorgang ausgeführt wird, wird ein zweiter Erstellungsvorgang in die StackSets Warteschlange gestellt. Wenn der erste Erstellungsvorgang abgeschlossen ist, wird der zweite Erstellungsvorgang StackSets ausgeführt, um der Zielregion eine Stack-Instanz hinzuzufügen. `AccountB`

`STOPPING`

Die Operation wird gerade auf Anforderung des Benutzers angehalten.

`STOPPED`

Der Vorgang wurde auf Anforderung des Benutzers angehalten.

Stack-Instance-Statuscodes

AWS CloudFormation StackSets generiert Statuscodes für Stack-Instances.

In der folgenden Tabelle werden die Statuscodes für die darin enthaltenen Stack-Instances beschrieben StackSets.

`CURRENT`

Der Stack ist auf dem neuesten Stand mit dem StackSet.

`OUTDATED`

Der Stapel ist aus einem der StackSet folgenden Gründe nicht auf dem neuesten Stand.

- Eine CreateStackSet- oder UpdateStackSet-Operation auf dem zugehörigen Stack ist fehlgeschlagen.
- Der Stack war Teil eines CreateStackSet- oder UpdateStackSet-Vorgangs, der fehlgeschlagen ist oder gestoppt wurde, bevor der Stack erstellt oder aktualisiert werden konnte.

`INOPERABLE`

Eine DeleteStackInstances-Operation ist fehlgeschlagen und der Stack ist in einem instabilen Status. Stacks in diesem Status werden von weiteren UpdateStackSet-Operationen ausgeschlossen. Möglicherweise benötigen Sie eine DeleteStackInstances-Operation, mit `RetainStacks` gleich `true`, um die Stack-Instance zu löschen und anschließend den Stack manuell zu löschen.

`CANCELLED`

Der Vorgang im angegebenen Konto und der angegebenen Region wurde storniert. Das passiert, weil ein Benutzer den StackSet Vorgang beendet hat oder weil die StackSet Operationen die Fehlertoleranz überschreiten.

`FAILED`

Der Vorgang im angegebenen Konto und der angegebenen Region ist fehlgeschlagen. Wenn der StackSet Vorgang bei genügend Konten innerhalb einer Region fehlschlägt, wird möglicherweise die Fehlertoleranz für den gesamten StackSet Vorgang überschritten.

`FAILED_IMPORT`

Der Import der Stack-Instance im angegebenen Konto und in der angegebenen Region ist fehlgeschlagen und der Stack befindet sich in einem instabilen Zustand. Sobald die Probleme, die den Fehler verursacht haben, behoben sind, kann der Importvorgang erneut versucht werden. Wenn genügend StackSet Operationen bei genügend Konten innerhalb einer Region fehlschlagen, kann die Fehlertoleranz für den gesamten StackSet Vorgang überschritten werden.

`PENDING`

Der Vorgang im angegebenen Konto und der angegebenen Region muss noch gestartet werden.

`RUNNING`

Der Vorgang im angegebenen Konto und der angegebenen Region wird derzeit ausgeführt.

`SKIPPED_SUSPENDED_ACCOUNT`

Der Vorgang im angegebenen Konto und der angegebenen Region wurde übersprungen, da das Konto zum Zeitpunkt des Vorgangs gesperrt war.

`SUCCEEDED`

Der Vorgang im angegebenen Konto und der angegebenen Region wurde erfolgreich abgeschlossen.

Voraussetzungen für die Verwendung AWS CloudFormation StackSets

StackSets erweitern Sie die Funktionalität von Stacks, sodass Sie Stacks für mehrere Konten und Regionen mit einem einzigen Vorgang erstellen, aktualisieren oder löschen können.

Da StackSets Sie Stack-Operationen über mehrere Konten hinweg durchführen können, benötigen Sie, bevor Sie Ihr erstes StackSet Konto erstellen können, die erforderlichen Berechtigungen, die in Ihrem definiert sind. AWS-Konten

Sie können die Verwaltung StackSets mit selbstverwalteten oder dienstverwalteten Berechtigungen durchführen.

- Für die Selbstverwaltung StackSets müssen Sie IAM-Rollen in jedem Zielkonto und erstellen und verwalten. AWS-Region Weitere Informationen finden Sie unter Selbstverwaltete Berechtigungen erteilen.
- Bei Service Managed StackSets müssen Sie die IAM-Rollen nicht in jedem Konto manuell erstellen und verwalten. Die Rollenerstellung und die AWS Berechtigungen werden für Sie übernommen. Weitere Informationen finden Sie unter Aktivieren Sie den vertrauenswürdigen Zugriff.

> **Note**
>
> Die Aktivierung des vertrauenswürdigen Zugriffs mit AWS Organizations for wird derzeit in den Regionen China, Peking und Ningxia AWS CloudFormation StackSets nicht unterstützt.

Themen

- Bereiten Sie sich auf die Ausführung von StackSet Vorgängen vor AWS-Regionen , die standardmäßig deaktiviert sind
- Selbstverwaltete Berechtigungen erteilen
- Aktivieren Sie den vertrauenswürdigen Zugriff für StackSets mit AWS Organizations

Bereiten Sie sich auf die Ausführung von StackSet Vorgängen vor AWS-Regionen , die standardmäßig deaktiviert sind

AWS-Regionen nach dem 20. März 2019 eingeführte Produkte wie Asien-Pazifik (Hongkong) sind standardmäßig deaktiviert. Sie müssen diese Regionen für Ihr Konto/Ihre Konten aktivieren, bevor Sie sie verwenden können. Weitere Informationen finden Sie im AWS -Kontenverwaltung Referenzhandbuch unter AWS-Regionen In Ihrem Konto aktivieren oder deaktivieren.

Um ein Administratorkonto (wenn Sie selbstverwaltete Berechtigungen verwenden) oder das Verwaltungskonto einer StackSet Organisation (wenn Sie StackSet vom Dienst verwaltete Berechtigungen verwenden) in einer Region zu erstellen, die standardmäßig deaktiviert ist, müssen Sie zuerst diese Region für das Administrator- oder Verwaltungskonto aktivieren.

AWS CloudFormation Um eine Stack-Instanz erfolgreich zu erstellen oder zu aktualisieren:

- Das Zielkonto muss sich in einer Region befinden, die derzeit für dieses Zielkonto aktiviert ist.
- Für StackSet das Administratorkonto oder das Verwaltungskonto der Organisation muss dieselbe Region aktiviert sein wie für das Zielkonto.

> **Important**
>
> Beachten Sie, dass Administrator- und Zielkonten während des StackSet Betriebs Metadaten zu den Konten selbst StackSet sowie zu den beteiligten StackSet Instanzen austauschen. Wenn Sie eine Region deaktivieren, die ein Konto enthält, in dem sich StackSet Instanzen befinden, sind Sie außerdem dafür verantwortlich, diese Instanzen oder Ressourcen auf Wunsch zu löschen. Beachten Sie auch, dass die Metadaten zum Zielkonto in der deaktivierten Region im Administratorkonto aufbewahrt werden.

Selbstverwaltete Berechtigungen erteilen

Dieses Thema enthält Anweisungen zum Erstellen der IAM-Dienstrollen, die für die kontenübergreifende Bereitstellung und AWS-Regionen mit StackSets selbstverwalteten Berechtigungen erforderlich sind. Diese Rollen sind erforderlich, um eine vertrauenswürdige Beziehung zwischen dem Konto, von dem StackSet aus Sie die Daten verwalten, und dem Konto, für das Sie Stack-Instances bereitstellen, herzustellen. Wenn Sie dieses Berechtigungsmodell

verwenden, StackSets können Sie die Bereitstellung für alle Benutzer AWS-Konto durchführen, für die Sie die Berechtigung zum Erstellen einer IAM-Rolle haben.

Informationen zur Verwendung von vom Dienst verwalteten Berechtigungen finden Sie stattdessen unterAktivieren Sie den vertrauenswürdigen Zugriff.

Themen

- Übersicht über selbstverwaltete Berechtigungen
- Erteilen Sie allen Benutzern des Administratorkontos die Berechtigung, Stacks in allen Zielkonten zu verwalten
- Richten Sie erweiterte Berechtigungsoptionen für StackSet Operationen ein
- Richten Sie globale Schlüssel ein, um Confused-Deputy-Probleme zu mindern.

Übersicht über selbstverwaltete Berechtigungen

Bevor Sie eine StackSet mit selbstverwalteten Berechtigungen erstellen, müssen Sie in jedem Konto IAM-Dienstrollen erstellt haben.

Die grundlegenden Schritte sind:

1. Ermitteln Sie, welches AWS-Konto das Administratorkonto ist.

   StackSets werden in diesem Administratorkonto erstellt. Ein Zielkonto ist das Konto, in dem Sie einzelne Stacks erstellen, die zu einem StackSet gehören.
2. Bestimmen Sie, wie Sie die Berechtigungen für die StackSet strukturieren möchten.

   Bei der einfachsten (und großzügigsten) Berechtigungskonfiguration geben Sie allen Benutzern und Gruppen im Administratorkonto die Möglichkeit, alle über dieses Konto StackSets verwalteten Berechtigungen zu erstellen und zu aktualisieren. Wenn Sie eine feinere Kontrolle benötigen, können Sie Berechtigungen einrichten, die Folgendes angeben:
   - Welche Benutzer und Gruppen können StackSet Operationen in welchen Zielkonten ausführen?
   - Welche Ressourcen Benutzer und Gruppen in ihre aufnehmen können StackSets.
   - Welche StackSet Operationen bestimmte Benutzer und Gruppen ausführen können.
3. Erstellen Sie die erforderlichen IAM-Servicerollen in Ihrem Administrator- und Ziel-Konto, um die gewünschten Berechtigungen zu definieren.

   Insbesondere sind die beiden Rollen erforderlich:

- AWSCloudFormationStackSetAdministrationRole— Diese Rolle wird für das Administratorkonto bereitgestellt.
- AWSCloudFormationStackSetExecutionRole— Diese Rolle wird für alle Konten bereitgestellt, in denen Sie Stack-Instances erstellen.

Erteilen Sie allen Benutzern des Administratorkontos die Berechtigung, Stacks in allen Zielkonten zu verwalten

In diesem Abschnitt erfahren Sie, wie Sie Berechtigungen einrichten, damit alle Benutzer und Gruppen des Administratorkontos StackSet Operationen in allen Zielkonten ausführen können. Er führt Sie durch die Erstellung der erforderlichen IAM-Servicerollen in Ihren Administrator- und Zielkonten. Jeder Benutzer des Administratorkontos kann dann alle Stacks in allen Zielkonten erstellen, aktualisieren oder löschen.

Durch diese Strukturierung von Berechtigungen übergeben Benutzer keine Administratorrolle, wenn sie eine erstellen oder aktualisieren. StackSet

Administrator account

Erstellen Sie im Administratorkonto eine IAM-Rolle mit dem Namen AWSCloudFormationStackSetAdministrationRole.

Sie können dies tun, indem Sie einen Stack aus der CloudFormation Vorlage erstellen, die unter https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetAdministrationRole.yml verfügbar ist.

Example Beispiel für eine Berechtigungsrichtlinie

Die mit der vorherigen Vorlage erstellte Administratorrolle umfasst die folgende Berechtigungsrichtlinie.

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

```
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example Beispiel für eine Vertrauensrichtlinie 1

Die vorherige Vorlage enthält auch die folgende Vertrauensrichtlinie, die dem Dienst die Berechtigung erteilt, die Administratorrolle und die mit der Rolle verknüpften Berechtigungen zu verwenden.

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

Example Beispiel für eine Vertrauensrichtlinie 2

Um Stack-Instances in einem Zielkonto bereitzustellen, das sich in einer Region befindet, die standardmäßig deaktiviert ist, müssen Sie auch den regionalen Dienstprinzipal für diese Region angeben. Jede standardmäßig deaktivierte Region verfügt über einen eigenen regionalen Service-Prinzipal.

Die folgende Beispiel-Vertrauensrichtlinie erteilt dem Dienst die Erlaubnis, die Administratorrolle in der Region Asien-Pazifik (Hongkong) (`ap-east-1`) zu verwenden, einer Region, die standardmäßig deaktiviert ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
            "cloudformation.amazonaws.com",
            "cloudformation.ap-east-1.amazonaws.com"
         ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Weitere Informationen finden Sie unter Bereiten Sie sich auf die Ausführung von StackSet Vorgängen vor AWS-Regionen , die standardmäßig deaktiviert sind. Eine Liste der Regionscodes finden Sie im Allgemeine AWS-Referenz Handbuch unter Regionale Endpunkte.

Target accounts

Erstellen Sie in jedem Zielkonto eine Servicerolle mit dem Namen AWSCloudFormationStackSetExecutionRole, der dem Administratorkonto vertraut. Die Rolle muss genau diesen Namen haben. Sie können dies tun, indem Sie einen Stack aus der CloudFormation Vorlage erstellen, die unter https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetExecutionRole.yml verfügbar ist. Wenn Sie diese Vorlage verwenden, werden Sie aufgefordert, die Konto-ID des Administratorkontos anzugeben, zu dem Ihr Zielkonto eine Vertrauensbeziehung haben muss.

> **Important**
>
> Beachten Sie, dass diese Vorlage Administratorzugriff gewährt. Nachdem Sie die Vorlage verwendet haben, um eine Ausführungsrolle für ein Zielkonto zu erstellen, müssen Sie die Berechtigungen in der Richtlinienanweisung auf die Ressourcentypen beschränken, die Sie verwenden StackSets.

Für die Servicerolle des Zielkontos sind Berechtigungen erforderlich, um alle in Ihrer CloudFormation Vorlage angegebenen Vorgänge ausführen zu können. Wenn Ihre Vorlage beispielsweise einen S3-Bucket erstellt, benötigen Sie Berechtigungen zum Erstellen neuer Objekte für S3. Ihr Zielkonto benötigt immer vollständige CloudFormation -Berechtigungen, darunter die Berechtigungen zum Erstellen, Aktualisieren, Löschen und Beschreiben von Stacks.

Example Beispiel für eine Berechtigungsrichtlinie 1

Die mit dieser Vorlage erstellte Rolle aktiviert die folgende Richtlinie in einem Zielkonto.

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

Example Beispiel für eine Berechtigungsrichtlinie 2

Das folgende Beispiel zeigt eine Richtlinienanweisung mit den Mindestberechtigungen für StackSets die Arbeit. Um Stapel in Zielkonten zu erstellen, die Ressourcen von anderen Diensten als verwenden CloudFormation, müssen Sie diese Dienstaktionen und Ressourcen zur AWSCloudFormationStackSetExecutionRoleRichtlinienerklärung für jedes Zielkonto hinzufügen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
```

```
}
```

Example Beispiel für eine Vertrauensrichtlinie

Die folgende Vertrauensbeziehung wird durch die Vorlage erstellt. Die ID des Administratorkontos wird als angezeigt*admin_account_id*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Sie können die Vertrauensbeziehung einer vorhandenen Ausführungsrolle für ein Zielkonto so konfigurieren, dass sie einer bestimmten Rolle im Administratorkonto vertraut. Wenn Sie die Rolle im Administratorkonto löschen und eine neue erstellen, um sie zu ersetzen, müssen Sie die Vertrauensstellung Ihres Zielkontos mit der neuen Administratorkontorolle konfigurieren, die *admin_account_id* im vorherigen Beispiel durch dargestellt wird.

Richten Sie erweiterte Berechtigungsoptionen für StackSet Operationen ein

Wenn Sie eine genauere Kontrolle darüber benötigen StackSets , was Benutzer und Gruppen über ein einziges Administratorkonto erstellen, können Sie mithilfe von IAM-Rollen Folgendes angeben:

- Welche Benutzer und Gruppen StackSet Operationen in welchen Zielkonten ausführen können.
- Welche Ressourcen Benutzer und Gruppen in ihre aufnehmen können StackSets.
- Welche StackSet Operationen bestimmte Benutzer und Gruppen ausführen können.

Steuern Sie, welche Benutzer StackSet Operationen in bestimmten Zielkonten ausführen können

Verwenden Sie benutzerdefinierte Administratorrollen, um zu steuern, welche Benutzer und Gruppen StackSet Operationen in welchen Zielkonten ausführen können. Möglicherweise möchten Sie steuern, welche Benutzer des Administratorkontos StackSet Operationen in welchen Zielkonten ausführen können. Zu diesem Zweck erstellen Sie eine Vertrauensbeziehung zwischen jedem Zielkonto und einer bestimmten benutzerdefinierten Administratorrolle, anstatt die AWSCloudFormationStackSetAdministrationRoleServicerolle im Administratorkonto selbst zu erstellen. Anschließend aktivieren Sie bestimmte Benutzer und Gruppen, um die benutzerdefinierte Administratorrolle bei der Ausführung von StackSet Vorgängen in einem bestimmten Zielkonto zu verwenden.

Sie können beispielsweise in Ihrem Administratorkonto die Rollen A und B erstellen. Rolle A erhält die Berechtigung zum Zugriff auf Zielkonto 1 über Konto 8. Rolle B erhält die Berechtigung zum Zugriff auf Zielkonto 9 über Konto 16.

Das Einrichten der erforderlichen Berechtigungen umfasst das Definieren einer benutzerdefinierten Administratorrolle, das Erstellen einer Servicerolle für das Zielkonto und das Erteilen der Berechtigungen für Benutzer, die benutzerdefinierte Administratorrolle bei der Ausführung von StackSet Vorgängen zu übergeben.

Im Allgemeinen funktioniert das wie folgt, sobald Sie über die erforderlichen Berechtigungen verfügen: Beim Erstellen einer StackSet muss der Benutzer eine benutzerdefinierte Administratorrolle angeben. Der Benutzer benötigt die Berechtigung zum Übergeben der Rolle an CloudFormation. Darüber hinaus muss die benutzerdefinierte Administratorrolle über eine Vertrauensbeziehung zu den Zielkonten verfügen, die für die angegeben sind StackSet. CloudFormation erstellt die benutzerdefinierte Verwaltungsrolle StackSet und ordnet ihr die benutzerdefinierte Administratorrolle zu. Beim Aktualisieren einer StackSet muss der Benutzer explizit eine benutzerdefinierte Administratorrolle angeben, auch wenn es sich um dieselbe benutzerdefinierte Administratorrolle handelt, die StackSet zuvor für diese Rolle verwendet wurde. CloudFormationverwendet diese Rolle, um den Stack zu aktualisieren, sofern die oben genannten Anforderungen erfüllt sind.

Administrator account

Example Beispiel für eine Berechtigungsrichtlinie

Erstellen Sie für jede StackSet Rolle eine benutzerdefinierte Administratorrolle mit der Berechtigung, die Ausführungsrolle des Zielkontos zu übernehmen.

Der Name der Ausführungsrolle für das Zielkonto muss in jedem Zielkonto identisch sein. Wenn der Rollenname lautet AWSCloudFormationStackSetExecutionRole, wird er automatisch beim Erstellen eines StackSets verwendet StackSet. Wenn Sie einen benutzerdefinierten Rollennamen angeben, müssen Benutzer den Namen der Ausführungsrolle beim Erstellen einer angeben StackSet.

Erstellen Sie eine IAM-Dienstrolle mit einem benutzerdefinierten Namen und der folgenden Berechtigungsrichtlinie. In den folgenden Beispielen *custom_execution_role* bezieht sich dies auf die Ausführungsrolle in den Zielkonten.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/custom_execution_role"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Um mehrere Konten in einem einzigen Kontoauszug anzugeben, trennen Sie sie durch Kommas.

```
"Resource": [
  "arn:aws:iam::target_account_id_1:role/custom_execution_role",
  "arn:aws:iam::target_account_id_2:role/custom_execution_role"
]
```

Sie können alle Zielkonten angeben, indem Sie anstelle einer Konto-ID einen Platzhalter (*) verwenden.

```
"Resource": [
  "arn:aws:iam::*:role/custom_execution_role"
]
```

Example Beispiel für Vertrauensrichtlinie 1

Sie müssen eine Vertrauensrichtlinie für die Servicerolle angeben, um zu definieren, welche IAM-Prinzipale die Rolle übernehmen können.

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

Example Beispiel für eine Vertrauensrichtlinie 2

Um Stack-Instances in einem Zielkonto bereitzustellen, das sich in einer Region befindet, die standardmäßig deaktiviert ist, müssen Sie auch den regionalen Dienstprinzipal für diese Region angeben. Jede standardmäßig deaktivierte Region verfügt über einen eigenen regionalen Service-Prinzipal.

Die folgende Beispiel-Vertrauensrichtlinie erteilt dem Dienst die Erlaubnis, die Administratorrolle in der Region Asien-Pazifik (Hongkong) (`ap-east-1`) zu verwenden, einer Region, die standardmäßig deaktiviert ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
            "cloudformation.amazonaws.com",
            "cloudformation.ap-east-1.amazonaws.com"
         ]
```

```
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Weitere Informationen finden Sie unter Bereiten Sie sich auf die Ausführung von StackSet Vorgängen vor AWS-Regionen , die standardmäßig deaktiviert sind. Eine Liste der Regionscodes finden Sie im AWS Allgemeinen Referenzhandbuch unter Regionale Endpunkte.

Example Beispiel für eine Pass-Rollenrichtlinie

Sie benötigen außerdem eine IAM-Berechtigungsrichtlinie für Ihre IAM-Benutzer, die es dem Benutzer ermöglicht, bei der Ausführung StackSet von Vorgängen die benutzerdefinierte Administratorrolle zu übergeben. Weitere Informationen finden Sie unter Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle an einen AWS -Service.

Im folgenden Beispiel *customized_admin_role* bezieht sich dies auf die Administratorrolle, die der Benutzer bestehen muss.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/customized_admin_role"
    }
  ]
}
```

Target accounts

Erstellen Sie in jedem Zielkonto eine Servicerolle, die der benutzerdefinierten Administratorrolle vertraut, die Sie mit diesem Konto verwenden möchten.

Für die Zielkontorolle sind Berechtigungen erforderlich, um alle in Ihrer CloudFormation Vorlage angegebenen Vorgänge ausführen zu können. Wenn Ihre Vorlage beispielsweise einen S3-Bucket erstellt, benötigen Sie Berechtigungen zum Erstellen neuer Objekte in S3. Ihr Zielkonto benötigt immer volle CloudFormation Berechtigungen, zu denen auch Berechtigungen zum Erstellen, Aktualisieren, Löschen und Beschreiben von Stacks gehören.

Der Rollenname des Zielkontos muss in jedem Zielkonto identisch sein. Wenn der Rollenname lautet AWSCloudFormationStackSetExecutionRole, wird er automatisch beim Erstellen eines StackSets verwendet StackSet. Wenn Sie einen benutzerdefinierten Rollennamen angeben, müssen Benutzer den Namen der Ausführungsrolle beim Erstellen einer angeben StackSet.

Example Beispiel für eine Berechtigungsrichtlinie

Das folgende Beispiel zeigt eine Richtlinienanweisung mit den Mindestberechtigungen für StackSets die Arbeit. Um Stacks in Zielkonten zu erstellen, die Ressourcen von anderen Diensten als verwenden CloudFormation, müssen Sie diese Dienstaktionen und Ressourcen zur Berechtigungsrichtlinie hinzufügen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Example Beispiel für eine Vertrauensrichtlinie

Sie müssen die folgende Vertrauensrichtlinie angeben, wenn Sie die Rolle zur Definition der Vertrauensbeziehung erstellen.

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

```
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Steuern Sie die Ressourcen, die Benutzer in bestimmten Fällen einbeziehen können StackSets

Verwenden Sie benutzerdefinierte Ausführungsrollen, um zu steuern, welche Stack-Ressourcen Benutzer und Gruppen in ihre Rollen aufnehmen können StackSets. Beispielsweise möchten Sie möglicherweise eine Gruppe einrichten, die nur Amazon S3-bezogene Ressourcen in die von StackSets ihnen erstellten Dateien einbeziehen kann, während ein anderes Team nur DynamoDB-Ressourcen einbeziehen kann. Zu diesem Zweck erstellen Sie eine Vertrauensbeziehung zwischen der benutzerdefinierten Administratorrolle für jede Gruppe und einer benutzerdefinierten Ausführungsrolle für jeden Ressourcensatz. Die benutzerdefinierte Ausführungsrolle definiert, in welche Stack-Ressourcen aufgenommen werden können StackSets. Die benutzerdefinierte Administratorrolle befindet sich im Administratorkonto, während sich die benutzerdefinierte Ausführungsrolle in jedem Zielkonto befindet, in dem Sie StackSets mithilfe der definierten Ressourcen etwas erstellen möchten. Anschließend aktivieren Sie bestimmte Benutzer und Gruppen, um die benutzerdefinierte Administratorrolle bei der Ausführung StackSets von Vorgängen zu verwenden.

Sie können beispielsweise benutzerdefinierte Administratorrollen A, B und C im Administratorkonto erstellen. Benutzer und Gruppen mit der Berechtigung zur Verwendung von Rolle A können Ressourcen erstellen, die die Stack-Ressourcen StackSets enthalten, die speziell in der benutzerdefinierten Ausführungsrolle X aufgeführt sind, aber nicht die Ressourcen in den Rollen Y oder Z oder Ressourcen, die in keiner Ausführungsrolle enthalten sind.

Bei der Aktualisierung einer StackSet muss der Benutzer explizit eine benutzerdefinierte Administratorrolle angeben, auch wenn es sich um dieselbe benutzerdefinierte Administratorrolle handelt, die StackSet zuvor für diese Rolle verwendet wurde. CloudFormation führt das Update mit der angegebenen benutzerdefinierten Administratorrolle durch, sofern der Benutzer über die entsprechenden Berechtigungen verfügt StackSet.

Entsprechend kann der Benutzer auch eine benutzerdefinierte Ausführungsrolle angeben. Wenn sie eine benutzerdefinierte Ausführungsrolle angeben, CloudFormation verwendet diese Rolle, um den Stack zu aktualisieren, sofern die oben genannten Anforderungen erfüllt sind. Wenn der Benutzer keine benutzerdefinierte Ausführungsrolle angibt, CloudFormation führt er das Update mithilfe der benutzerdefinierten Ausführungsrolle durch, die zuvor mit der verknüpft war StackSet, sofern der Benutzer berechtigt ist, Operationen an dieser Rolle auszuführen StackSet.

Administrator account

Erstellen Sie eine benutzerdefinierte Administratorrolle in Ihrem Administratorkonto, wie unter beschriebenSteuern Sie, welche Benutzer StackSet Operationen in bestimmten Zielkonten ausführen können. Fügen Sie eine Vertrauensstellung zwischen der benutzerdefinierten Administratorrolle und den benutzerdefinierten Ausführungsrollen hinzu, die sie verwenden soll.

Example Beispiel für eine Berechtigungsrichtlinie

Das folgende Beispiel ist eine Berechtigungsrichtlinie sowohl für die für das Zielkonto AWSCloudFormationStackSetExecutionRoledefinierten als auch für eine benutzerdefinierte Ausführungsrolle.

```
{
  "Version": "2012-10-17",
   "Statement": [
    {
      "Sid": "Stmt1487980684000",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole",
        "arn:aws:iam::*:role/custom_execution_role"
      ]
    }
  ]
}
```

Target accounts

Erstellen Sie in den Zielkonten, in denen Sie Ihre erstellen möchten StackSets, eine benutzerdefinierte Ausführungsrolle, die Berechtigungen für die Dienste und Ressourcen gewährt, zu denen Benutzer und Gruppen berechtigt sein sollen StackSets.

Example Beispiel für eine Berechtigungsrichtlinie

Das folgende Beispiel bietet die Mindestberechtigungen für StackSets sowie die Berechtigung zum Erstellen von Amazon DynamoDB-Tabellen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamoDb:createTable"
      ],
      "Resource": "*"
    }
  ]
}
```

Example Beispiel für eine Vertrauensrichtlinie

Sie müssen die folgende Vertrauensrichtlinie angeben, wenn Sie die Rolle zur Definition der Vertrauensbeziehung erstellen.

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

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Richten Sie Berechtigungen für bestimmte StackSet Operationen ein

Darüber hinaus können Sie Berechtigungen einrichten, mit denen Benutzer und Gruppen bestimmte StackSet Operationen ausführen können, z. B. Instanzen erstellen, aktualisieren, löschen StackSets oder stapeln können. Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für CloudFormation in der Service-Autorisierungs-Referenz.

Richten Sie globale Schlüssel ein, um Confused-Deputy-Probleme zu mindern.

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. Bei einem AWS dienstübergreifenden Identitätswechsel kann es zu einem Problem mit dem verwirrten Stellvertreter kommen. Ein serviceübergreifender Identitätswechsel kann auftreten, wenn ein Service (der Anruf-Service) einen anderen Service anruft (den aufgerufenen Service). Der Anruf-Service 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 Kontextschlüssel aws:SourceArnund die aws:SourceAccountglobalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, AWS CloudFormation StackSets die der Ressource einen anderen Dienst gewähren. Wenn Sie beide globalen Bedingungskontextschlüssel verwenden, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. 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. Beispiel, `arn:aws:cloudformation::123456789012:*`. Verwenden Sie nach Möglichkeit `aws:SourceArn`, da es spezifischer ist. Verwenden Sie `aws:SourceAccount` nur wenn Sie den richtigen ARN oder das ARN-Muster nicht ermitteln können.

When StackSets übernimmt die Administratorrolle in Ihrem Administratorkonto und StackSets füllt Ihre Administratorkonto-ID und Ihren StackSets Amazon-Ressourcennamen (ARN) aus. Sie können daher Bedingungen für die globalen Schlüssel `aws:SourceAccount` und `aws:SourceArn` in den Vertrauensbeziehungen definieren, um Confused-Deputy-Probleme zu vermeiden. Das folgende Beispiel zeigt, wie Sie die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globale Bedingung verwenden können, StackSets um das Problem des verwirrten Stellvertreters zu vermeiden.

Administrator account

Example Globale Schlüssel für **`aws:SourceAccount`** und **`aws:SourceArn`**

Definieren Sie bei der Verwendung StackSets die globalen Schlüssel `aws:SourceAccount` und `aws:SourceArn` in Ihrer AWSCloudFormationStackSetAdministrationRoleVertrauensrichtlinie, um Probleme mit verwirrenden Stellvertretern zu vermeiden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:cloudformation:*:111122223333:stackset/*"
        }
      }
    }
  ]
```

```
}
```

Example StackSets ARNs

Geben Sie StackSets ARNs für eine genauere Kontrolle Ihren zugehörigen Namen an.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudformation.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": [
                        "arn:aws:cloudformation:STACKSETS-
REGION:111122223333:stackset/STACK-SET-ID-1",
                        "arn:aws:cloudformation:STACKSETS-
REGION:111122223333:stackset/STACK-SET-ID-2"
                    ]
                }
            }
        }
    ]
}
```

Aktivieren Sie den vertrauenswürdigen Zugriff für StackSets mit AWS Organizations

Dieses Thema enthält Anweisungen zur Aktivierung des vertrauenswürdigen Zugriffs mit AWS Organizations. Dies ist für die kontoübergreifende Bereitstellung und StackSets die AWS-Regionen Verwendung von vom Dienst verwalteten Berechtigungen erforderlich. Informationen zur Verwendung

von selbstverwalteten Berechtigungen finden Sie stattdessen unterSelbstverwaltete Berechtigungen erteilen.

Bevor Sie eine StackSet mit vom Dienst verwalteten Berechtigungen erstellen, müssen Sie zunächst die folgenden Aufgaben ausführen:

- Aktivieren Sie alle Funktionen in AWS Organizations. Wenn nur Funktionen für die konsolidierte Abrechnung aktiviert sind, können Sie keine StackSet mit vom Service verwalteten Berechtigungen erstellen.
- Aktivieren Sie den vertrauenswürdigen Zugriff mit AWS Organizations. Diese Aktion ermöglicht CloudFormation das Erstellen einer dienstbezogenen Rolle im Verwaltungskonto. Wenn Sie nach der Aktivierung des vertrauenswürdigen Zugriffs eine StackSet mit vom Dienst verwalteten Berechtigungen erstellen, werden sowohl die erforderliche dienstverknüpfte Rolle als auch eine Servicerolle CloudFormation erstellt, die `stacksets-exec-*` in den Zielkonten (Mitgliedskonten) benannt ist.

  Wenn der vertrauenswürdige Zugriff aktiviert ist, können das Verwaltungskonto und die delegierten Administratorkonten dienstverwaltete StackSets Konten für ihre Organisation erstellen und verwalten.

Um den vertrauenswürdigen Zugriff zu aktivieren, müssen Sie ein Administratorbenutzer im Verwaltungskonto sein. Ein Administratorbenutzer ist ein Benutzer mit vollen Rechten für Ihren AWS-Konto. Weitere Informationen finden Sie im AWS -Kontenverwaltung Referenzhandbuch unter Einen Administratorbenutzer erstellen. Empfehlungen zum Schutz der Sicherheit des Verwaltungskontos finden Sie im AWS Organizations Benutzerhandbuch unter Bewährte Methoden für das Verwaltungskonto.

Um den vertrauenswürdigen Zugriff zu aktivieren

1. Melden Sie sich AWS als Administrator des Verwaltungskontos an und öffnen Sie die CloudFormation Konsole unterhttps://console.aws.amazon.com/cloudformation.
2. StackSets. Wenn der vertrauenswürdige Zugriff deaktiviert ist, wird ein Banner mit einer Aufforderung angezeigt, den vertrauenswürdigen Zugriff zu aktivieren.

3. Wählen Sie Vertrauenswürdigen Zugriff aktivieren aus.

   Der vertrauenswürdige Zugriff ist erfolgreich aktiviert, wenn das folgende Banner angezeigt wird.

> **Note**
>
> „Organisationszugriff aktivieren“ ist dasselbe wie „Organisationszugriff aktivieren“ und „Organisationszugriff deaktivieren“ ist dasselbe wie „Organisationszugriff deaktivieren“. Diese Bedingungen wurden auf der Grundlage von Marketingrichtlinien aktualisiert.

Deaktivieren des vertrauenswürdigen Zugriffs

Weitere Informationen finden Sie unter AWS CloudFormation StackSets und AWS Organizations im AWS Organizations Benutzerhandbuch.

Bevor Sie den vertrauenswürdigen Zugriff mit deaktivieren können AWS Organizations, müssen Sie die Registrierung aller delegierten Administratoren aufheben. Weitere Informationen finden Sie unter Einen delegierten Administrator registrieren.

> **Note**
>
> Informationen zur Aktivierung oder Deaktivierung des vertrauenswürdigen Zugriffs mithilfe von API-Vorgängen anstelle der Konsole finden Sie unter:
>
> - ActivateOrganizationsAccess
> - DeactivateOrganizationsAccess
> - DescribeOrganizationsAccess

Service-verknüpfte Rollen

Das Verwaltungskonto verwendet die AWSServiceRoleForCloudFormationStackSetsOrgAdmindienstverknüpfte Rolle. Sie können diese Rolle nur ändern oder löschen, wenn der vertrauenswürdige Zugriff mit AWS Organizations deaktiviert ist.

Jedes Zielkonto verwendet eine AWSServiceRoleForCloudFormationStackSetsOrgMemberdienstverknüpfte Rolle. Sie können diese Rolle nur unter zwei Bedingungen ändern oder löschen: wenn der vertrauenswürdige Zugriff mit deaktiviert AWS Organizations ist oder wenn das Konto aus der Zielorganisation oder Organisationseinheit (OU) entfernt wird.

Weitere Informationen finden Sie unter AWS CloudFormation StackSets und AWS Organizations im AWS Organizations Benutzerhandbuch.

Registrieren Sie ein delegiertes Administrator-Mitgliedskonto

Zusätzlich zum Verwaltungskonto Ihrer Organisation können Mitgliedskonten mit delegierten Administratorberechtigungen dienstverwaltete Berechtigungen für die Organisation erstellen und verwalten StackSets . StackSets mit vom Dienst verwalteten Berechtigungen werden im Verwaltungskonto erstellt, auch solche, die von delegierten Administratoren StackSets erstellt wurden. Um als delegierter Administrator für Ihre Organisation registriert zu sein, muss Ihr Mitgliedskonto in der Organisation sein. Weitere Informationen zum Beitritt zu einer Organisation finden Sie unter Einen zum Beitritt AWS-Konto zu Ihrer Organisation einladen.

Ihre Organisation kann bis zu fünf registrierte delegierte Administratoren gleichzeitig haben. Delegierte Administratoren können wählen, ob die Bereitstellung für alle Konten in Ihrer Organisation oder für bestimmte OUs Konten erfolgen soll. Der vertrauenswürdige Zugriff mit AWS Organizations muss aktiviert werden, bevor delegierte Administratoren die Bereitstellung auf Konten vornehmen können, die von Organizations verwaltet werden. Weitere Informationen finden Sie unter Aktivieren Sie den vertrauenswürdigen Zugriff für StackSets mit AWS Organizations.

> **Important**
>
> Bitte beachten Sie Folgendes:
>
> - Delegierte Administratoren haben volle Berechtigungen für die Bereitstellung auf Konten Ihrer Organisation. Das Verwaltungskonto kann delegierte Administratorrechte nicht einschränken, um die Bereitstellung auf bestimmte Vorgänge OUs oder die Ausführung bestimmter StackSet Vorgänge zu beschränken.
> - Stellen Sie sicher, dass Ihre delegierten Administratoren über die `organizations:ListDelegatedAdministrators` entsprechenden Rechte verfügen, um mögliche Fehler zu vermeiden.

Sie können delegierte Administratoren für Ihre Organisation in den folgenden Regionen registrieren: USA Ost (Ohio), USA Ost (Nord-Virginia), USA West (Nordkalifornien), USA West (Oregon), Asien-Pazifik (Mumbai), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio), Kanada (Zentral), Europa (Frankfurt), Europa (Irland), Europa (London), Europa

(Paris), Europa (Stockholm), Israel (Tel Aviv), Südamerika (São Paulo), AWS GovCloud (US-Ost) und AWS GovCloud (US-West).

Sie können delegierte Administratoren mit der AWS CloudFormation Konsole, AWS CLI, oder registrieren und deregistrieren AWS SDKs.

So registrieren Sie einen delegierten Administrator (Konsole)

1. Melden Sie sich AWS als Administrator des Verwaltungskontos an und öffnen Sie die AWS CloudFormation Konsole unter. https://console.aws.amazon.com/cloudformation/
2. StackSets.
3. Wählen Sie unter Delegierte Administratoren die Option Delegierten Administrator registrieren aus.
4. Wählen Sie im Dialogfeld Delegierten Administrator registrieren die Option Delegierten Administrator registrieren aus.

   Die Erfolgsmeldung zeigt an, dass das Mitgliedskonto erfolgreich als delegierter Administrator registriert wurde.

So heben Sie die Registrierung eines delegierten Administrators auf (Konsole)

1. Melden Sie sich AWS als Administrator des Verwaltungskontos an und öffnen Sie die AWS CloudFormation Konsole unterhttps://console.aws.amazon.com/.
2. StackSets.
3. Wählen Sie unter Delegierte Administratoren das Konto aus, das Sie abmelden möchten, und wählen Sie dann Registrierung aufheben aus.

   Die Erfolgsmeldung zeigt an, dass das Mitgliedskonto erfolgreich als delegierter Administrator abgemeldet wurde.

   Sie können dieses Konto jederzeit wieder registrieren.

Um einen delegierten Administrator zu registrieren ()AWS CLI

1. Öffnen Sie das AWS CLI.
2. Führen Sie den Befehl `register-delegated-administrator` aus.

```
$ aws organizations register-delegated-administrator \
```

```
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Führen Sie den Befehl `list-delegated-administrators` aus, um zu überprüfen, ob das angegebene Mitgliedskonto erfolgreich als delegierter Administrator registriert wurde.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Um einen delegierten Administrator abzumelden ()AWS CLI

1. Öffnen Sie das. AWS CLI
2. Führen Sie den Befehl `deregister-delegated-administrator` aus.

```
$ aws organizations deregister-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Führen Sie den Befehl `list-delegated-administrators` aus, um zu überprüfen, ob das angegebene Mitgliedskonto erfolgreich als delegierter Administrator abgemeldet wurde.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Sie können dieses Konto jederzeit wieder registrieren.

Erste Schritte mit StackSets der Verwendung einer Beispielvorlage

Dieses Tutorial hilft Ihnen beim Einstieg in die StackSets AWS Management Console Verwendung von. Es führt Sie durch die Erstellung einer Vorlage StackSet anhand einer Beispielvorlage. Sie erfahren, wie Sie Stacks in mehreren Regionen bereitstellen, StackSet Abläufe überwachen und sich die Ergebnisse ansehen.

In diesem Tutorial erstellen Sie eine, StackSet die AWS Config AWS-Konto in Ihren Regionen USA West (Oregon) (`us-west-2`) und USA Ost (Nord-Virginia) () aktiviert wird. `us-east-1` Mit StackSets können Sie Stacks für mehrere Konten und Regionen mit einem einzigen Vorgang erstellen, aktualisieren oder löschen, was es zu einer idealen Lösung für die Verwaltung von Infrastrukturen

in großem Maßstab macht. In diesem Tutorial wird zwar der Einfachheit halber ein einziges Konto verwendet, es zeigt jedoch effektiv die regionsübergreifenden Funktionen von. StackSets

Die Beispielvorlage ist im folgenden S3-Bucket verfügbar: https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/Enable AWSConfig .yml.

> **Note**
>
> StackSets ist kostenlos, aber die AWS Ressourcen, die Sie damit erstellen, werden Ihnen in Rechnung gestellt, wie AWS Config in diesem Tutorial. Weitere Informationen finden Sie unter AWS Config Preise.

Themen

- Voraussetzungen
- Erstellen Sie eine StackSet mit einer Beispielvorlage aus der Konsole
- Überwachen Sie StackSet die Erstellung
- StackSet Ergebnisse anzeigen
- Aktualisiere deine StackSet
- Füge Stapel zu deinem hinzu StackSet
- Bereinigen
- Nächste Schritte

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllt haben:

- Sie müssen die erforderlichen IAM-Rollen für selbstverwaltete Berechtigungen eingerichtet haben. Um Stacks in einem einzigen Konto zu erstellen StackSet und bereitzustellen, benötigen Sie die folgenden Rollen in Ihrem Konto:
  - `AWSCloudFormationStackSetAdministrationRole`
  - `AWSCloudFormationStackSetExecutionRole`

  Eine ausführliche Anleitung zur Einrichtung dieser Rollen finden Sie unterSelbstverwaltete Berechtigungen erteilen.

Erstellen Sie eine StackSet mit einer Beispielvorlage aus der Konsole

Um eine zu erstellen StackSet , die Folgendes ermöglicht AWS Config

1. Öffnen Sie die CloudFormation-Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die AWS-Region aus, von der StackSet aus Sie das verwalten möchten.

   Sie können jede Region auswählen, die unterstützt wird StackSets. Die Region, die Sie auswählen, hat keinen Einfluss darauf, in welchen Regionen Sie mit Ihrem bereitstellen können StackSet.
3. StackSets.
4. Wählen Sie oben StackSetsauf der Seite die Option Erstellen aus StackSet.
5. Wählen Sie unter Berechtigungen die Option Self-Service-Berechtigungen und dann die IAM-Rollen aus, die Sie in den Voraussetzungen erstellt haben.
   - Wählen Sie für die IAM-Administratorrolle die Option. AWSCloudFormationStackSetAdministrationRole
   - Wählen Sie für den Namen der IAM-Ausführungsrolle die Option. AWSCloudFormationStackSetExecutionRole
6. Wählen Sie unter Prerequisite - Prepare template (Voraussetzung: Vorbereiten der Vorlage) die Option Use a sample template (Beispielvorlage verwenden) aus.
7. Wählen Sie unter Beispielvorlage auswählen die Option AWS Config Vorlage aktivieren aus. Wählen Sie anschließend Weiter.

   Diese Vorlage erstellt die Ressourcen, die für die Aktivierung AWS Config in Ihrem Konto erforderlich sind, einschließlich eines Konfigurationsrekorders und eines Bereitstellungskanals.
8. Geben Sie auf der Seite „ StackSet Details angeben“ als StackSet Namen ein**`my-awsconfig-stackset`**.
9. Geben Sie als StackSet Beschreibung ein**`A StackSet that enables Config across multiple Regions`**.
10. Konfigurieren Sie unter Parameter die AWS Config Einstellungen wie folgt:
    a. Behalten Sie für Alle Ressourcentypen Support den Standardwert true bei, um alle unterstützten Ressourcentypen aufzuzeichnen.

b. Behalten Sie für „Globale Ressourcentypen einbeziehen“ den Standardwert False bei, um globale Ressourcen wie IAM-Rollen auszuschließen.

c. <All>Lassen Sie die Einstellung Liste der Ressourcentypen, falls nicht alle unterstützt werden, auf stehen.

d. Ersetzen Sie < DeployToAnyRegion > für Die Region, die die mit dem Dienst Config verknüpfte Rollenressource enthält, durch. **`us-west-2`**

   Das bedeutet, dass die angegebene serviceverknüpfte Rolle nur erstellt `AWSServiceRoleForConfig` wird, wenn ein Stack in der Region USA West (Oregon) bereitgestellt wird. Sie wählen die Bereitstellungsregionen später in diesem Verfahren aus.

e. Wählen Sie für die Aufzeichnungsfrequenz des Konfigurationsrekorders die Option TÄGLICHE Aufnahme aus.

11. Wählen Sie Next (Weiter), um fortzufahren.

12. Wählen Sie auf der Seite mit den StackSet Konfigurationsoptionen die Option Neues Tag hinzufügen aus und fügen Sie ein Tag hinzu, indem Sie ein Schlüssel-Wert-Paar angeben:

    a. Geben Sie für Key (Schlüssel) **`Stage`** ein.

    b. Geben Sie für Wert **`Test`** ein.

    Tags, auf die Sie StackSets sich beziehen, werden auf Ressourcen angewendet, die durch Ihre Stacks erstellt wurden.

13. Wählen Sie für die Ausführungskonfiguration die Option Aktiv aus, um CloudFormation die optimierte Betriebsbehandlung zu aktivieren:

    - Konfliktfreie Operationen werden gleichzeitig ausgeführt, um die Bereitstellungszeiten zu verkürzen.
    - Widersprüchliche Vorgänge werden automatisch in die Warteschlange gestellt und in der Reihenfolge verarbeitet, in der sie angefordert wurden.

    Solange Vorgänge ausgeführt werden oder sich in der Warteschlange befinden, werden alle eingehenden Vorgänge in die CloudFormation Warteschlange gestellt, auch wenn sie nicht miteinander in Konflikt stehen. Während dieser Zeit können Sie die Ausführungseinstellungen nicht ändern.

14. Wählen Sie Weiter aus.

15. Wählen Sie auf der Seite Bereitstellungsoptionen festlegen für Stacks hinzufügen zu die StackSet Option Neue Stacks bereitstellen aus.
16. Wählen Sie für Accounts (Konten) die Option Deploy stacks in accounts (Stacks in Konten bereitstellen) aus.
17. Geben Sie in das Textfeld Ihre AWS-Konto ID ein.
18. Wählen Sie unter Regionen angeben die folgenden Regionen in dieser Reihenfolge aus:

    1. Region USA West (Oregon) (`us-west-2`)
    2. Region USA Ost (Nord-Virginia) (`us-east-1`)

    Verwenden Sie den Aufwärtspfeil neben der Region USA West (Oregon), um sie bei Bedarf zum ersten Eintrag in der Liste zu verschieben. Die Reihenfolge der Regionen bestimmt ihre Bereitstellungsreihenfolge.
19. Konfigurieren Sie für Bereitstellungsoptionen die folgenden Einstellungen:

    a. Für Maximum concurrent accounts (Maximale Anzahl paralleler Konten) behalten Sie die Standardwerte Number (Zahl) und 1 bei.

       Bei Bereitstellungen mit mehreren Konten bedeutet diese Einstellung, dass Ihr Stack jeweils nur in einem Konto CloudFormation bereitgestellt wird.

    b. Behalten Sie für Failure tolerance (Fehlertoleranz) die Standardwerte Number (Zahl) und 0 bei.

       Das bedeutet, dass maximal Null-Stack-Bereitstellungen in einer der von Ihnen angegebenen Regionen fehlschlagen können, bevor die Bereitstellung in der aktuellen Region CloudFormation gestoppt und Bereitstellungen in den übrigen Regionen storniert werden.

    c. Wählen Sie für Regionsparallelität die Option Sequentiell (Standard).

       Diese Einstellung stellt sicher, dass Bereitstellungen in einer Region CloudFormation abgeschlossen werden, bevor zur nächsten übergegangen wird.

    d. Behalten Sie für den Parallelitätsmodus die Standardeinstellung Strikte Fehlertoleranz bei.

       Bei Bereitstellungen mit mehreren Konten reduziert dies den Grad der Parallelität der Konten bei Ausfällen, sodass die Fehlertoleranz +1 eingehalten wird.
20. Wählen Sie Weiter aus.

21. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie im entsprechenden Abschnitt Bearbeiten aus.
22. Wenn Sie bereit sind, Ihre zu erstellen StackSet, wählen Sie Senden.

Überwachen Sie StackSet die Erstellung

Nachdem Sie „Senden“ ausgewählt haben, CloudFormation beginnt er mit der Erstellung Ihrer Stacks StackSet und der Bereitstellung von Stacks für die angegebenen Regionen in Ihrem Konto. Die StackSet Detailseite wird automatisch geöffnet, auf der Sie den Fortschritt des Vorgangs verfolgen können.

Um die StackSet Erstellung zu überwachen

1. Auf der StackSet Detailseite wird standardmäßig die Registerkarte Vorgänge angezeigt, auf der der aktuelle Vorgang angezeigt wird.
2. Der Vorgangsstatus sollte `RUNNING` anfänglich sein. CloudFormationerstellt Stapel in den Regionen, die Sie entsprechend den von Ihnen konfigurierten Bereitstellungsoptionen angegeben haben.
3. Um weitere Informationen zu dem Vorgang zu erhalten, wählen Sie die Vorgangs-ID in der Liste aus.
4. Auf der Seite mit den Vorgangsdetails können Sie den Status der Stack-Instances einsehen, die in jeder Region erstellt werden.
5. Warten Sie, bis sich der Betriebsstatus auf ändert`SUCCEEDED`, was darauf hinweist, dass die StackSet und alle zugehörigen Stack-Instances erfolgreich erstellt wurden.

StackSet Ergebnisse anzeigen

Nach Abschluss der StackSet Erstellung können Sie sich die bereitgestellten Stack-Instances ansehen und überprüfen, AWS Config ob sie in Ihrem Konto in den angegebenen Regionen aktiviert wurden.

Um die StackSet Ergebnisse anzusehen

1. Wählen Sie auf der StackSet Detailseite den Tab Stack-Instances aus.

2. Sie sollten eine Liste der Stack-Instances sehen, die in Ihrem Konto in den angegebenen Regionen erstellt wurden. Jede Stack-Instance sollte den Status haben`SUCCEEDED`, was darauf hinweist, dass sie erfolgreich bereitgestellt wurde.
3. Um zu überprüfen, ob dies in Ihrem Konto aktiviert AWS Config ist, können Sie die AWS Config Konsole in jeder der bereitgestellten Regionen überprüfen.

Aktualisiere deine StackSet

Nachdem Sie Ihre erstellt haben StackSet, möchten Sie sie möglicherweise aktualisieren, um die Parameterwerte zu ändern oder weitere Regionen hinzuzufügen. In diesem Abschnitt erfahren Sie, wie Sie den Parameter für die AWS Config Aufnahmefrequenz aktualisieren.

Um deine zu aktualisieren StackSet

1. Wählen Sie auf der StackSetsSeite Ihre aus**`my-awsconfig-stackset`**.
2. Wenn Sie das StackSet ausgewählt haben, wählen Sie im Menü Aktionen die Option StackSet Details bearbeiten aus.
3. Wählen Sie auf der Seite „Vorlage auswählen“ unter Voraussetzung — Vorlage vorbereiten die Option Aktuelle Vorlage verwenden aus.
4. Wählen Sie Weiter aus.
5. Suchen Sie auf der Seite „ StackSet Details angeben“ unter Parameter nach der Aufzeichnungsfrequenz des Konfigurationsrekorders und ändern Sie sie von TÄGLICH auf KONTINUIERLICH.
6. Wählen Sie Weiter aus.
7. Lassen Sie auf der Seite „ StackSet Optionen konfigurieren“ die Einstellungen unverändert und wählen Sie Weiter.
8. Geben Sie auf der Seite „Bereitstellungsoptionen festlegen“ Ihre Konto-ID und dieselben Regionen an, die Sie bei der Erstellung von verwendet haben StackSet.
9. Behalten Sie für die Bereitstellungsoptionen dieselben Einstellungen wie zuvor bei.
10. Wählen Sie Weiter aus.
11. Überprüfen Sie auf der Seite „Überprüfen“ Ihre Änderungen und wählen Sie „Senden“.
12. CloudFormation beginnt mit der Aktualisierung Ihres StackSet. Sie können den Fortschritt auf der Registerkarte Operationen auf der StackSet Detailseite überwachen.

Füge Stapel zu deinem hinzu StackSet

Sie können Ihrem weitere Stapel hinzufügen, StackSet indem Sie ihn in weiteren Regionen bereitstellen. In diesem Abschnitt erfahren Sie, wie Sie Stacks zu einer neuen Region hinzufügen.

Um Stacks zu deinem hinzuzufügen StackSet

1. Wählen Sie auf der StackSetsSeite Ihre **`my-awsconfig-stackset`** aus.
2. Wenn das StackSet ausgewählt ist, wählen Sie im Menü Aktionen die Option Stapel hinzufügen zu StackSet aus.
3. Wählen Sie auf der Seite „Bereitstellungsoptionen festlegen“ für Stacks hinzufügen zu die StackSet Option Neue Stacks bereitstellen aus.
4. Wählen Sie für Konten die Option Stacks in Konten bereitstellen aus und geben Sie Ihre Konto-ID ein.
5. Wählen Sie unter Regionen angeben eine neue Region aus, z. B. Europa (Irland) (`eu-west-1`).
6. Behalten Sie für die Bereitstellungsoptionen dieselben Einstellungen wie zuvor bei.
7. Wählen Sie Weiter aus.
8. Lassen Sie auf der Seite „Überschreibungen angeben“ die angegebenen Eigenschaftswerte unverändert und klicken Sie auf Weiter.
9. Überprüfen Sie auf der Seite Review (Überprüfen) Ihre Auswahl und wählen Sie anschließend Submit (Einsenden) aus.
10. CloudFormation beginnt mit der Erstellung neuer Stacks in der angegebenen Region. Sie können den Fortschritt auf der Registerkarte „Operationen“ der StackSet Detailseite überwachen.

Bereinigen

Um zu vermeiden, dass Gebühren für unerwünschte AWS Config Ressourcen anfallen, sollten Sie aufräumen, indem Sie die Stacks von Ihrem Computer löschen StackSet, die Stacks StackSet selbst löschen und die IAM-Rollen entfernen, die Sie für dieses Tutorial erstellt haben. Da alle Ressourcen in Ihrem Konto bereitgestellt werden, ist die Bereinigung unkompliziert.

Um Stapel von deinem zu löschen StackSet

1. Wählen Sie auf der StackSetsSeite Ihre **`my-awsconfig-stackset`** aus.
2. Wenn das StackSet ausgewählt ist, wählen Sie im Menü Aktionen die Option Stapel löschen StackSet aus.

3. Wählen Sie auf der Seite „Bereitstellungsoptionen festlegen“ für Konten die Option Stacks in Konten bereitstellen aus und geben Sie Ihre Konto-ID ein.
4. Wählen Sie unter Regionen angeben alle Regionen aus, in denen Sie Stacks bereitgestellt haben.
5. Behalten Sie für Bereitstellungsoptionen die Standardeinstellungen bei.
6. Stellen Sie sicher, dass Stacks beibehalten nicht aktiviert ist, damit die Stapel und ihre Ressourcen gelöscht werden.
7. Wählen Sie Weiter aus.
8. Überprüfen Sie auf der Seite Review (Überprüfen) Ihre Auswahl und wählen Sie anschließend Submit (Einsenden) aus.
9. CloudFormation beginnt mit dem Löschen der Stacks von Ihrem. StackSet Sie können den Fortschritt auf der Registerkarte „Operationen“ der StackSet Detailseite überwachen.

Um deine zu löschen StackSet

1. Nachdem alle Stapel gelöscht wurden, wählen Sie auf der StackSetsSeite Ihre **`my-awsconfig-stackset`** aus.
2. Wenn Sie StackSet ausgewählt sind, wählen Sie im Aktionsmenü die Option Löschen StackSet aus.
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie „Löschen“.

Um die IAM-Dienstrollen zu löschen

Da Sie die Bereitstellung nur für Ihr Konto vorgenommen haben, müssen Sie nur die IAM-Rollen aus diesem einzelnen Konto löschen. Dadurch ist die Bereinigung wesentlich einfacher als bei Bereitstellungen mit mehreren Konten.

1. Öffnen Sie die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Roles (Rollen) aus.
3. Geben Sie in das Suchfeld ein, um nach den Rollen **`AWSCloudFormationStackSet`** zu suchen, die Sie für dieses Tutorial erstellt haben.
4. Aktivieren Sie das Kontrollkästchen neben AWSCloudFormationStackSetAdministrationRole.
5. Wählen Sie oben auf der Seite die Option Löschen aus.
6. Geben Sie im Bestätigungsdialogfeld die Eingabe ein **`delete`** und wählen Sie Löschen.

7. Wiederholen Sie den gleichen Vorgang, um das zu löschen AWSCloudFormationStackSetExecutionRole.

Nach dem Löschen von StackSet verbleibt AWS-Region aufgrund des `DeletionPolicy` Attributs auf der `AWS::S3::Bucket` Ressource jeweils ein Amazon S3 S3-Bucket. Dadurch bleiben Ihre AWS Config Verlaufsdaten erhalten. Wenn Sie diese Daten nicht mehr benötigen, können Sie den Bucket problemlos manuell löschen. Bevor Sie einen Bucket löschen können, müssen Sie ihn zuerst leeren. Beim Leeren eines Buckets werden alle darin enthaltenen Objekte gelöscht.

Um die Amazon S3 S3-Buckets zu leeren und zu löschen

1. Öffnen Sie die Amazon S3-Konsole.
2. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Buckets.
3. In der Buckets-Liste finden Sie Buckets, die dafür StackSet in jeder Region, in der Sie sie bereitgestellt haben, erstellt wurden. Wählen Sie die Option neben dem Namen des Buckets aus, der dafür erstellt wurde StackSet, und wählen Sie dann Leer aus.
4. Bestätigen Sie auf der Seite Bucket leeren, dass Sie den Bucket leeren möchten, indem Sie **`permanently delete`** in das Textfeld eingeben, und wählen Sie dann Leeren.
5. Überwachen Sie den Fortschritt des Vorgangs zum Entleeren von Buckets auf der Seite Bucket entleeren: Status).
6. Um zur Bucket-Liste zurückzukehren, wählen Sie Exit (Beenden) aus.
7. Wählen Sie die Option neben dem Namen des Buckets aus und klicken Sie dann auf Löschen.
8. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den Namen des Buckets ein und wählen Sie dann Bucket löschen.
9. Überwachen Sie den Fortschritt des Bucket-Löschvorgangs in der Buckets-Liste. Wenn Amazon S3 das Löschen des Buckets abgeschlossen hat, entfernt es den Bucket aus der Liste.
10. Wiederholen Sie diesen Vorgang für jeden Bucket, der von den StackSet in den verschiedenen Regionen erstellt wurde.

Nächste Schritte

Herzlichen Glückwunsch! Sie haben StackSet mithilfe einer Beispielvorlage erfolgreich eine Vorlage erstellt, Stacks in mehreren Regionen Ihres Kontos bereitgestellt, aktualisiert StackSet, weitere Stacks hinzugefügt und Ihre Ressourcen bereinigt. Indem Sie sich auf die Bereitstellung

eines einzelnen Kontos konzentriert haben, haben Sie den Bereinigungsprozess vereinfacht und gleichzeitig die wichtigsten Funktionen von für mehrere Regionen kennengelernt. StackSets

Weitere Informationen StackSets finden Sie in den folgenden Themen:

- Überschreiben Sie Parameterwerte auf Stacks in Ihrem CloudFormation StackSet— Erfahren Sie, wie Sie Parameterwerte für bestimmte Konten und Regionen überschreiben können.
- AWS CloudFormation StackSets Mit vom Service verwalteten Berechtigungen erstellen— Erfahren Sie mehr über StackSets das Erstellen von Bereitstellungen mit mehreren Konten mit. AWS Organizations

AWS CloudFormation StackSets Mit selbstverwalteten Berechtigungen erstellen

In diesem Thema wird beschrieben, wie Sie StackSets mithilfe von selbstverwalteten Berechtigungen Stacks für die regionsübergreifende AWS-Konten Bereitstellung erstellen.

> **Note**
>
> Bevor Sie fortfahren, erstellen Sie die IAM-Dienstrollen, die erforderlich sind, StackSets um eine vertrauenswürdige Beziehung zwischen dem Konto, von dem StackSet aus Sie die Daten verwalten, und dem Konto, für das Sie Stacks bereitstellen, herzustellen. Weitere Informationen finden Sie unter Selbstverwaltete Berechtigungen erteilen.

Themen

- Erstellen Sie eine StackSet mit selbstverwalteten Berechtigungen (Konsole)
- Erstellen Sie eine StackSet mit selbstverwalteten Berechtigungen ()AWS CLI

Erstellen Sie eine StackSet mit selbstverwalteten Berechtigungen (Konsole)

Um eine zu erstellen StackSet

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, von der aus Sie AWS-Region die StackSet verwalten möchten.
3. StackSets.
4. Wählen Sie oben StackSetsauf der Seite die Option Erstellen aus StackSet.
5. Wählen Sie unter Berechtigungen die Option Self-Service-Berechtigungen und dann die IAM-Rollen aus, die Sie erstellt haben.
6. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorlage ist bereit aus.
7. Wählen Sie unter Specify template (Vorlage angeben) entweder die URL für den S3-Bucket, der Ihre Stack-Vorlage enthält, aus oder laden Sie eine Stack-Vorlagendatei hoch. Wählen Sie anschließend Weiter.
8. Geben Sie auf der Seite „ StackSet Details angeben“ einen Namen für die ein StackSet, geben Sie alle Parameter an, und klicken Sie dann auf Weiter.
9. Wählen Sie Next (Weiter), um fortzufahren.
10. Geben Sie auf der Seite StackSet Optionen konfigurieren unter Tags alle Tags an, die auf Ressourcen in Ihrem Stack angewendet werden sollen. Weitere Informationen zur Verwendung von Tags finden Sie im AWSAWS Fakturierung und Kostenmanagement Benutzerhandbuch unter Organisieren und Nachverfolgen von AWS Kosten mithilfe von Kostenzuordnungs-Tags.
11. Wählen Sie für die Ausführungskonfiguration die Option Aktiv aus, um CloudFormation die optimierte Betriebsbehandlung zu aktivieren:

    - Konfliktfreie Operationen werden gleichzeitig ausgeführt, um die Bereitstellungszeiten zu verkürzen.
    - Widersprüchliche Vorgänge werden automatisch in die Warteschlange gestellt und in der Reihenfolge verarbeitet, in der sie angefordert wurden.

    Solange Vorgänge ausgeführt werden oder sich in der Warteschlange befinden, werden alle eingehenden Vorgänge in die CloudFormation Warteschlange gestellt, auch wenn sie nicht miteinander in Konflikt stehen. Während dieser Zeit können Sie die Ausführungseinstellungen nicht ändern.
12. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie unter Capabilities die Option Ich bestätige, dass diese Vorlage IAM-Ressourcen erstellen kann aus, um anzugeben, dass Sie IAM-Ressourcen in der Vorlage verwenden möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.
13. Wählen Sie Weiter aus.

14. Wählen Sie auf der Seite „Bereitstellungsoptionen festlegen“ für Stacks hinzufügen zu die Option Neue Stacks StackSet bereitstellen aus.
15. Wählen Sie für Accounts (Konten) die Option Deploy stacks in accounts (Stacks in Konten bereitstellen) aus. Fügen Sie Ihre AWS-Konto Zielzahlen in das Textfeld ein und trennen Sie mehrere Zahlen durch Kommas.

    > **Note**
    >
    > Sie können Ihre Administratorkonto-ID angeben, wenn Sie Stacks auch in diesem Konto bereitstellen möchten.

16. Wählen Sie unter Regionen angeben die Regionen aus, in denen Sie Stacks bereitstellen möchten.
17. Gehen Sie für Bereitstellungsoptionen wie folgt vor:
    - Geben Sie unter Maximale Anzahl gleichzeitiger Konten an, wie viele Konten gleichzeitig verarbeitet werden.
    - Geben Sie für Fehlertoleranz die maximale Anzahl von Kontoausfällen an, die pro Region zulässig sind. Sobald dieses Limit erreicht ist, wird der Vorgang beendet und nicht in andere Regionen übertragen.
    - Wählen Sie für die Regionenparallelität aus, wie Regionen verarbeitet werden sollen: Sequentiell (jeweils eine Region) oder Parallel (mehrere Regionen gleichzeitig).
    - Wählen Sie für den Parallelitätsmodus aus, wie sich Parallelität während der Ausführung des Vorgangs verhält.
      - Strikte Fehlertoleranz — Reduziert den Grad der Parallelität der Konten bei Ausfällen und bleibt dabei innerhalb der Fehlertoleranz +1.
      - Weiche Fehlertoleranz — Behält die angegebene Parallelitätsstufe (den Wert für Maximale Anzahl gleichzeitiger Konten) unabhängig von Ausfällen bei.
18. Wählen Sie Weiter aus.
19. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie im entsprechenden Bereich die Option Bearbeiten aus.
20. Wenn Sie bereit sind, Ihre zu erstellen StackSet, wählen Sie Senden.

    CloudFormation beginnt mit der Erstellung Ihres StackSet. Sehen Sie sich den Fortschritt und den Status der Erstellung der Stacks auf der StackSet Detailseite an, die geöffnet wird, wenn Sie „Senden“ wählen. StackSet

Erstellen Sie eine StackSet mit selbstverwalteten Berechtigungen ()AWS CLI

Folgen Sie den Schritten in diesem Abschnitt, um Folgendes AWS CLI zu verwenden:

- Erstellen Sie den StackSet Container.
- Stellen Sie Stack-Instanzen bereit.

Um eine zu erstellen StackSet

1. Verwenden Sie den create-stack-setBefehl, um einen neuen StackSet Namen zu erstellen*my-stackset*. Das folgende Beispiel verwendet eine in einem S3-Bucket gespeicherte Vorlage und enthält einen Parameter, der a *KeyPairName* mit dem Wert festlegt*TestKey*.

```
aws cloudformation create-stack-set \
  --stack-set-name my-stackset \
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/
MyApp.template \
  --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. Wenn Ihr create-stack-set Befehl abgeschlossen ist, führen Sie ihn aus, um zu überprüfen, ob Ihr list-stack-setsBefehl erstellt StackSet wurde. Sie sollten Ihr neues StackSet in den Ergebnissen sehen.

```
aws cloudformation list-stack-sets
```

3. Verwenden Sie den create-stack-instancesBefehl, um Stapel in Ihrem StackSet bereitzustellen. Im folgenden Beispiel werden Stacks in zwei AWS-Konten (*account_ID_1*und*account_ID_2*) in zwei Regionen (*us-west-2*und) bereitgestellt. *us-east-1*

   Legen Sie mithilfe der Option Einstellungen für die gleichzeitige Kontoverarbeitung und andere Bereitstellungseinstellungen fest. `--operation-preferences` In diesem Beispiel werden auf der Anzahl basierende Einstellungen verwendet. Beachten Sie, dass dieser `MaxConcurrentCount` Wert `FailureToleranceCount` + 1 nicht überschreiten darf. Verwenden Sie für prozentuale Einstellungen stattdessen `FailureTolerancePercentage` oder`MaxConcurrentPercentage`.

```
aws cloudformation create-stack-instances \
  --stack-set-name my-stackset \
```

```
  --accounts account_ID_1 account_ID_2 \
  --regions us-west-2 us-east-1 \
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Weitere Informationen finden Sie unter CreateStackInstances in der AWS CloudFormation -API-Referenz.

4. Verwenden Sie den describe-stack-set-operationBefehl, um zu überprüfen, ob Ihre Stacks erfolgreich erstellt wurden. Geben Sie für die `--operation-id` Option die Vorgangs-ID an, die als Teil der create-stack-instances Ausgabe zurückgegeben wurde.

```
aws cloudformation describe-stack-set-operation \
  --stack-set-name my-stackset \
  --operation-id operation_ID
```

AWS CloudFormation StackSets Mit vom Service verwalteten Berechtigungen erstellen

Mit vom Dienst verwalteten Berechtigungen können Sie Stacks für Konten bereitstellen, die von AWS Organizations in bestimmten Regionen verwaltet werden. Bei diesem Modell müssen Sie nicht die erforderlichen IAM-Rollen erstellen, sondern CloudFormation erstellt die IAM-Rollen in Ihrem Namen. Weitere Informationen finden Sie unter Aktivieren Sie den vertrauenswürdigen Zugriff.

Themen

- Überlegungen
- Erstellen Sie eine StackSet mit vom Service verwalteten Berechtigungen (Konsole)
- Erstellen Sie eine StackSet mit vom Service verwalteten Berechtigungen ()AWS CLI

Überlegungen

Bevor Sie eine StackSet mit vom Service verwalteten Berechtigungen erstellen, sollten Sie Folgendes berücksichtigen:

- StackSets mit vom Dienst verwalteten Berechtigungen werden im Verwaltungskonto erstellt, auch solche, StackSets die von delegierten Administratoren erstellt wurden.

- Sie StackSet können Ihre gesamte Organisation (einschließlich aller Konten) oder bestimmte Organisationseinheiten (OUs) als Ziel angeben. Wenn Sie eine übergeordnete Organisationseinheit ins StackSet Visier nehmen, zielen Sie auch auf jedes Kind ab OUs. Wenn Ihre StackSet Ziele spezifisch sind OUs, OUs sind standardmäßig alle Konten innerhalb dieser Ziele enthalten. Sie können jedoch mithilfe der Kontofilteroptionen auf bestimmte Konten abzielen.
- Mehrere StackSets können auf dieselbe Organisation oder Organisationseinheit abzielen.
- Sie StackSet können nicht auf Konten außerhalb Ihrer Organisation abzielen.
- Sie StackSet können keine verschachtelten Stacks bereitstellen.
- CloudFormation stellt keine Stacks für das Verwaltungskonto der Organisation bereit, auch wenn sich das Verwaltungskonto in Ihrer Organisation oder in einer Organisationseinheit in Ihrer Organisation befindet.
- Die automatische Bereitstellung ist auf der StackSet Ebene festgelegt. Sie können automatische Bereitstellungen nicht selektiv für OUs Konten oder Regionen anpassen.
- Die Berechtigungen der IAM-Prinzipalentität (Benutzer, Rolle oder Gruppe), mit der Sie sich beim Verwaltungskonto anmelden, bestimmen, ob Sie für die Bereitstellung autorisiert sind. StackSets Eine IAM-Beispielrichtlinie, die Berechtigungen für die Bereitstellung in einer Organisation erteilt, finden Sie unter Beschränken Sie Stack-Set-Operationen auf der Grundlage von Region und Ressourcentypen.
- Delegierte Administratoren haben volle Berechtigungen für die Bereitstellung auf Konten Ihrer Organisation. Das Verwaltungskonto kann delegierte Administratorberechtigungen nicht auf die Bereitstellung auf bestimmte Vorgänge OUs oder die Ausführung bestimmter StackSet Vorgänge beschränken.

Erstellen Sie eine StackSet mit vom Service verwalteten Berechtigungen (Konsole)

Um eine zu erstellen StackSet

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, von der aus Sie AWS-Region die StackSet verwalten möchten.
3. StackSets.
4. Wählen Sie oben StackSetsauf der Seite die Option Erstellen aus StackSet.

5. Wählen Sie unter Permissions (Berechtigungen) die Option Service-managed permissions (Serviceverwaltete Berechtigungen) aus.

> **Note**
>
> Wenn der vertrauenswürdige Zugriff mit deaktiviert AWS Organizations ist, wird ein Banner angezeigt. Vertrauenswürdiger Zugriff ist erforderlich, um eine StackSet mit vom Dienst verwalteten Berechtigungen zu erstellen oder zu aktualisieren. Nur der Administrator im Verwaltungskonto der Organisation verfügt über Berechtigungen für Aktivieren Sie den vertrauenswürdigen Zugriff für StackSets mit AWS Organizations.

6. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorlage ist bereit aus.
7. Wählen Sie unter Specify template (Vorlage angeben) entweder die URL für den S3-Bucket, der Ihre Stack-Vorlage enthält, aus oder laden Sie eine Stack-Vorlagendatei hoch. Wählen Sie anschließend Weiter.
8. Geben Sie auf der Seite „ StackSet Details angeben“ einen Namen für die ein StackSet, geben Sie alle Parameter an, und klicken Sie dann auf Weiter.
9. Geben Sie auf der Seite StackSet Optionen konfigurieren unter Tags alle Tags an, die auf Ressourcen in Ihrem Stack angewendet werden sollen. Weitere Informationen zur Verwendung von Tags finden Sie im AWSBenutzerhandbuch unter Organisieren und Nachverfolgen von Kosten mithilfe von AWS Kostenzuweisungs-Tags.AWS Fakturierung und Kostenmanagement
10. Wählen Sie für die Ausführungskonfiguration die Option Aktiv aus, um CloudFormation die optimierte Betriebsbehandlung zu aktivieren:

    - Konfliktfreie Operationen werden gleichzeitig ausgeführt, um die Bereitstellungszeiten zu verkürzen.
    - Widersprüchliche Vorgänge werden automatisch in die Warteschlange gestellt und in der Reihenfolge verarbeitet, in der sie angefordert wurden.

    Solange Vorgänge ausgeführt werden oder sich in der Warteschlange befinden, werden alle eingehenden Vorgänge in die CloudFormation Warteschlange gestellt, auch wenn sie nicht miteinander in Konflikt stehen. Während dieser Zeit können Sie die Ausführungseinstellungen nicht ändern.
11. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie unter Capabilities die Option Ich bestätige, dass diese Vorlage IAM-Ressourcen erstellen kann aus, um anzugeben, dass Sie

IAM-Ressourcen in der Vorlage verwenden möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.

12. Wählen Sie Weiter aus, um fortzufahren und den vertrauenswürdigen Zugriff zu aktivieren, falls er noch nicht aktiviert ist.
13. Führen Sie auf der Seite Bereitstellungsoptionen festlegen unter Bereitstellungsziele einen der folgenden Schritte aus:

    - Um die Bereitstellung für alle Konten in Ihrer Organisation durchzuführen, wählen Sie In Organisation bereitstellen aus.
    - Um die Bereitstellung nur für alle Konten vorzunehmen OUs, wählen Sie Für Organisationseinheiten bereitstellen (OUs) aus. Wählen Sie Add an OU (OU hinzufügen) aus und fügen Sie dann die Ziel-OU-ID in das Textfeld ein. Wiederholen Sie den Vorgang für jede neue Ziel-OU.

    Wenn Sie für den Filtertyp Konto die Option Für Organisationseinheiten bereitstellen (OUs) ausgewählt haben, können Sie Ihre Bereitstellungsziele auf bestimmte einzelne Konten festlegen, indem Sie eine der folgenden Optionen wählen und die Kontonummern angeben.

    - Keine (Standard) — Stacks werden für alle Konten in den angegebenen OUs Konten bereitgestellt.
    - Schnittmenge — Stellen Sie Stacks für bestimmte einzelne Konten innerhalb der ausgewählten Konten bereit. OUs
    - Unterschied — Stellen Sie Stacks für alle Konten in den ausgewählten Konten bereit, OUs mit Ausnahme bestimmter Konten.
    - Vereinigung — Stellt Stacks für die angegebenen Konten OUs sowie für weitere individuelle Konten bereit.
14. Wählen Sie unter Automatische Bereitstellung aus, ob die Bereitstellung automatisch für Konten erfolgen soll, die der Zielorganisation hinzugefügt werden, oder für Konten, die der Zielorganisation hinzugefügt werden, oder OUs für future Konten. Weitere Informationen finden Sie unter Automatische Bereitstellungen für StackSets in aktivieren oder deaktivieren AWS Organizations.
15. Wenn Sie die automatische Bereitstellung aktiviert haben, wählen Sie unter Account removal behavior (Kontoentfernungsverhalten) aus, ob Stack-Ressourcen beibehalten oder gelöscht werden, wenn ein Konto aus einer Zielorganisation oder OU entfernt wird.

> **Note**
>
> Wenn Stapel beibehalten ausgewählt ist, werden Stapel aus Ihren entfernt StackSet, aber die Stapel und die zugehörigen Ressourcen bleiben erhalten. Die Ressourcen bleiben in ihrem aktuellen Zustand, sind aber nicht mehr Teil von. StackSet

16. Wählen Sie unter Regionen angeben die Regionen aus, in denen Sie Stacks bereitstellen möchten.
17. Gehen Sie für Bereitstellungsoptionen wie folgt vor:
    - Geben Sie unter Maximale Anzahl gleichzeitiger Konten an, wie viele Konten gleichzeitig verarbeitet werden.
    - Geben Sie für Fehlertoleranz die maximale Anzahl von Kontoausfällen an, die pro Region zulässig sind. Sobald dieses Limit erreicht ist, wird der Vorgang beendet und nicht in andere Regionen übertragen.
    - Wählen Sie für die Regionenparallelität aus, wie Regionen verarbeitet werden sollen: Sequentiell (jeweils eine Region) oder Parallel (mehrere Regionen gleichzeitig).
    - Wählen Sie für den Parallelitätsmodus aus, wie sich Parallelität während der Ausführung des Vorgangs verhält.
      - Strikte Fehlertoleranz — Reduziert den Grad der Parallelität der Konten bei Ausfällen und bleibt innerhalb der Fehlertoleranz +1.
      - Weiche Fehlertoleranz — Behält die angegebene Parallelitätsstufe (den Wert für Maximale Anzahl gleichzeitiger Konten) unabhängig von Ausfällen bei.
18. Wählen Sie Next (Weiter), um fortzufahren.
19. Vergewissern Sie sich auf der Seite „Überprüfen“, dass StackSet Sie die Bereitstellung für die richtigen Konten in den richtigen Regionen durchführen, und wählen Sie dann „Erstellen“. StackSet

    Die StackSet Detailseite wird geöffnet. Sie können den Fortschritt und den Status der Erstellung der Stacks in Ihrem StackSet einsehen.

Erstellen Sie eine StackSet mit vom Service verwalteten Berechtigungen ()AWS CLI

Folgen Sie den Schritten in diesem Abschnitt, um Folgendes AWS CLI zu verwenden:

- Erstellen Sie den StackSet Container.
- Stellen Sie Stack-Instanzen bereit.

> **Note**
>
> Wenn Sie als delegierter Administrator agieren, müssen Sie Folgendes `--call-as DELEGATED_ADMIN` in den Befehl aufnehmen.

Deploy to your organization

Um eine zu erstellen StackSet

1. Verwenden Sie den create-stack-setBefehl, um einen neuen StackSet Namen zu erstellen*my-stackset*. Das folgende Beispiel verwendet eine in einem S3-Bucket gespeicherte Vorlage, ermöglicht automatische Bereitstellungen und bewahrt Stapel, wenn Konten entfernt werden. Weitere Informationen finden Sie unter Automatische Bereitstellungen für StackSets in aktivieren oder deaktivieren AWS Organizations.

```
aws cloudformation create-stack-set \
  --stack-set-name my-stackset \
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/
MyApp.template \
  --permission-model SERVICE_MANAGED \
  --auto-deployment Enabled=true,RetainStacksOnAccountRemoval=true
```

2. Verwenden Sie den list-stack-setsBefehl, um zu bestätigen, dass Ihr erstellt wurde StackSet . Ihr neues StackSet wird in den Ergebnissen aufgeführt.

```
aws cloudformation list-stack-sets
```

- Wenn Sie die `--call-as` Option auf einstellen, `DELEGATED_ADMIN` während Sie in Ihrem Mitgliedskonto angemeldet sind, werden alle StackSets mit vom Dienst verwalteten Berechtigungen im Verwaltungskonto der Organisation list-stack-sets zurückgegeben.
- Wenn Sie die `--call-as` Option auf einstellen, `SELF` während Sie bei Ihrem angemeldet sind AWS-Konto, werden alle selbstverwalteten Daten StackSets in Ihrem list-stack-sets zurückgegeben. AWS-Konto
- Wenn Sie die `--call-as` Option auf einstellen, `SELF` während Sie beim Verwaltungskonto der Organisation angemeldet sind, werden alle Daten StackSets im Verwaltungskonto der Organisation list-stack-sets zurückgegeben.

3. Verwenden Sie den create-stack-instancesBefehl, um Stapel zu Ihrem StackSet hinzuzufügen. Geben Sie für `--deployment-targets` diese Option die Root-ID der Organisation an, die für alle Konten in Ihrer Organisation bereitgestellt werden soll.

   Legen Sie mithilfe der `--operation-preferences` Option Einstellungen für die gleichzeitige Kontoverarbeitung und andere Bereitstellungseinstellungen fest. In diesem Beispiel werden auf der Anzahl basierende Einstellungen verwendet. Beachten Sie, dass dieser `MaxConcurrentCount` Wert `FailureToleranceCount` + 1 nicht überschreiten darf. Verwenden Sie für prozentuale Einstellungen stattdessen `FailureTolerancePercentage` oder`MaxConcurrentPercentage`.

   ```
   aws cloudformation create-stack-instances --stack-set-name my-stackset \
     --deployment-targets OrganizationalUnitIds=r-a1b2c3d4e5 \
     --regions us-west-2 us-east-1 \
     --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
   ```

   Weitere Informationen finden Sie unter CreateStackInstances in der AWS CloudFormation - API-Referenz.

4. Verwenden Sie den folgenden Befehl`operation-id`, um zu überprüfen, ob Ihre Stapel erfolgreich create-stack-instances erstellt wurden, und verwenden Sie den folgenden describe-stack-set-operationBefehl, um zu überprüfen, ob Ihre Stapel erfolgreich erstellt wurden.

   ```
   aws cloudformation describe-stack-set-operation \
     --stack-set-name my-stackset \
     --operation-id operation_ID
   ```

Deploy to organizational units (OUs)

Um eine zu erstellen StackSet

1. Verwenden Sie den create-stack-setBefehl, um einen neuen StackSet Namen zu erstellen*my-stackset*. Das folgende Beispiel verwendet eine in einem S3-Bucket gespeicherte Vorlage und enthält einen Parameter, der a *KeyPairName* mit dem Wert festlegt *TestKey*

```
aws cloudformation create-stack-set \
  --stack-set-name my-stackset \
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/
MyApp.template \
  --permission-model SERVICE_MANAGED \
  --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. Verwenden Sie den list-stack-setsBefehl, um zu bestätigen, dass Ihr erstellt StackSet wurde. Ihr neues StackSet wird in den Ergebnissen aufgeführt.

```
aws cloudformation list-stack-sets
```

   - Wenn Sie die `--call-as` Option auf einstellen, `DELEGATED_ADMIN` während Sie in Ihrem Mitgliedskonto angemeldet sind, werden alle StackSets mit vom Dienst verwalteten Berechtigungen im Verwaltungskonto der Organisation list-stack-sets zurückgegeben.
   - Wenn Sie die `--call-as` Option auf einstellen, `SELF` während Sie bei Ihrem angemeldet sind AWS-Konto, werden alle selbstverwalteten Daten StackSets in Ihrem list-stack-sets zurückgegeben. AWS-Konto
   - Wenn Sie die `--call-as` Option auf einstellen, `SELF` während Sie beim Verwaltungskonto der Organisation angemeldet sind, werden alle Daten StackSets im Verwaltungskonto der Organisation list-stack-sets zurückgegeben.

3. Verwenden Sie den create-stack-instancesBefehl, um Stapel zu Ihrem StackSet hinzuzufügen. Geben Sie für die `--deployment-targets` Option die Organisationseinheit an, in der die Bereitstellung erfolgen IDs soll.

   Legen Sie mithilfe der `--operation-preferences` Option Einstellungen für die gleichzeitige Kontoverarbeitung und andere Bereitstellungseinstellungen fest. In diesem Beispiel werden auf der Anzahl basierende Einstellungen verwendet. Beachten Sie, dass dieser `MaxConcurrentCount` Wert `FailureToleranceCount` + 1

nicht überschreiten darf. Verwenden Sie für prozentuale Einstellungen stattdessen `FailureTolerancePercentage` oder`MaxConcurrentPercentage`.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo,ou-rcuk-slr5lh0a \
  --regions us-west-2 us-east-1 \
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Weitere Informationen finden Sie unter CreateStackInstances in der AWS CloudFormation - API-Referenz.

4. Verwenden Sie den folgenden Befehl`operation-id`, um zu überprüfen, ob Ihre Stapel erfolgreich create-stack-instances erstellt wurden, und verwenden Sie den folgenden describe-stack-set-operationBefehl, um zu überprüfen, ob Ihre Stapel erfolgreich erstellt wurden.

```
aws cloudformation describe-stack-set-operation \
  --stack-set-name my-stackset \
  --operation-id operation_ID
```

Deploy to specific accounts in OUs

Sie können auf bestimmte Organisationseinheiten (OUs) abzielen und mithilfe der Kontofilterung genau steuern, welche Konten Stack-Bereitstellungen erhalten. Standardmäßig werden Stacks für alle Konten innerhalb der angegebenen Konten bereitgestellt, OUs wenn keine Kontofilterung angegeben ist.

In der AWS CLI geben Sie die Kontofilterung mit der `--deployment-targets` Option an. Weitere Informationen finden Sie unter DeploymentTargets.

Nachdem Sie den StackSet Container mit dem create-stack-set Befehl erstellt haben, verwenden Sie eines der folgenden Beispiele, um Stacks für bestimmte Konten bereitzustellen.

Zielt auf bestimmte Konten in einer Organisationseinheit ab

Im folgenden Beispiel werden Stacks nur für die Konten A1 und A2 in bereitgestellt. OU1

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \
  --deployment-targets
 OrganizationalUnitIds=OU1,Accounts=A1,A2,AccountFilterType=INTERSECTION \
```

```
  --regions us-west-2 us-east-1
```

Schließt Konten aus einer Organisationseinheit aus

Im folgenden Beispiel werden Stacks für alle Konten OU1 außer den Konten A1 und A2 bereitgestellt.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \
  --deployment-targets
 OrganizationalUnitIds=OU1,Accounts=A1,A2,AccountFilterType=DIFFERENCE \
  --regions us-west-2 us-east-1
```

Automatische Bereitstellungen für StackSets in aktivieren oder deaktivieren AWS Organizations

CloudFormation kann automatisch zusätzliche Stacks für neue AWS Organizations Konten bereitstellen, wenn diese zu Ihrer Zielorganisation oder Ihren Organisationseinheiten hinzugefügt werden ()OUs. Sie können automatische Bereitstellungen aktivieren und wählen, ob Stacks und die zugehörigen Ressourcen gelöscht oder beibehalten werden sollen, wenn Konten aus dem Ziel entfernt werden. OUs Diese Einstellungen können jederzeit geändert werden, wenn Sie vom Service StackSets verwaltete Berechtigungen verwenden.

Wie funktionieren automatische Bereitstellungen

Wenn automatische Bereitstellungen aktiviert sind, werden sie ausgelöst, wenn Konten zu einer Zielorganisation oder Organisationseinheit hinzugefügt, aus einer Zielorganisation oder Organisationseinheit entfernt oder zwischen Zielen verschoben werden. OUs

Denken Sie beispielsweise an `StackSet1` die Ziele `OU1` in der `us-east-1` Region und `StackSet2` die Ziele `OU2` in der `us-east-1` Region. `OU1`enthält`AccountA`.

Wenn Sie `OU2` bei aktivierter automatischer Bereitstellung `AccountA` von `OU1` zu wechseln, CloudFormation wird automatisch ein Löschvorgang ausgeführt, um den `StackSet1` Stapel zu entfernen, `AccountA` und stellt einen Erstellungsvorgang in die Warteschlange, mit dem der `StackSet2` Stapel hinzugefügt wird. `AccountA`

Überlegungen

Bei der Verwendung automatischer Bereitstellungen sollten Sie Folgendes beachten:

- Die Funktion für automatische Bereitstellungen ist auf der StackSet Ebene aktiviert. Sie können automatische Bereitstellungen nicht selektiv für Konten oder OUs Regionen anpassen.
- Überschriebene Parameterwerte gelten nur für die Konten, die sich derzeit im Ziel OUs befinden, und für ihr untergeordnetes Konto. OUs Konten, die dem Ziel OUs und ihrem Kind OUs in future hinzugefügt werden, verwenden die StackSet Standardwerte und nicht die überschriebenen Werte.
- Wenn Sie auf bestimmte Konten abzielen und automatische Bereitstellungen aktivieren, verwenden Sie StackSet weiterhin den Filter auf Kontoebene, der in der letzten Bereitstellung definiert wurde, während Sie gleichzeitig die Bereitstellung für neu hinzugefügte Konten innerhalb der bereitgestellten Organisation durchführen. Um Bereitstellungen für neu hinzugefügte Konten zu verhindern, deaktivieren Sie automatische Bereitstellungen.

Automatische Bereitstellungen aktivieren oder deaktivieren (Konsole)

Um automatische Bereitstellungen zu aktivieren oder zu deaktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm den aus, den AWS-Region Sie erstellt haben. StackSet
3. StackSets.
4. Wählen Sie auf der StackSetsSeite die Option neben dem Namen der StackSet zu aktualisierenden Datei aus.
5. Wählen Sie im Aktionsmenü in der oberen rechten Ecke die Option Automatische Bereitstellung bearbeiten aus.
6. Gehen Sie in dem sich öffnenden Dialogfeld wie folgt vor:
   a. Wählen Sie für Automatische Bereitstellung die Option Aktiviert oder Deaktiviert.
   b. Wählen Sie für das Verhalten bei der Kontoentfernung die Option Stapel löschen oder Stapel beibehalten aus. Zurückbehaltene Ressourcen behalten ihren aktuellen Status bei, sind aber nicht mehr Teil von. StackSet
7. Wählen Sie Speichern.

Automatische Bereitstellungen aktivieren oder deaktivieren ()AWS CLI

Um automatische Bereitstellungen zu aktivieren oder zu deaktivieren

1. Verwenden Sie den Befehl update-stack-set mit der Option `--auto-deployment`.

   Der folgende Befehl ermöglicht automatische Bereitstellungen.

   ```
   aws cloudformation update-stack-set --stack-set-name my-stackset \
     --use-previous-template --auto-deployment
    Enabled=true,RetainStacksOnAccountRemoval=true
   ```

   Um automatische Bereitstellungen zu deaktivieren, geben Sie `Enabled=false` alternativ den Wert für die `--auto-deployment` Option an, wie im folgenden Beispiel.

   ```
   aws cloudformation update-stack-set --stack-set-name my-stackset \
     --use-previous-template --auto-deployment Enabled=false
   ```

2. Verwenden Sie die Vorgangs-ID, die als Teil der update-stack-set Ausgabe zurückgegeben wurde, um describe-stack-set-operationzu überprüfen, ob Ihre Aktualisierung erfolgreich StackSet war.

   ```
   aws cloudformation describe-stack-set-operation --operation-id operation_ID
   ```

Aktualisiere AWS CloudFormation StackSets

Sie können Ihre Aktualisierung entweder StackSet über die CloudFormation Konsole oder die durchführen AWS CLI.

Informationen zum Hinzufügen und Entfernen von Konten und Regionen zu einer StackSet finden Sie unter Stapel hinzufügen zu StackSets undStapel löschen von StackSets. Informationen zum Überschreiben von Parameterwerten für einen Stack finden Sie unterÜberschreiben Sie Parameter auf Stacks.

Themen

- Aktualisieren Sie Ihre StackSet (Konsole)
- Aktualisieren Sie Ihre StackSet (AWS CLI)

Aktualisieren Sie Ihre StackSet (Konsole)

Um ein zu aktualisieren StackSet

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm den aus, den AWS-Region Sie erstellt haben. StackSet
3. StackSets.
4. Wählen Sie auf der StackSetsSeite die aus, die StackSet Sie aktualisieren möchten.
5. Wenn Sie die StackSet Option ausgewählt haben, wählen Sie im Menü Aktionen die Option StackSet Details bearbeiten aus.
6. Aktualisieren Sie auf der Seite „Vorlage auswählen“ den Abschnitt „Berechtigungen“ nach Bedarf, oder fahren Sie mit dem nächsten Schritt fort.
7. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Aktuelle Vorlage verwenden aus, um die aktuelle Vorlage zu verwenden, oder Aktuelle Vorlage ersetzen, um eine S3-URL zu einer anderen Vorlage anzugeben oder eine neue Vorlage hochzuladen.
8. Wählen Sie Weiter aus.
9. Aktualisieren Sie auf der Seite „ StackSet Details angeben“ bei StackSet Bedarf die Beschreibung für. StackSet
10. Aktualisieren Sie für Parameter die Parameterwerte nach Bedarf.
11. Wählen Sie Weiter aus.
12. Ändern Sie auf der Seite „ StackSet Optionen konfigurieren“ für Tags die Tags nach Bedarf. Sie können Tags hinzufügen, aktualisieren oder löschen. Weitere Informationen zur Verwendung von Stichwörtern finden Sie im AWSAWS Fakturierung und Kostenmanagement Benutzerhandbuch unter Organisieren und Nachverfolgen von AWS Kosten mithilfe von Kostenverteilungs-Tags.
13. Für die Ausführungskonfiguration können Sie die Ausführungskonfiguration nach Bedarf aktualisieren.

> **Note**
>
> Denken Sie daran, dass Sie die Ausführungseinstellungen nicht ändern können, wenn Vorgänge ausgeführt werden oder sich in der Warteschlange befinden.

14. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie unter Funktionen die Option Ich bestätige, dass diese Vorlage möglicherweise IAM-Ressourcen erstellt, um anzugeben, dass Sie IAM-Ressourcen in der Vorlage verwenden möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.
15. Wählen Sie Weiter aus.
16. Geben Sie auf der Seite „Bereitstellungsoptionen festlegen“ die Konten und Regionen für das Update an.

    CloudFormation stellt Stack-Updates für die angegebenen Konten innerhalb der ersten Region bereit und fährt dann mit der nächsten fort usw., solange die Bereitstellungsfehler einer Region eine festgelegte Fehlertoleranz nicht überschreiten.

    a. [Selbstverwaltete Berechtigungen] Wählen Sie für Konten und Bereitstellungsorte die Option Stacks in Konten bereitstellen aus. Fügen Sie das Zielkonto IDs , mit dem Sie Ihr Konto erstellt haben, StackSet in das Textfeld ein und trennen Sie mehrere Zahlen durch Kommas.

       [Vom Dienst verwaltete Berechtigungen] Führen Sie einen der folgenden Schritte aus:

       - Wählen Sie Für Organisationseinheiten bereitstellen (OUs) aus. Geben Sie das Ziel ein OUs , mit dem Sie Ihr erstellt haben StackSet.
       - Wählen Sie Deploy to accounts (Für Konten bereitstellen) aus. Fügen Sie die Ziel-OU IDs oder IDs das Zielkonto ein, mit dem Sie Ihre erstellt haben StackSet.

    b. Geben Sie unter Regionen angeben die Reihenfolge an, in der Sie Ihre Updates bereitstellen CloudFormation möchten.

    c. Gehen Sie für Bereitstellungsoptionen wie folgt vor:

       - Geben Sie unter Maximale Anzahl gleichzeitiger Konten an, wie viele Konten gleichzeitig verarbeitet werden.
       - Geben Sie unter Fehlertoleranz die maximale Anzahl von Kontoausfällen an, die pro Region zulässig sind. Sobald dieses Limit erreicht ist, wird der Vorgang beendet und nicht in andere Regionen übertragen.
       - Wählen Sie für die Regionenparallelität aus, wie Regionen verarbeitet werden sollen: Sequentiell (jeweils eine Region) oder Parallel (mehrere Regionen gleichzeitig).
       - Wählen Sie für den Parallelitätsmodus aus, wie sich Parallelität während der Ausführung des Vorgangs verhält.

- Strikte Fehlertoleranz — Reduziert den Grad der Parallelität der Konten bei Ausfällen und bleibt dabei innerhalb der Fehlertoleranz +1.
- Weiche Fehlertoleranz — Behält die angegebene Parallelitätsstufe (den Wert für Maximale Anzahl gleichzeitiger Konten) unabhängig von Ausfällen bei.

d. Wählen Sie Next (Weiter), um fortzufahren.

17. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie im entsprechenden Bereich die Option Bearbeiten aus.
18. Wenn Sie bereit sind, fortzufahren, wählen Sie Senden aus.

    CloudFormation beginnt mit der Anwendung Ihrer Aktualisierungen auf Ihrem StackSet und zeigt den Tab „Vorgänge“ auf der StackSet Detailseite an. Sie können Fortschritt und Status der Aktualisierungsoperationen auf der Registerkarte Operations (Operationen) anzeigen.

Aktualisieren Sie Ihre StackSet (AWS CLI)

> **Note**
>
> Wenn Sie als delegierter Administrator agieren, müssen Sie Folgendes `--call-as DELEGATED_ADMIN` in den Befehl aufnehmen.

1. Um ein zu aktualisieren StackSet

   Verwenden Sie den update-stack-setBefehl, um Änderungen an Ihrem vorzunehmen StackSet.

   In den folgenden Beispielen aktualisieren wir die `--parameters` Option StackSet by using. Insbesondere ändern wir die standardmäßige Snapshot-Bereitstellungshäufigkeit für die Konfiguration des Bereitstellungskanals von `TwentyFour_Hours` bis`Twelve_Hours`. Da wir immer noch die aktuelle Vorlage verwenden, fügen wir die `--use-previous-template` Option hinzu.

   Legen Sie mithilfe der `--operation-preferences` Option die gleichzeitige Kontoverarbeitung und andere Bereitstellungseinstellungen fest. In diesen Beispielen werden Einstellungen verwendet, die auf der Anzahl basieren. Beachten Sie, dass dieser `MaxConcurrentCount` Wert `FailureToleranceCount` +1 nicht überschreiten darf. Verwenden Sie für prozentuale Einstellungen stattdessen `FailureTolerancePercentage` oder`MaxConcurrentPercentage`.

[Selbstverwaltete Berechtigungen] Geben IDs Sie für `--accounts` diese Option das Konto an, auf das Ihr Update abzielen soll.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \
  --use-previous-template \
  --parameters ParameterKey=MaximumExecutionFrequency,ParameterValue=Twelve_Hours \
  --accounts account_ID_1 account_ID_2 \
  --regions us-west-2 us-east-1 \
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

[Vom Dienst verwaltete Berechtigungen] Geben IDs Sie für `--deployment-targets` diese Option die Stamm-ID oder Organisationseinheit (OU) der Organisation an, auf die sich Ihr Update beziehen soll.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \
  --use-previous-template \
  --parameters ParameterKey=MaximumExecutionFrequency,ParameterValue=Twelve_Hours \
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo,ou-rcuk-slr5lh0a \
  --regions us-west-2 us-east-1 \
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Weitere Informationen finden Sie unter UpdateStackSet in der AWS CloudFormation -API-Referenz.

2. Stellen Sie sicher, dass Ihr Update erfolgreich StackSet war, indem Sie den describe-stack-set-operation Befehl ausführen, um den Status und die Ergebnisse Ihres Aktualisierungsvorgangs anzuzeigen. Verwenden Sie als `--operation-id` die Operations-ID, die durch den Befehl update-stack-set zurückgegeben wurde.

```
aws cloudformation describe-stack-set-operation \
  --operation-id operation_ID
```

Stapel hinzufügen zu CloudFormation StackSets

Wenn Sie eine erstellen StackSet, können Sie die Stapel dafür erstellen. StackSet CloudFormation ermöglicht es Ihnen auch, jederzeit nach der Erstellung weitere Stacks für zusätzliche Konten und Regionen hinzuzufügen. StackSet Sie können Stacks entweder mit der CloudFormation Konsole oder mit dem hinzufügen. AWS CLI

Themen

- Stapel zu einer StackSet (Konsole) hinzufügen
- Fügen Sie Stacks zu einem StackSet () hinzu AWS CLI

Stapel zu einer StackSet (Konsole) hinzufügen

Um Stapel zu einem hinzuzufügen StackSet

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm den aus, den AWS-Region Sie erstellt haben. StackSet
3. StackSets. Wählen Sie auf der StackSets Seite die aus StackSet , die Sie erstellt haben.
4. Wenn die StackSet Option ausgewählt ist, wählen Sie im Menü Aktionen die Option Stapel hinzufügen zu StackSet.
5. Gehen Sie auf der Seite „Bereitstellungsoptionen festlegen“ wie folgt vor:
   a. Wählen Sie für Stacks hinzufügen zu die StackSet Option Neue Stacks bereitstellen aus.
   b. Gehen Sie als Nächstes je nach Ihrer StackSet Berechtigungskonfiguration wie folgt vor:
      - [Selbstverwaltete Berechtigungen] Wählen Sie für Konten, Bereitstellungsorte die Option Stapel in Konten bereitstellen aus. Fügen Sie Ihre Zielkonto-Zahlen in das Textfeld ein und trennen Sie dabei die einzelnen Zahlen durch Kommas.
      - [Vom Dienst verwaltete Berechtigungen] Führen Sie für Bereitstellungsziele einen der folgenden Schritte aus:
        - Wählen Sie Deploy to organization (In Organisation bereitstellen) aus, um für alle Konten in Ihrer Organisation bereitzustellen.
        - Wählen Sie „Für Organisationseinheiten bereitstellen“ (OUs), um die Bereitstellung für alle bestimmten OUs Konten vorzunehmen. Wählen Sie Add another OU (Weitere OU hinzufügen) aus und fügen Sie dann die Ziel-OU-ID in das Textfeld ein. Wiederholen Sie den Vorgang für jede neue Ziel-OU. CloudFormation zielt auch auf jedes Kind OUs Ihrer ausgewählten Ziele ab.

> **Note**
>
> Wenn Sie eine Organisationseinheit hinzufügen, auf die Sie StackSet bereits abzielen, werden neue Stacks in allen Konten in der Organisationseinheit CloudFormation erstellt, für die noch keine Stacks von Ihnen vorhanden sind StackSet (z. B. Konten, die der Organisationseinheit hinzugefügt wurden, nachdem Ihre Organisationseinheit erstellt StackSet wurde und bei denen automatische Bereitstellungen deaktiviert wurden).

c. Geben Sie unter Regionen angeben in den Zielkonten, AWS-Regionen die Sie im vorherigen Schritt angegeben haben, an, für welche Regionen die Bereitstellung erfolgen soll. Standardmäßig CloudFormation werden Stacks in den angegebenen Konten innerhalb der ersten Region bereitgestellt und dann mit der nächsten Region fortgefahren usw., sofern die Bereitstellungsfehler einer Region eine angegebene Fehlertoleranz nicht überschreiten.

d. Gehen Sie für Bereitstellungsoptionen wie folgt vor:

- Geben Sie unter Maximale Anzahl gleichzeitiger Konten an, wie viele Konten gleichzeitig verarbeitet werden.
- Geben Sie für Fehlertoleranz die maximale Anzahl von Kontoausfällen an, die pro Region zulässig sind. Sobald dieses Limit erreicht ist, wird der Vorgang beendet und nicht in andere Regionen übertragen.
- Wählen Sie für die Regionenparallelität aus, wie Regionen verarbeitet werden sollen: Sequentiell (jeweils eine Region) oder Parallel (mehrere Regionen gleichzeitig).
- Wählen Sie für den Parallelitätsmodus aus, wie sich Parallelität während der Ausführung des Vorgangs verhält.
  - Strikte Fehlertoleranz — Reduziert den Grad der Parallelität der Konten bei Ausfällen und bleibt innerhalb der Fehlertoleranz +1.
  - Weiche Fehlertoleranz — Behält die angegebene Parallelitätsstufe (den Wert für Maximale Anzahl gleichzeitiger Konten) unabhängig von Ausfällen bei.

e. Wählen Sie Weiter aus.

6. Lassen Sie auf der Seite Specify Overrides (Außerkraftsetzungen angeben) die Eigenschaftswerte unverändert. Für die zu erstellenden Stacks werden keine Eigenschaftswerte überschrieben. Wählen Sie Weiter aus.

7. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie im entsprechenden Abschnitt Bearbeiten aus.
8. Wenn Sie bereit sind, fortzufahren, wählen Sie Senden.

   CloudFormation beginnt mit der Erstellung Ihrer Stacks. Sehen Sie sich den Fortschritt und den Status der Erstellung der Stacks auf der StackSet Detailseite an, die geöffnet wird, wenn Sie Absenden wählen. StackSet Wenn Sie fertig sind, sollten Ihre neuen Stacks auf der Registerkarte Stack-Instances aufgeführt werden.

Fügen Sie Stacks zu einem StackSet () hinzu AWS CLI

> **Note**
>
> Wenn Sie als delegierter Administrator agieren, müssen Sie Folgendes `--call-as DELEGATED_ADMIN` in den Befehl aufnehmen.

Um Stacks zu einem StackSet mit selbstverwalteten Berechtigungen hinzuzufügen

Verwenden Sie den create-stack-instances-CLI-Befehl. Geben Sie für die `--accounts` Option die Konten an, IDs für die Sie Stacks erstellen möchten.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \
  --accounts account_id --regions eu-west-1 us-west-2
```

Um Stacks zu einem StackSet mit vom Service verwalteten Berechtigungen hinzuzufügen

Verwenden Sie den create-stack-instances-CLI-Befehl. Geben Sie für `--deployment-targets` diese Option die Organisations- (Root-) ID oder OU an, IDs für die Sie Stacks erstellen möchten. Befehle, die auf bestimmte Konten abzielen, finden Sie beispielsweise unterErstellen Sie eine StackSet mit vom Service verwalteten Berechtigungen ()AWS CLI.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \
  --deployment-targets OrganizationalUnitIds=ou-rcuk-r1qi0wl7 --regions eu-west-1 us-
west-2
```

> **Note**
>
> Wenn Sie eine Organisationseinheit hinzufügen, auf die Sie StackSet bereits abzielen, werden neue Stacks in allen Konten in der Organisationseinheit CloudFormation erstellt, die noch keine Stacks von Ihnen haben StackSet (z. B. Konten, die der Organisationseinheit hinzugefügt wurden, nachdem Ihre Organisationseinheit erstellt StackSet wurde und automatische Bereitstellungen deaktiviert sind).

Überschreiben Sie Parameterwerte auf Stacks in Ihrem CloudFormation StackSet

In bestimmten Fällen möchten Sie vielleicht, dass Stapel in bestimmten Regionen oder Konten andere Eigenschaftswerte haben als die, die in den StackSet Stacks selbst angegeben sind. So möchten Sie beispielsweise abhängig davon, ob ein Konto für die Entwicklung oder die Produktion verwendet wird, für einen bestimmten Parameter einen anderen Wert angeben. In diesen Situationen CloudFormation können Sie damit Parameterwerte in Stacks nach Konto und Region überschreiben. Sie können Vorlagenparameterwerte überschreiben, wenn Sie die Stapel zum ersten Mal erstellen, und Sie können Parameterwerte für bestehende Stapel überschreiben. Sie können nur Parameter, die Sie zuvor in Stacks überschrieben haben, wieder auf die Werte setzen, die in der angegeben sind. StackSet

Überschreibungen von Parameterwerten gelten für Stacks in den von Ihnen ausgewählten Konten und Regionen. Bei StackSet Aktualisierungen werden alle Parameterwerte, die für einen Stack außer Kraft gesetzt wurden, nicht aktualisiert, sondern behalten ihren überschriebenen Wert.

Sie können nur Parameterwerte überschreiben StackSet, die im angegeben sind. Um einen Parameter selbst hinzuzufügen oder zu löschen, müssen Sie die Vorlage aktualisieren. StackSet Wenn Sie einer StackSet Vorlage einen Parameter hinzufügen, müssen Sie, bevor Sie diesen Parameterwert in einem Stapel überschreiben können, zunächst alle Stapel mit dem neuen Parameter und Wert aktualisieren, der in der StackSet angegeben ist. Sobald alle Stapel mit dem neuen Parameter aktualisiert wurden, können Sie den Parameterwert in einzelnen Stacks nach Bedarf überschreiben.

Informationen zum Überschreiben von StackSet Parameterwerten beim Erstellen von Stacks finden Sie unter. Stapel hinzufügen zu StackSets

Themen

- Parameter auf Stacks überschreiben (Konsole)
- Parameter auf Stacks überschreiben ()AWS CLI

Parameter auf Stacks überschreiben (Konsole)

Um Parameter für bestimmte Stacks zu überschreiben

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm den aus, den AWS-Region Sie erstellt haben. StackSet
3. StackSets. Wählen Sie auf der StackSets Seite Ihre aus StackSet.
4. Wenn Sie die StackSet Option ausgewählt haben, wählen Sie im Menü Aktionen die Option StackSetParameter überschreiben aus.
5. Geben Sie auf der Seite „Bereitstellungsoptionen festlegen“ die Konten und Regionen für die Stacks an, für die Sie Überschreibungen erstellen möchten.

   Standardmäßig CloudFormation werden Stacks in den angegebenen Konten innerhalb der ersten Region bereitgestellt und dann mit der nächsten Region fortgefahren usw., vorausgesetzt, dass die Bereitstellungsfehler einer Region eine angegebene Fehlertoleranz nicht überschreiten.

   a. [Selbstverwaltete Berechtigungen] Wählen Sie für Bereitstellungsorte die Option Stacks in Konten bereitstellen aus. Fügen Sie einige oder alle Zielkonten ein IDs, mit denen Sie Ihr Konto erstellt haben. StackSet

      [Vom Dienst verwaltete Berechtigungen] Führen Sie einen der folgenden Schritte aus:

      - Wählen Sie Für Organisationseinheiten bereitstellen (OUs) aus. Geben Sie eines oder mehrere Ziele ein, mit OUs denen Sie Ihr Ziel erstellt haben StackSet. Die überschriebenen Parameterwerte gelten nur für die Konten, die sich derzeit im Ziel befinden, OUs und für ihre untergeordneten Konten. OUs Konten, die dem Ziel OUs und ihrem Kind OUs in future hinzugefügt werden, verwenden die StackSet Standardwerte und nicht die überschriebenen Werte.
      - Wählen Sie Deploy to accounts (Für Konten bereitstellen) aus. Fügen Sie einige oder alle Ziel-OU IDs oder das Zielkonto ein IDs , mit dem Sie Ihre erstellt haben. StackSet

b. Fügen Sie unter Regionen angeben eine oder mehrere der Regionen hinzu, in denen Sie zu diesem StackSet Zweck Stacks bereitgestellt haben.

   Wenn Sie mehrere Regionen hinzufügen, bestimmt die Reihenfolge der Regionen unter Specify regions (Regionen angeben) die Reihenfolge ihrer Bereitstellung.

c. Gehen Sie für Bereitstellungsoptionen wie folgt vor:

   - Geben Sie unter Maximale Anzahl gleichzeitiger Konten an, wie viele Konten gleichzeitig verarbeitet werden.
   - Geben Sie unter Fehlertoleranz die maximale Anzahl von Kontoausfällen an, die pro Region zulässig sind. Sobald dieses Limit erreicht ist, wird der Vorgang beendet und nicht in andere Regionen übertragen.
   - Wählen Sie für die Regionenparallelität aus, wie Regionen verarbeitet werden sollen: Sequentiell (jeweils eine Region) oder Parallel (mehrere Regionen gleichzeitig).
   - Wählen Sie für den Parallelitätsmodus aus, wie sich Parallelität während der Ausführung des Vorgangs verhält.
     - Strikte Fehlertoleranz — Reduziert den Grad der Parallelität der Konten bei Ausfällen und bleibt dabei innerhalb der Fehlertoleranz +1.
     - Weiche Fehlertoleranz — Behält die angegebene Parallelitätsstufe (den Wert für Maximale Anzahl gleichzeitiger Konten) unabhängig von Ausfällen bei.

d. Wählen Sie Weiter aus.

6. Aktivieren Sie auf der Seite „Überschreibungen angeben“ die Kontrollkästchen für die zu überschreibenden Parameter und wählen Sie dann im Menü Überschreibungswert bearbeiten die Option „ StackSetWert überschreiben“.
7. Nehmen Sie auf der Seite StackSet „Parameterwerte überschreiben“ Ihre Änderungen vor und wählen Sie dann Änderungen speichern.

   > **Note**
   >
   > Um für alle überschriebenen Parameter wieder den in der angegebenen Wert zu verwenden StackSet, markieren Sie alle Parameter und wählen Sie im Menü StackSetÜberschreibungswert bearbeiten die Option Auf Wert setzen. Dadurch werden alle überschriebenen Werte entfernt, sobald Sie die Stapel aktualisiert haben.

8. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie im entsprechenden Abschnitt die Option Bearbeiten aus.

9. Wenn Sie bereit sind, fortzufahren, wählen Sie Senden aus.

   CloudFormation beginnt mit der Aktualisierung Ihrer Stacks. Auf der StackSet Detailseite, die geöffnet wird, wenn Sie „Senden“ wählen, können Sie den Fortschritt und den Status der Stacks einsehen.

Parameter auf Stacks überschreiben ()AWS CLI

> **Note**
>
> Wenn Sie als delegierter Administrator agieren, müssen Sie Folgendes `--call-as DELEGATED_ADMIN` in den Befehl aufnehmen.

Um Parameter für bestimmte Stacks zu überschreiben

1. Verwenden Sie den update-stack-instances AWS CLI Befehl und geben Sie die `--parameter-overrides` Option an.

   [Selbstverwaltete Berechtigungen] Geben Sie für die `--accounts` Option das Konto an, IDs für das Sie Parameterwerte in Stacks überschreiben möchten.

   ```
   aws cloudformation update-stack-instances --stack-set-name my-stackset \
     --parameter-overrides ParameterKey=Subnets,ParameterValue=subnet-1baa3351\
   \,subnet-27b86940 \
     --accounts account_id --regions us-east-1
   ```

   [Vom Dienst verwaltete Berechtigungen] Geben Sie für die `--deployment-targets` Option die Root-ID, OU oder das AWS Organizations Konto der Organisation an IDs, IDs für das Sie Parameter in Stacks überschreiben möchten. In diesem Beispiel überschreiben wir die Parameterwerte für Stacks in allen Konten in der Organisationseinheit mit der ID. *ou-rcuk-1x5j1lwo*

   Die überschriebenen Parameterwerte gelten nur für die Konten, die sich derzeit in der Ziel-OU und ihrer untergeordneten Organisationseinheit befinden. OUs Konten, die der Ziel-OU und ihrer untergeordneten OUs Organisation in future hinzugefügt werden, verwenden die StackSet Standardwerte und nicht die überschriebenen Werte.

```
aws cloudformation update-stack-instances --stack-set-name my-stackset \
  --parameter-overrides ParameterKey=Subnets,ParameterValue=subnet-1baa3351\
\,subnet-27b86940 \
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo \
  --regions us-east-1
```

2. Stellen Sie sicher, dass Ihre Parameterwerte in Stacks erfolgreich überschrieben wurden, indem Sie den describe-stack-set-operation Befehl ausführen, um den Status und die Ergebnisse Ihres Aktualisierungsvorgangs anzuzeigen. Verwenden Sie als `--operation-id` die Operations-ID, die durch den Befehl update-stack-instances zurückgegeben wurde.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Stapel löschen von AWS CloudFormation StackSets

Sie können Stapel entweder StackSets über die CloudFormation Konsole oder über das löschen. AWS CLI

Themen

- Löschen Sie Stacks von Ihrer StackSet (Konsole)
- Lösche Stacks von deinem StackSet ()AWS CLI

> **Note**
>
> Wenn Sie Stacks aus einer Organisationseinheit (OU) der obersten Ebene löschen, wird diese Organisationseinheit als Ziel entfernt. StackSet

Löschen Sie Stacks von Ihrer StackSet (Konsole)

So löschen Sie Stacks

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm den aus, den AWS-Region Sie erstellt haben. StackSet

3. StackSets. Wählen Sie auf der StackSets Seite die StackSet.
4. Wenn Sie StackSet ausgewählt sind, wählen Sie im Menü Aktionen die Option Stapel löschen StackSet aus.
5. Wählen Sie auf der Seite „Bereitstellungsoptionen festlegen“ zunächst die Konten und Regionen aus, in denen Sie die Stacks löschen möchten.

   a. [Selbstverwaltete Berechtigungen] Wählen Sie für Konten die Option Stacks in Konten bereitstellen oder Stacks in Organisationseinheiten bereitstellen aus.

      Wenn Sie Stacks in Konten bereitstellen auswählen, fügen Sie Ihre Zielkontonummern in das Textfeld Kontonummern ein und trennen Sie mehrere Zahlen durch Kommas.

      Wenn Sie Stacks in Organisationseinheiten bereitstellen auswählen, fügen Sie eine Ziel-OU-ID in das Textfeld Organisationsnummern ein, um alle Konten als Ziel festzulegen, die Teil der angegebenen Organisation sind.

   b. [Vom Dienst verwaltete Berechtigungen] Geben Sie für Organisationseinheiten (OUs) die Ziel-OU an. IDs

      > **Important**
      >
      > CloudFormation löscht Stapel sowohl vom angegebenen Ziel als auch vom OUs untergeordneten Ziel. OUs

      Für den Filtertyp Konto können Sie verfeinern, für welche Konten Stapel gelöscht werden sollen, indem Sie eine der folgenden Optionen wählen und die Kontonummern angeben.

      - Keine (Standard) — Löscht Stacks von allen Konten in den angegebenen Konten. OUs
      - Schnittmenge — Löscht nur Stapel von bestimmten einzelnen Konten innerhalb der ausgewählten Konten. OUs
      - Unterschied — Löscht Stacks von allen Konten in den ausgewählten Konten OUs mit Ausnahme bestimmter Konten.
      - Vereinigung — Löscht Stapel aus den angegebenen Konten OUs sowie aus weiteren Einzelkonten.

   c. Wählen Sie unter Regionen angeben die Regionen aus, aus denen Sie Stacks innerhalb der Zielkonten löschen möchten.
6. Gehen Sie für Bereitstellungsoptionen wie folgt vor:

- Geben Sie unter Maximale Anzahl gleichzeitiger Konten an, wie viele Konten gleichzeitig verarbeitet werden.
- Geben Sie unter Fehlertoleranz die maximale Anzahl von Kontoausfällen an, die pro Region zulässig sind. Sobald dieses Limit erreicht ist, wird der Vorgang beendet und nicht in andere Regionen übertragen.
- Aktivieren Sie für Stapel behalten diese Option, um die Stapel und die zugehörigen Ressourcen zu speichern, wenn Sie sie aus Ihren entfernen. StackSet Die Ressourcen bleiben in ihrem aktuellen Zustand, sind aber nicht mehr Teil von. StackSet
- Wählen Sie für Regionsparallelität aus, wie Regionen verarbeitet werden sollen: Sequentiell (jeweils eine Region) oder Parallel (mehrere Regionen gleichzeitig).
- Wählen Sie für den Parallelitätsmodus aus, wie sich Parallelität während der Ausführung des Vorgangs verhält.
  - Strikte Fehlertoleranz — Reduziert den Grad der Parallelität der Konten bei Ausfällen und bleibt innerhalb der Fehlertoleranz +1.
  - Weiche Fehlertoleranz — Behält die angegebene Parallelitätsstufe (den Wert für Maximale Anzahl gleichzeitiger Konten) unabhängig von Ausfällen bei.

7. Wählen Sie Weiter aus.
8. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie im entsprechenden Abschnitt Bearbeiten aus.
9. Wenn Sie bereit sind, die Stapel aus Ihrem zu entfernen StackSet, wählen Sie Senden.

   Nachdem das Löschen des Stacks abgeschlossen ist, können Sie auf der StackSet Detailseite auf der Registerkarte Stack-Instances überprüfen, ob die Stacks von Ihrem StackSet gelöscht wurden.

Lösche Stacks von deinem StackSet ()AWS CLI

> **Note**
>
> Wenn Sie als delegierter Administrator agieren, müssen Sie Folgendes `--call-as DELEGATED_ADMIN` in den Befehl aufnehmen.

Verwenden Sie den delete-stack-instances Befehl mit Ihrem StackSet Namen.

In diesen Beispielen verwenden wir die `--no-retain-stacks` Option, weil wir keine Stapel behalten. Verwenden Sie `--retain-stacks` statt`--no-retain-stacks`, wenn Sie die Stapel und ihre Ressourcen behalten möchten.

Geben Sie für an`--regions`, aus welchen Stapel AWS-Regionen Sie löschen möchten, z. B. und`us-west-2`. `us-east-1`

Legen Sie mithilfe der Option die gleichzeitige Kontoverarbeitung und andere Einstellungen fest. `--operation-preferences` In diesen Beispielen werden Einstellungen verwendet, die auf der Anzahl basieren. Beachten Sie, dass dieser `MaxConcurrentCount` Wert `FailureToleranceCount` +1 nicht überschreiten darf. Verwenden Sie für prozentuale Einstellungen stattdessen `FailureTolerancePercentage` oder`MaxConcurrentPercentage`.

Um Stapel zu löschen (selbstverwaltete Berechtigungen)

Geben Sie für die `--accounts` Option das Konto an, IDs aus dem Stapel gelöscht werden sollen.

```
aws cloudformation delete-stack-instances --stack-set-name my-stackset \
  --accounts account_ID_1 account_ID_2 \
  --regions us-west-2 us-east-1 \
  --no-retain-stacks \
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Um Stacks zu löschen (vom Service verwaltete Berechtigungen)

Geben Sie für `--deployment-targets` die Root-ID der Organisation oder die Organisationseinheit (OU) IDs an, aus der Stapel gelöscht werden sollen.

> **Important**
>
> CloudFormation löscht Stapel sowohl vom angegebenen Ziel als auch vom OUs untergeordneten Ziel. OUs

```
aws cloudformation delete-stack-instances --stack-set-name my-stackset \
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5jlwo,ou-rcuk-slr5lh0a \
  --regions us-west-2 us-east-1 \
  --no-retain-stacks \
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Weitere Informationen finden Sie unter DeleteStackInstances in der AWS CloudFormation -API-Referenz.

Optional können Sie nach Abschluss des Stack-Löschvorgangs überprüfen, ob Stapel aus Ihrem gelöscht wurden, StackSet indem Sie den describe-stack-set-operation Befehl ausführen, der den Status und die Ergebnisse des Löschvorgangs für Stapel anzeigt. Verwenden Sie als `--operation-id` die Operations-ID, die durch den Befehl delete-stack-instances zurückgegeben wurde.

```
aws cloudformation describe-stack-set-operation --stack-set-name my-stackset \
  --operation-id ddf16f54-ad62-4d9b-b0ab-3ed8e9example
```

Löschen AWS CloudFormation StackSets

Um eine zu löschen StackSet, müssen Sie zuerst alle Stapel in der StackSet löschen. Informationen zum Löschen aller Stapel finden Sie unter. Stapel löschen von StackSets

Themen

- Löschen Sie eine StackSet (Konsole)
- Löschen Sie ein StackSet (AWS CLI)
- Löschen von Servicerollen (optional)

Löschen Sie eine StackSet (Konsole)

Um ein zu löschen StackSet

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm den aus, den AWS-Region Sie erstellt haben. StackSet
3. Wählen Sie auf der StackSetsSeite die StackSet.
4. Wenn das StackSet ausgewählt ist, wählen Sie im Menü Aktionen die Option Löschen StackSet aus.
5. Wenn Sie aufgefordert werden, zu bestätigen, dass Sie das löschen möchten StackSet, wählen Sie Löschen StackSet.

Löschen Sie ein StackSet (AWS CLI)

> **Note**
>
> Wenn Sie als delegierter Administrator agieren, müssen Sie Folgendes `--call-as DELEGATED_ADMIN` in den Befehl aufnehmen.

Um ein zu löschen StackSet

1. Verwenden Sie den folgenden delete-stack-set-Befehl. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **y** ein und drücken Sie anschließend Eingabe.

   ```
   aws cloudformation delete-stack-set --stack-set-name my-stackset
   ```

2. Stellen Sie sicher, dass das gelöscht StackSet wurde, indem list-stack-sets Sie den Befehl ausführen. Die Ergebnisse des list-stack-sets Befehls sollten Ihren Stack mit dem Status anzeigen`DELETED`.

   ```
   aws cloudformation list-stack-sets
   ```

Löschen von Servicerollen (optional)

Wenn Sie die IAM-Servicerollen, die für die Ausführung von StackSet Vorgängen CloudFormation erforderlich sind, nicht mehr benötigen, empfehlen wir Ihnen, die Rollen zu löschen.

Bei selbstverwalteten Rollen die Rollen StackSets, die Sie erstellt haben. Weitere Informationen zu diesen Rollen finden Sie unterSelbstverwaltete Berechtigungen erteilen.

Für vom Dienst verwaltete Rollen StackSets haben die Rollen StackSets, für die automatisch erstellt wurden, das Suffix `CloudFormationStackSetsOrgAdmin` im Organisationsverwaltungskonto und `CloudFormationStackSetsOrgMember` in jedem Zielkonto. Weitere Informationen finden Sie unter Service-verknüpfte Rollen.

Um eine Servicerolle zu löschen (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, und füllen Sie dann das Kontrollkästchen neben der Rolle aus, die Sie löschen möchten.
3. Wählen Sie im Menü Role actions oben auf der Seite Delete role.
4. Wählen Sie im Bestätigungsdialogfeld Yes, Delete aus. Wenn Sie sich sicher sind, können Sie mit dem Löschen fortfahren, auch wenn das letzte Service-Zugriffsdatum noch geladen wird.

Um eine Servicerolle zu löschen (AWS CLI)

- Verwenden Sie den folgenden delete-role-Befehl. Geben Sie ein, wenn Sie zur Bestätigung aufgefordert werden**y**, und drücken Sie dannEnter.

```
aws iam delete-role --role-name role name
```

Weitere Informationen zum Löschen von Rollen finden Sie unter Löschen von Rollen oder Instanzprofilen im IAM-Benutzerhandbuch.

Vermeiden Sie fehlgeschlagene StackSets Bereitstellungen mithilfe von Target Account Gates

Ein Account-Gate ist eine optionale Funktion, mit der Sie überprüfen können, ob ein Zielkonto bestimmte Anforderungen erfüllt, bevor der StackSets Betrieb CloudFormation in diesem Konto aufgenommen wird. Diese Überprüfung erfolgt über eine AWS Lambda Funktion, die als Überprüfung der Voraussetzungen dient.

Ein gängiges Beispiel für ein Account-Gate ist die Überprüfung, ob auf dem Zielkonto keine CloudWatch Alarme aktiv oder ungelöst sind. CloudFormation ruft die Lambda-Funktion jedes Mal auf, wenn Sie Stack-Operationen im Zielkonto starten, und fährt nur fort, wenn die Funktion einen `SUCCEEDED` Code zurückgibt. Wenn die Lambda-Funktion den Status zurückgibt`FAILED`, setzt Sie den angeforderten Vorgang CloudFormation nicht fort. Wenn Sie keine Lambda-Funktion für das Account-Gating konfiguriert haben, CloudFormation überspringen Sie die Prüfung und setzen Sie Ihren Vorgang fort.

Wenn die Konto-Gate-Überprüfung des Zielkontos fehlschlägt, wird der fehlgeschlagene Vorgang Ihrer angegebene Ausfalltoleranz bzw. dem Prozentsatz für Stacks angerechnet. Weitere Informationen zur Fehlertoleranz finden Sie unter StackSet Operationsoptionen.

Account-Gating ist nur für Operationen verfügbar. StackSets Diese Funktion ist für andere CloudFormation Operationen außerhalb von StackSets nicht verfügbar.

Voraussetzungen

Die folgenden Anforderungen müssen für das Account-Gating erfüllt sein:

- Ihre Lambda-Funktion muss benannt werden`AWSCloudFormationStackSetAccountGate`, um diese Funktion verwenden zu können.
- Sie AWSCloudFormationStackSetExecutionRolemüssen über Berechtigungen verfügen, um Ihre Lambda-Funktion aufzurufen. Ohne diese Berechtigungen CloudFormation wird die Überprüfung der Kontozuweisung übersprungen und die Stack-Operationen ausgeführt.
- Damit Konto-Gating funktioniert, muss Zielkonten die Lambda-Berechtigung `InvokeFunction` hinzugefügt werden. Die Vertrauensbeziehungs-Richtlinie des Zielkontos muss über eine Vertrauensbeziehung mit dem Administratorkonto verfügen. Es folgt ein Beispiel für eine Richtlinienanweisung, die die Lambda-Berechtigungen `InvokeFunction` erteilt.

  JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudFormation Vorlagen für die Erstellung von Lambda-Funktionen

Verwenden Sie die folgenden Beispielvorlagen, um `AWSCloudFormationStackSetAccountGate` Lambda-Funktionen zu erstellen. Informationen zum Erstellen eines neuen Stacks mit einer dieser Vorlagen finden Sie unterEinen Stack von der CloudFormation Konsole aus erstellen.

Vorlagenspeicherort	Beschreibung
https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AccountGateSucceeded.yml	Erstellt einen Stack, der eine Lambda-Konto-Gate-Funktion implementiert, die den Status `SUCCEEDED` zurückgibt.
https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AccountGateFailed.yml	Erstellt einen Stack, der eine Lambda-Konto-Gate-Funktion implementiert, die den Status `FAILED` zurückgibt.

Wählen Sie den Parallelitätsmodus für AWS CloudFormation StackSets

Der Parallelitätsmodus ist ein Parameter, mit dem Sie auswählen können StackSetOperationPreferences, wie sich die Parallelitätsebene während des Betriebs verhält. StackSet Dabei haben Sie die Wahl zwischen den folgenden Modi:

- Strikte Fehlertoleranz: Diese Option senkt den Umfang der Parallelität dynamisch, damit die Anzahl der fehlerhaften Konten niemals den Wert der Fehlertoleranz +1 überschreitet. Die anfängliche tatsächliche Parallelität wird auf den Wert der Option Maximale Anzahl gleichzeitiger Konten oder auf den Wert der Fehlertoleranz +1 festgelegt (je nachdem, welcher niedriger ist). Die tatsächliche Parallelität wird dann proportional zur Anzahl der Fehler reduziert. Dies ist das Standardverhalten.
- Lose Fehlertoleranz: Diese Option entkoppelt die Fehlertoleranz von der tatsächlichen Parallelität. Dadurch können StackSet Operationen unabhängig von der Anzahl der Fehler auf der Parallelitätsebene ausgeführt werden, die durch den Wert „Maximale Anzahl gleichzeitiger Konten“ festgelegt ist.

Strict Failure Tolerance senkt die Bereitstellungsgeschwindigkeit, wenn StackSet Betriebsfehler auftreten, da die Parallelität mit jedem Fehler abnimmt. Soft Failure Tolerance priorisiert die Geschwindigkeit der Bereitstellung und nutzt CloudFormation gleichzeitig die Sicherheitsfunktionen. Auf diese Weise können Sie StackSet Betriebsfehler im Hinblick auf häufig auftretende Probleme überprüfen und beheben, z. B. Probleme im Zusammenhang mit vorhandenen Ressourcen, Dienstkontingenten und Berechtigungen.

Weitere Informationen zu Fehlern bei StackSets Stack-Vorgängen finden Sie unterHäufige Gründe für Störungen der Stack-Operationen.

Weitere Informationen zu den Optionen Maximale Anzahl gleichzeitiger Konten und Fehlertoleranz finden Sie unter StackSet Operationsoptionen.

So funktionieren die einzelnen Parallelitätsmodi

Die folgenden Bilder zeigen visuell, wie die einzelnen Parallelitätsmodi während eines StackSet Vorgangs funktionieren. Die Knotenfolge stellt eine Bereitstellung für einen einzelnen Knoten dar, AWS-Region und jeder Knoten ist ein Ziel AWS-Konto.

Strikte Fehlertoleranz

Wenn bei einem StackSet Vorgang, der Strict Failure Tolerance verwendet, der Wert für die Fehlertoleranz auf 5 und der Wert für die maximale Anzahl gleichzeitiger Konten auf 10 festgelegt ist, ist die tatsächliche Parallelität 6. Die tatsächliche Parallelität beträgt 6, da der Wert der Fehlertoleranz von 5 +1 niedriger ist als der Wert für Maximale Anzahl gleichzeitiger Konten.

Die folgende Abbildung zeigt die Auswirkungen, die der Wert für die Fehlertoleranz auf den Wert „Maximale Anzahl gleichzeitiger Konten“ hat, und zeigt, wie sich beide Werte auf die tatsächliche Parallelität des Vorgangs auswirken: StackSet

Wenn die Bereitstellung beginnt und es fehlerhafte Stack-Instances gibt, verringert sich die tatsächliche Parallelität, um eine sichere Bereitstellung zu gewährleisten. Die tatsächliche Parallelität wird von 6 auf 5 reduziert, wenn eine StackSets Stack-Instance nicht bereitgestellt werden kann.

Der Modus Strikte Fehlertoleranz verringert die tatsächliche Parallelität proportional zur Anzahl der fehlerhaften Stack-Instances. Im folgenden Beispiel wird die tatsächliche Parallelität von 5 auf 3 reduziert, wenn 2 weitere Stack-Instances StackSets nicht bereitgestellt werden können, wodurch sich die Gesamtzahl der ausgefallenen Stack-Instances auf 3 erhöht.

StackSets schlägt beim StackSet Vorgang fehl, wenn die Anzahl der ausgefallenen Stack-Instances dem definierten Wert für Fehlertoleranz +1 entspricht. Im folgenden Beispiel StackSets schlägt der Vorgang fehl, wenn es 6 ausgefallene Stack-Instances gibt und der Fehlertoleranzwert 5 ist.

In diesem Beispiel CloudFormation wurden 9 Stack-Instances (3 erfolgreich und 6 gescheitert) bereitgestellt, bevor der StackSet Vorgang gestoppt wurde.

Lose Fehlertoleranz

Wenn bei einem StackSet Vorgang, der Soft Failure Tolerance verwendet, der Fehlertoleranzwert auf 5 und der Wert für maximale Anzahl gleichzeitiger Konten auf 10 festgelegt ist, ist die tatsächliche Parallelität 10.

Wenn die Bereitstellung beginnt und es fehlerhafte Stack-Instances gibt, ändert sich die tatsächliche Parallelität nicht. Im folgenden Beispiel ist 1 Stack-Vorgang fehlgeschlagen, die tatsächliche Parallelität bleibt jedoch bei 10.

Die tatsächliche Parallelität bleibt auch nach 2 weiteren Stack-Instance-Fehlern bei 10.

StackSets schlägt beim StackSet Vorgang fehl, wenn ausgefallene Stack-Instances den Fehlertoleranzwert überschreiten. Im folgenden Beispiel StackSets schlägt der Vorgang fehl, wenn 6 ausgefallene Stack-Instances vorhanden sind und die Anzahl der Fehlertoleranz 5 beträgt. Der Vorgang wird jedoch erst beendet, wenn die restlichen Vorgänge in der Parallelitätswarteschlange abgeschlossen sind.

StackSets setzt die Bereitstellung von Stack-Instances fort, die sich bereits in der Parallelitätswarteschlange befinden. Das bedeutet, dass die Anzahl der fehlerhaften Stack-Instances höher sein kann als die Fehlertoleranz. Im folgenden Beispiel sind 8 Stack-Instances ausgefallen, da in der Parallelitätswarteschlange noch 7 Operationen ausgeführt werden mussten, obwohl der StackSet Vorgang die Fehlertoleranz von 5 erreicht hatte.

In diesem Beispiel StackSets wurden 15 Stack-Instances (7 erfolgreich und 8 gescheitert) bereitgestellt, bevor der Stack-Vorgang gestoppt wurde.

Wählen zwischen strikter und loser Fehlertoleranz auf Grundlage der Bereitstellungsgeschwindigkeit

Die Wahl zwischen den Modi Strikte Fehlertoleranz und Weiche Fehlertoleranz hängt von der bevorzugten Geschwindigkeit Ihrer StackSet Bereitstellung und der zulässigen Anzahl von Bereitstellungsfehlern ab.

Die folgenden Tabellen zeigen, wie jeder Parallelitätsmodus mit einem StackSet Vorgang umgeht, der beim Versuch, insgesamt 1000 Stack-Instances bereitzustellen, fehlschlägt. In jedem Szenario ist der Wert der Fehlertoleranz auf 100 Stack-Instances und der Wert für Maximale Anzahl gleichzeitiger Konten auf 250 Stack-Instances eingestellt.

In diesem Beispiel werden Konten StackSets zwar als verschiebbares Fenster in die Warteschlange gestellt (siehe So funktionieren die einzelnen Parallelitätsmodi), aber dieses Beispiel zeigt den Vorgang in Batches, um die Geschwindigkeit der einzelnen Modi zu demonstrieren.

Strikte Fehlertoleranz

In diesem Beispiel mit dem Modus Strikte Fehlertoleranz wird die tatsächliche Parallelität im Verhältnis zur Anzahl der Fehler verringert, die jeweils im vorherigen Stapel aufgetreten sind. Jeder Stapel hat 20 fehlgeschlagene Instanzen, wodurch die tatsächliche Parallelität des folgenden Batches um 20 verringert wird, bis der StackSet Vorgang den Fehlertoleranzwert von 100 erreicht.

In der folgenden Tabelle liegt die anfängliche tatsächliche Parallelität des ersten Stapels bei 101 Stack-Instances. Die tatsächliche Parallelität beträgt 101, da der Wert der Fehlertoleranz (100) +1 niedriger ist als der Wert für Maximale Anzahl gleichzeitiger Konten (250). Jeder Stapel enthält 20 fehlerhafte Bereitstellungen von Stack-Instances, wodurch sich die tatsächliche Parallelität jedes nachfolgenden Stapels um 20 Stack-Instances verringert.

Strikte Fehlertoleranz	Stapel 1	Stapel 2	Stapel 3	Stapel 4	Stapel 5	Stapel 6
Anzahl der tatsächlichen Parallelität	101	81	61	41	21	-

Strikte Fehlertoleranz	Stapel 1	Stapel 2	Stapel 3	Stapel 4	Stapel 5	Stapel 6
Anzahl fehlerhafter Instances	20	20	20	20	20	-
Anzahl erfolgreicher Stack-Instances	81	61	41	21	1	-

Bei dem Vorgang, bei dem Strict Failure Tolerance verwendet wurde, wurden 305 Stack-Instance-Bereitstellungen in 5 Batches abgeschlossen, bis der StackSet Vorgang die Fehlertoleranz von 100 Stack-Instances erreichte. Der StackSet Vorgang stellt erfolgreich 205 Stack-Instances bereit, bevor er fehlschlägt.

Lose Fehlertoleranz

In diesem Beispiel mit dem Modus Lose Fehlertoleranz wird unabhängig von der Anzahl der fehlerhaften Instances die gleiche Anzahl der tatsächlichen Parallelität beibehalten, die durch den Wert für Maximale Anzahl gleichzeitiger Konten von 250 Stack-Instances definiert ist. Bei den StackSet Vorgängen wird dieselbe tatsächliche Parallelität beibehalten, bis der Fehlertoleranzwert von 100 Instanzen erreicht ist.

In der folgenden Tabelle liegt die anfängliche tatsächliche Parallelität des ersten Stapels bei 250 Stack-Instances. Die tatsächliche Parallelität liegt bei 250, da der Wert „Maximale Anzahl gleichzeitiger Konten“ auf 250 festgelegt ist und der Modus „Weiche Fehlertoleranz“ es StackSets ermöglicht, diesen Wert unabhängig von der Anzahl der Fehler als tatsächliche Parallelität zu verwenden. Obwohl in diesem Beispiel in jedem Stapel 50 Fehler aufgetreten sind, bleibt die tatsächliche Parallelität davon unberührt.

Lose Fehlertoleranz	Stapel 1	Stapel 2	Stapel 3	Stapel 4	Stapel 5	Stapel 6
Anzahl der tatsächlichen Parallelität	250	250	-	-	-	-
Anzahl fehlerhafter Instances	50	50	-	-	-	-
Anzahl erfolgreicher Stack-Instances	200	200	-	-	-	-

Mit den gleichen Werten für Maximale Anzahl gleichzeitiger Konten und Fehlertoleranz wurden beim Vorgang mit dem Modus Lose Fehlertoleranz 500 Bereitstellungen von Stack-Instances in 2 Stapeln abgeschlossen. Der StackSet Vorgang stellt erfolgreich 400 Stack-Instances bereit, bevor er fehlschlägt.

Wählen Sie Ihren Parallelitätsmodus (Konsole)

Wählen Sie beim Erstellen oder Aktualisieren eines StackSet auf der Seite „Bereitstellungsoptionen festlegen“ für den Parallelitätsmodus die Option Strikte Fehlertoleranz oder Weiche Fehlertoleranz aus.

Wählen Sie Ihren Parallelitätsmodus ()AWS CLI

Sie können den `ConcurrencyMode` Parameter mit den folgenden StackSets Befehlen verwenden:

- create-stack-instances
- delete-stack-instances
- detect-stack-set-drift
- import-stacks-to-stack-set

- update-stack-instances
- update-stack-set

Diese Befehle verfügen über einen bestehenden Parameter namens `--operation-preferences`, bei dem die Einstellung `ConcurrencyMode` verwendet werden kann. `ConcurrencyMode` kann auf einen der folgenden Werte eingestellt werden:

- `STRICT_FAILURE_TOLERANCE`
- `SOFT_FAILURE_TOLERANCE`

Im folgenden Beispiel wird eine Stack-Instance mithilfe von `STRICT_FAILURE_TOLERANCEConcurrencyMode`, mit einem `FailureToleranceCount` Satz auf 10 und einem `MaxConcurrentCount` Satz auf 5 erstellt.

```
aws cloudformation create-stack-instances \
  --stack-set-name example-stackset \
  --accounts 123456789012 \
  --regions eu-west-1  \
  --operation-preferences
 ConcurrencyMode=STRICT_FAILURE_TOLERANCE,FailureToleranceCount=10,MaxConcurrentCount=5
```

> **Note**
>
> Ausführliche Verfahren zum Erstellen und Aktualisieren einer StackSet finden Sie in den folgenden Themen:
>
> - Erstellen StackSets (selbstverwaltete Berechtigungen)
> - Erstellen StackSets (vom Service verwaltete Berechtigungen)
> - Aktualisiere StackSets

Drifterkennung durchführen am CloudFormation StackSets

Selbst wenn Sie Ihre Stacks und die darin enthaltenen Ressourcen verwalten CloudFormation, können Benutzer diese Ressourcen außerhalb von CloudFormation ändern. Benutzer können Ressourcen direkt bearbeiten, indem sie den zugrunde liegenden Service verwenden, der die Ressource erstellt hat. Indem Sie eine Drift-Erkennung für eine durchführen StackSet, können Sie

feststellen, ob sich eine der Stack-Instances, die zu dieser Gruppe gehören, von ihrer erwarteten Konfiguration StackSet abweicht oder von der erwarteten Konfiguration abweicht.

Themen

- Wie CloudFormation funktioniert die Drifterkennung auf einem StackSet
- Erkennen Sie Drift auf einer StackSet (Konsole)
- Erkennt Drift auf einem StackSet (AWS CLI)
- Stoppen der Drifterkennung auf einem StackSet

Wie CloudFormation funktioniert die Drifterkennung auf einem StackSet

Wenn eine Drifterkennung auf einem CloudFormation durchgeführt wird StackSet, führt sie eine Drifterkennung auf dem Stack durch, der jeder Stack-Instance in der zugeordnet ist StackSet. CloudFormation vergleicht dazu den aktuellen Status jeder Ressource im Stack mit dem erwarteten Zustand dieser Ressource, wie in der Vorlage des Stacks definiert, und den angegebenen Eingabeparametern. Wenn der aktuelle Status einer Ressource vom erwarteten Zustand abweicht, wird davon ausgegangen, dass diese Ressource abgewichen ist. Wenn eine oder mehrere Ressourcen in einem Stack abgewichen sind, wird der Stack selbst als abgewichen angesehen, und die Stack-Instances, denen der Stack zugeordnet ist, werden auch als abgewichen angesehen. Wenn sich eine oder mehrere Stack-Instances in einem verändert StackSet haben, wird davon ausgegangen, dass die Stack-Instances StackSet selbst gedriftet sind.

Die Drift-Erkennung identifiziert nicht verwaltete Änderungen, d. h. Änderungen, die an Stacks außerhalb von vorgenommen wurden. CloudFormation Änderungen, die direkt und nicht CloudFormation auf StackSet Ebene eines Stacks vorgenommen werden, gelten nicht als Drift. Nehmen wir zum Beispiel an, Sie haben einen Stack, der mit einer Stack-Instanz von a verknüpft ist StackSet. Wenn Sie diesen Stack aktualisieren, um eine andere Vorlage zu verwenden, wird dies nicht als Abweichung angesehen, obwohl dieser Stack jetzt eine andere Vorlage hat als alle anderen Stacks, die zu dem StackSet gehören. CloudFormation Dies liegt daran, dass der Stack immer noch mit seiner erwarteten Vorlagen- und Parameterkonfiguration in CloudFormation übereinstimmt.

Ausführliche Informationen darüber, wie die CloudFormation Drifterkennung auf einem Stack durchgeführt wird, finden Sie unterErkennen Sie mit Drift-Erkennung nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen.

Da die CloudFormation Drifterkennung für jeden Stapel einzeln durchgeführt wird, werden bei der Bestimmung, ob ein Stapel driftet ist, alle überschriebenen Parameterwerte berücksichtigt.

Weitere Hinweise zum Überschreiben von Vorlagenparametern in Stack-Instances finden Sie unter Überschreiben Sie Parameterwerte auf Stacks in Ihrem CloudFormation StackSet.

Wenn Sie die Drifterkennung direkt auf einem Stack durchführen, der einer Stack-Instance zugeordnet ist, sind diese Drift-Ergebnisse nicht auf der StackSetsKonsolenseite verfügbar.

Erkennen Sie Drift auf einer StackSet (Konsole)

Um Drift auf einem zu erkennen StackSet

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der StackSetsSeite die StackSet aus, für die Sie die Drift-Erkennung durchführen möchten.
3. Wählen Sie im Menü Aktionen die Option Abweichungen erkennen aus.

   CloudFormation zeigt eine Informationsleiste an, in der angegeben wird, dass die Drifterkennung für das gewählte Objekt eingeleitet wurde StackSet.
4. Optional: So überwachen Sie den Fortschritt des Abweichungserkennungsvorgangs:

   a. Wählen Sie den StackSet Namen aus, um die Stackset-Detailseite anzuzeigen.

   b. Wählen Sie die Registerkarte Vorgänge, den Abweichungserkennungsvorgang und dann Abweichungsdetails anzeigen aus.

   CloudFormation zeigt das Dialogfeld mit den Vorgangsdetails an.
5. Warten Sie, bis der Vorgang zur Drifterkennung CloudFormation abgeschlossen ist. Wenn der Vorgang zur Drifterkennung abgeschlossen ist, werden der Driftstatus und die Uhrzeit der letzten Driftprüfung für Ihren CloudFormation aktualisiert StackSet. Diese Felder sind auf der Registerkarte „Übersicht“ der StackSet Detailseite für die ausgewählten Felder aufgeführt StackSet.

   Die Drift-Erkennung kann einige Zeit in Anspruch nehmen, abhängig von der Anzahl der in der StackSet enthaltenen Stack-Instances und der Anzahl der in der enthaltenen Ressourcen StackSet. Sie können jeweils nur einen einzigen Drifterkennungsvorgang für eine bestimmte StackSet Datei ausführen. CloudFormation setzt die Drifterkennung fort, auch wenn Sie die Informationsleiste schließen.

6. Um die Ergebnisse der Drifterkennung für die Stack-Instances in a zu überprüfen StackSet, wählen Sie die Registerkarte Stack-Instances aus.

   In der Spalte Stack-Name wird der Name des Stacks aufgeführt, der jeder Stack-Instance zugeordnet ist, und in der Spalte Abweichungsstatus wird der Abweichungsstatus dieses Stacks aufgeführt. Ein Stack gilt als abgewichen, wenn eine oder mehrere seiner Ressourcen abgewichen sind.

7. So überprüfen Sie die Ergebnisse der Abweichungserkennung für den Stack, der mit einer bestimmten Stack-Instance verknüpft ist:

   a. Wählen Sie die Registerkarte Vorgänge aus.

   b. Wählen Sie den Abweichungsvorgang aus, für den Sie die Ergebnisse der Abweichungserkennung ansehen möchten. Der Status der Stack-Instance und der Grund für den ausgewählten Vorgang werden in einem geteilten Fenster angezeigt. Bei einem Abweichungsvorgang wird in der Spalte „Statusgrund“ der Abweichungsstatus einer Stack-Instance angezeigt.

   c. Wählen Sie die Stack-Instance aus, für die Sie Abweichungsdetails sehen möchten, und gehen Sie dann auf Ressourcenabweichungen anzeigen. In der Tabelle Ressourcenabweichungsstatus auf der Seite Ressourcenabweichungen werden alle Stack-Ressourcen mit ihrem jeweiligen Abweichungsstatus aufgeführt und Sie sehen, wann für die einzelnen Ressourcen zuletzt die Abweichungserkennung eingeleitet wurde. Die logische ID und die physische ID jeder Ressource werden angezeigt, um Ihnen bei der Identifizierung zu helfen.

8. Sie können die Ressourcen basierend auf ihrem Abweichungsstatus mithilfe der Spalte Drift status (Abweichungsstatus) sortieren.

   So zeigen Sie die Details zu einer geänderten Ressource an:

   - Wenn Sie die Ressource ausgewählt haben, wählen Sie Abweichungsdetails anzeigen aus.

     CloudFormation zeigt die Drift-Detailseite für diese bestimmte Ressource an. Auf dieser Seite werden die Unterschiede der Ressource aufgelistet. Außerdem werden die erwarteten und aktuellen Eigenschaftswerte der Ressource aufgeführt.

> **Note**
>
> Wenn die zugehörige Region bzw. das Konto des Stacks von der Region und dem Konto abweichen, in dem Sie derzeit angemeldet sind, ist die Schaltfläche Abweichung erkennen deaktiviert und Sie können die Details nicht anzeigen.

Erkennt Drift auf einem StackSet (AWS CLI)

Gehen Sie wie folgt vor AWS CLI, um Drift auf einem gesamten Stack mithilfe von zu erkennen:

Um Drift auf einem zu erkennen StackSet

1. Verwenden Sie den detect-stack-set-driftBefehl, um Drift bei einer gesamten Stack-Instance StackSet und den zugehörigen Stack-Instances zu erkennen.

   Das folgende Beispiel initiiert die Drifterkennung auf dem StackSet`stack-set-drift-example`.

   ```
   aws cloudformation detect-stack-set-drift \
       --stack-set-name stack-set-drift-example
   ```

   Ausgabe:

   ```
   {
       "OperationId": "c36e44aa-3a83-411a-b503-cb611example"
   }
   ```

2. Da StackSet Drifterkennungsvorgänge lange dauern können, verwenden Sie den describe-stack-set-operationBefehl, um den Status des Drift-Vorgangs zu überwachen. Dieser Befehl verwendet die vom detect-stack-set-drift Befehl zurückgegebene StackSet Vorgangs-ID.

   In den folgenden Beispielen wird die Operations-ID aus dem vorherigen Beispiel verwendet, um Informationen über den Vorgang zur StackSet Drifterkennung zurückzugeben. In diesem Beispiel wird der Vorgang noch ausgeführt. Von den sieben damit StackSet verknüpften Stack-Instances wurde bei einer Stack-Instance bereits ein Drift festgestellt, zwei Instances sind synchron und die Drift-Erkennung für die verbleibenden vier Stack-Instances ist noch im Gange. Da bei einer Instance ein Drift-Status eingetreten ist, ist der Drift-Status der Instance StackSet jetzt `DRIFTED`

```
aws cloudformation describe-stack-set-operation \
    --stack-set-name stack-set-drift-example \
    --operation-id c36e44aa-3a83-411a-b503-cb611example
```

Ausgabe:

```
{
    "StackSetOperation": {
        "Status": "RUNNING",
        "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
        "OperationPreferences": {
            "RegionOrder": []
        },
        "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
        "StackSetDriftDetectionDetails": {
            "DriftedStackInstancesCount": 1,
            "TotalStackInstancesCount": 7,
            "LastDriftCheckTimestamp": "2019-12-04T20:34:28.543Z",
            "InSyncStackInstancesCount": 2,
            "InProgressStackInstancesCount": 4,
            "DriftStatus": "DRIFTED",
            "FailedStackInstancesCount": 0
        },
        "Action": "DETECT_DRIFT",
        "CreationTimestamp": "2019-12-04T20:33:13.673Z",
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22example",
        "OperationId": "c36e44aa-3a83-411a-b503-cb611example"
    }
}
```

Wenn Sie den gleichen Befehl später ausführen, werden in diesem Beispiel die Informationen angezeigt, die zurückgegeben werden, sobald der Abweichungserkennungsvorgang abgeschlossen ist. Zwei der insgesamt sieben Stack-Instances, die damit in Verbindung stehen, StackSet haben sich verändert, sodass der Drift-Status der Instances StackSet sich wie folgt darstellt. `DRIFTED`

```
aws cloudformation describe-stack-set-operation \
    --stack-set-name stack-set-drift-example \
```

```
    --operation-id c36e44aa-3a83-411a-b503-cb611example
```

Ausgabe:

```
{
    "StackSetOperation": {
        "Status": "SUCCEEDED",
        "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
        "OperationPreferences": {
            "RegionOrder": []
        }
        "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
        "EndTimestamp": "2019-12-04T20:37:32.829Z",
        "StackSetDriftDetectionDetails": {
            "DriftedStackInstancesCount": 2,
            "TotalStackInstancesCount": 7,
            "LastDriftCheckTimestamp": "2019-12-04T20:36:55.612Z",
            "InSyncStackInstancesCount": 5,
            "InProgressStackInstancesCount": 0,
            "DriftStatus": "DRIFTED",
            "FailedStackInstancesCount": 0
        },
        "Action": "DETECT_DRIFT",
        "CreationTimestamp": "2019-12-04T20:33:13.673Z",
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22example",
        "OperationId": "c36e44aa-3a83-411a-b503-cb611example"
    }
}
```

3. Wenn der Vorgang zur StackSet Drifterkennung abgeschlossen ist, verwenden Sie die list-stack-instance-resource-drifts Befehle describe-stack-setlist-stack-instances,describe-stack-instance, und, um die Ergebnisse zu überprüfen.

   Der describe-stack-set-Befehl enthält dieselben detaillierten Abweichungsinformationen, die vom describe-stack-set-operation-Befehl zurückgegeben werden.

```
aws cloudformation describe-stack-set \
    --stack-set-name stack-set-drift-example
```

   Ausgabe:

```
{
    "StackSet": {
        "Status": "ACTIVE",
        "Description": "Demonstration of drift detection on stack sets.",
        "Parameters": [],
        "Tags": [
            {
                "Value": "Drift detection",
                "Key": "Feature"
            }
        ],
        "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
        "Capabilities": [],
        "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
        "StackSetDriftDetectionDetails": {
            "DriftedStackInstancesCount": 2,
            "TotalStackInstancesCount": 7,
            "LastDriftCheckTimestamp": "2019-12-04T20:36:55.612Z",
            "InProgressStackInstancesCount": 0,
            "DriftStatus": "DRIFTED",
            "DriftDetectionStatus": "COMPLETED",
            "InSyncStackInstancesCount": 5,
            "FailedStackInstancesCount": 0
        },
        "StackSetARN": "arn:aws:cloudformation:us-east-1:123456789012:stackset/
stack-set-drift-example:bd1f4017-d4f9-432e-a73f-8c22example",
        "TemplateBody": [details omitted],
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22ebexample",
        "StackSetName": "stack-set-drift-example"
    }
}
```

Sie können den list-stack-instancesBefehl verwenden, um zusammenfassende Informationen über die Stack-Instances zurückzugeben, die mit a verknüpft sind StackSet, einschließlich des Drift-Status jeder Stack-Instance.

In diesem Beispiel `DRIFTED` können Sie list-stack-instances anhand des Beispiels StackSet mit eingestelltem Driftstatus-Filter ermitteln, welche zwei Stack-Instances den Drift-Status haben`DRIFTED`.

```
aws cloudformation list-stack-instances \
    --stack-set-name stack-set-drift-example \
    --filters Name=DRIFT_STATUS,Values=DRIFTED
```

Ausgabe:

```
{
"Summaries": [
        {
"StackId": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
StackSet-stack-set-drift-example-b0fb6083-60c0-4e39-
af15-2f071e0db90c/0e4f0940-16d4-11ea-93d8-0641cexample",
            "Status": "CURRENT",
            "Account": "012345678910",
            "Region": "eu-west-1",
            "LastDriftCheckTimestamp": "2019-12-04T20:37:32.687Z",
            "DriftStatus": "DRIFTED",
            "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
            "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
        },
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
StackSet-stack-set-drift-example-b7fde68e-e541-44c2-b33d-
ef2e2988071a/008e6030-16d4-11ea-8090-12f89example",
            "Status": "CURRENT",
            "Account": "123456789012",
            "Region": "us-east-1",
            "LastDriftCheckTimestamp": "2019-12-04T20:34:28.275Z",
            "DriftStatus": "DRIFTED",
            "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
            "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
        },

        [additional stack instances omitted]

    ]
}
```

Der describe-stack-instance-Befehl gibt auch diese Informationen zurück, aber für eine einzelne Stack-Instance, wie im folgenden Beispiel.

```
aws cloudformation describe-stack-instance \
    --stack-set-name stack-set-drift-example \
    --stack-instance-account 012345678910 --stack-instance-region us-east-1
```

Ausgabe:

```
{
    "StackInstance": {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
StackSet-stack-set-drift-example-b7fde68e-e541-44c2-b33d-
ef2e2988071a/008e6030-16d4-11ea-8090-12f89example",
        "Status": "CURRENT",
        "Account": "123456789012",
        "Region": "us-east-1",
        "ParameterOverrides": [],
        "DriftStatus": "DRIFTED",
        "LastDriftCheckTimestamp": "2019-12-04T20:34:28.275Z",
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
        "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
    }
}
```

4. Sobald Sie festgestellt haben, welche Stack-Instances sich verändert haben, können Sie die Informationen über die Stack-Instances verwenden, die von den describe-stack-instance Befehlen list-stack-instances oder zurückgegeben werden, um den list-stack-instance-resource-driftsBefehl auszuführen. Dieser Befehl gibt detaillierte Informationen darüber zurück, welche Ressourcen im Stack für einen bestimmten Abweichungsvorgang abgewichen sind.

   Im folgenden Beispiel wird der Parameter `--stack-instance-resource-drift-statuses` verwendet, um Informationen zur Stack-Abweichung für die Ressourcen abzurufen, die im Abweichungsvorgang aus dem vorherigen Beispiel geändert oder gelöscht wurden. Die Anfrage gibt Informationen über die eine Ressource zurück, die geändert wurde, einschließlich Details über zwei ihrer Eigenschaften und deren geänderte Werte. Es wurden keine Ressourcen gelöscht.

```
aws cloudformation list-stack-instance-resource-drifts \
```

```
    --stack-set-name my-stack-set-with-resource-drift \
    --stack-instance-account 123456789012 \
    --stack-instance-region us-east-1 \
    --operation-id c36e44aa-3a83-411a-b503-cb611example \
    --stack-instance-resource-drift-statuses MODIFIED DELETED
```

Ausgabe:

```
{
    "Summaries": [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/my-
stack-set-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
            "ResourceType": "AWS::SQS::Queue",
            "Timestamp": "2018-03-26T17:23:34.489Z",
            "PhysicalResourceId": "https://sqs.us-
east-1.amazonaws.com/123456789012/my-stack-with-resource-drift-Queue-494PBHCO76H4",
            "StackResourceDriftStatus": "MODIFIED",
            "PropertyDifferences": [
                {
                    "PropertyPath": "/DelaySeconds",
                    "ActualValue": "120",
                    "ExpectedValue": "20",
                    "DifferenceType": "NOT_EQUAL"
                },
                {
                    "PropertyPath": "/RedrivePolicy/maxReceiveCount",
                    "ActualValue": "12",
                    "ExpectedValue": "10",
                    "DifferenceType": "NOT_EQUAL"
                }
            ],
            "LogicalResourceId": "Queue"
        }
    ]
}
```

Stoppen der Drifterkennung auf einem StackSet

Da die Drifterkennung auf einem ein langwieriger Vorgang sein StackSet kann, kann es vorkommen, dass Sie einen Drifterkennungsvorgang beenden möchten, der gerade auf einem StackSet ausgeführt wird.

Um die Drifterkennung auf einer StackSet (Konsole) zu beenden

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der StackSetsSeite den Namen des. StackSet

   CloudFormation zeigt die StackSets Detailseite für das Ausgewählte an StackSet.
3. Wählen Sie auf der StackSets Detailseite die Registerkarte Operationen und dann den Vorgang zur Drifterkennung aus.
4. Wählen Sie Vorgang stoppen aus.

Um die Drifterkennung auf einem StackSet (AWS CLI) zu beenden

- Verwenden Sie den stop-stack-set-operation-Befehl. Sie müssen sowohl den StackSet Namen als auch die Vorgangs-ID des StackSet Drifterkennungsvorgangs angeben.

```
aws cloudformation stop-stack-set-operation \
    --stack-set-name stack-set-drift-example \
    --operation-id 624af370-311a-11e8-b6b7-500cexample
```

Stapel importieren in AWS CloudFormation StackSets

Ein Stack-Importvorgang kann bestehende Stacks in neue oder bestehende importieren StackSets, sodass Sie vorhandene Stacks in einen Vorgang migrieren können. StackSet

Bei der Selbstverwaltung StackSets können beim Importvorgang Stacks in das Administratorkonto oder in verschiedene Zielkonten und importiert werden. AWS-Regionen Bei Service Managed StackSets kann der Importvorgang jeden beliebigen Stack importieren, der sich in demselben AWS Organizations Verwaltungskonto befindet.

Im Folgenden finden Sie Überlegungen und Einschränkungen beim Import von Stacks in: StackSets

- Der Importvorgang kann bis zu 10 Stapel mithilfe eines Inline-Stacks IDs oder bis zu 200 Stapel mithilfe eines Amazon S3 S3-Objekts importieren.
- Die Eigenschaft `NoEcho` wird nicht unterstützt. Stapel, die Folgendes enthalten, werden `NoEcho` nicht durch Import in neue StackSets importiert. StackSet
- Stapel können nur zu einem gehören. StackSet
- Sie können Stack-Tags für die implementieren, StackSet indem Sie beim Stack-Import Tags explizit als Parameter angeben.
- Die benutzerdefinierten Parameter-Überschreibungen eines Stacks sind während des Importvorgangs nicht betroffen.
- Die StackSets Kontingente und Stack-Instanzen gelten beim Import von Stacks. Weitere Informationen zu Kontingenten finden Sie unter CloudFormation Kontingente verstehen.

Themen

- Selbstverwalteter Stack-Import für AWS CloudFormation StackSets
- Vom Service verwalteter Stack-Import für AWS CloudFormation StackSets
- Stack-Importe rückgängig machen in AWS CloudFormation StackSets

Selbstverwalteter Stack-Import für AWS CloudFormation StackSets

Mit dem CloudFormation Stack-Importvorgang können vorhandene Stacks in neue oder bestehende Stacks importiert werden StackSets, sodass Sie vorhandene Stacks in einen StackSet Vorgang migrieren können. Durch die Verwendung von Stack-Import vermeiden Sie Ausfallzeiten und Ausfälle, ohne diese Ressourcen zu löschen und neu zu erstellen. Sobald der Stack in einen importiert wurde StackSet, wird der ursprüngliche Stack zu einer Stack-Instanz des angegebenen Stack-Sets.

Überlegungen zu selbstverwalteten Stack-Importen

- Der Stack-Importvorgang erfordert ein Administratorkonto, in dem Sie ein Konto erstellen, StackSet und ein Zielkonto, das einen Stack enthält.
- Das Zielkonto muss die Berechtigung haben, den `GetTemplate`-Vorgang mit der Eingabe der Stack-ID oder des ARN zu verwenden. Aus diesem Grund müssen Ihrem Administratorkonto AWSCloudFormationStackSetsExectionRoleBerechtigungen erteilt AWSCloudFormationStackSetAdministrationRolewerden.

Themen

- Importiert einen vorhandenen Stack in einen neuen StackSet (Konsole)
- Importiert einen vorhandenen Stack in einen vorhandenen StackSet (Konsole)
- Importiert einen Stack in ein StackSet (AWS CLI)

Importiert einen vorhandenen Stack in einen neuen StackSet (Konsole)

Bevor Sie beginnen, identifizieren Sie den Stack, den Sie importieren möchten.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. StackSets.
3. Wählen Sie oben auf der StackSetsSeite Create aus. StackSet
4. Geben Sie auf der Seite Vorlage auswählen eine Vorlage mit einer der folgenden Optionen an und klicken Sie auf Weiter.
   - Wählen Sie Amazon-S3-URL aus und geben Sie dann die URL für Ihre Vorlage im Textfeld an.
   - Wählen Sie Upload a template file (Vorlagendatei hochladen) aus und suchen Sie nach Ihrer Vorlage.
   - Wählen Sie From Stack-ID und geben Sie Ihre Stack-ID ein.
5. Geben Sie auf der Seite „ StackSet Details angeben“ den Namen einer Datei ein, die StackSet Sie erstellen möchten, und klicken Sie auf Weiter.

   (Optional) Geben Sie eine Beschreibung von ein StackSet.
6. Überprüfen Sie auf der Seite „ StackSet Optionen konfigurieren“ Ihre Auswahl und wählen Sie Weiter.
7. Wählen Sie auf der Seite Bereitstellungsoptionen festlegen die Option Stacks in Stack-Set importieren aus.
8. Geben Sie die Stack-ID des Stacks, den Sie importieren möchten, in das Feld Stacks to import (Zu importierende Stacks) ein. Beispiel, *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*.

   (Optional) Wählen Sie Weitere Stack-ID hinzufügen und geben Sie die Stack-ID eines anderen Stacks ein, den Sie importieren möchten. Sie können bis zu 10 Stacks pro Stack-Importvorgang hinzufügen.
9. Überprüfen Sie Ihre Bereitstellungsoptionen und wählen Sie Weiter.

10. Überprüfen Sie auf der Seite „Überprüfen“ Ihre Auswahl und Ihre StackSet Eigenschaften. Wenn Sie bereit sind, Ihren Stack in Ihren zu importieren StackSet, wählen Sie Senden.

Ergebnisse: Der importierte Stack ist jetzt eine Stack-Instanz des angegebenen StackSet. Weitere Informationen zum Status des Stack-Imports finden Sie unterStackSets Statuscodes.

Importiert einen vorhandenen Stack in einen vorhandenen StackSet (Konsole)

Bevor Sie beginnen, identifizieren Sie den Stack, den Sie importieren möchten.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. StackSets.
3. Wählen Sie auf der StackSetsSeite den StackSet aus, in den Sie einen Stack importieren möchten.
4. Wenn das StackSet ausgewählt ist, wählen Sie im Menü Aktionen die Option Stapel hinzufügen StackSet aus.
5. Wählen Sie auf der Seite Bereitstellungsoptionen festlegen die Option Stacks in Stack-Set importieren aus und geben Sie die Stack-ID des Stacks, den Sie importieren möchten, in das Feld Zu importierende Stacks ein. Beispiel, *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*.

   (Optional) Wählen Sie Weitere Stack-ID hinzufügen und geben Sie die Stack-ID eines anderen Stacks ein, den Sie importieren möchten. Sie können bis zu 10 Stacks pro Stack-Importvorgang hinzufügen.
6. Wählen Sie Weiter aus.
7. Überprüfen Sie auf der Seite „Überschreibungen angeben“ Ihre Auswahl und wählen Sie Weiter.
8. Überprüfen Sie auf der Seite „Überprüfen“ Ihre Auswahl und Ihre StackSet Eigenschaften. Wenn Sie bereit sind, Ihre zu erstellen StackSet, wählen Sie Senden.

Ergebnisse: Der importierte Stack ist jetzt eine Stack-Instanz des angegebenen StackSet. Weitere Informationen zum Status des Stack-Imports finden Sie unterStackSets Statuscodes.

Importiert einen Stack in ein StackSet (AWS CLI)

Um einen vorhandenen Stapel in einen neuen zu importieren StackSet

Der folgende `create-stack-set` Befehl erstellt einen StackSet und importiert den angegebenen Stack. Der zu importierende Stack wird durch seinen ARN identifiziert. Ersetzen Sie den Platzhaltertext durch Ihre eigenen Informationen.

```
aws cloudformation create-stack-set \
  --stack-set-name MyStackSet \
  --stack-id arn:aws:cloudformation:us-east-1:123456789012:stack/
StackToImport/466df9e0-0dff-08e3-8e2f-5088487c4896 \
  --administration-role-arn arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole \
  --execution-role-name AWSCloudFormationStackSetExecutionRole
```

Um einen vorhandenen Stack in einen vorhandenen zu importieren StackSet

Der folgende `import-stacks-to-stack-sets` Befehl importiert den angegebenen Stack in den *`MyStackSet`* StackSet. Der zu importierende Stack wird durch seinen ARN identifiziert. Ersetzen Sie den Platzhaltertext durch Ihre eigenen Informationen.

```
aws cloudformation import-stacks-to-stack-set \
  --stack-set MyStackSet \
  --stack-ids arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/
f449b250-b969-11e0-a185-5081d0136786
```

Um mehr als einen Stapel anzugeben, verwenden Sie das folgende Format für den Wert der `--stack-ids` Option.

```
--stack-ids "arn_1" "arn_2"
```

Um den importierten Stack in andere Regionen und Konten zu klonen

Der folgende `create-stack-instances` Befehl fügt Stack-Instances zu Ihrem hinzu StackSet. Ersetzen Sie den Platzhaltertext durch Ihre eigenen Informationen.

```
aws cloudformation create-stack-instances \
  --stack-set-name MyStackSet \
  --accounts '["account_ID_1","account_ID_2"]' \
  --regions '["region_1","region_2"]'
```

Vom Service verwalteter Stack-Import für AWS CloudFormation StackSets

Der CloudFormation Stapelimportvorgang kann vorhandene Stapel in neue oder bestehende importieren StackSets, sodass Sie vorhandene Stapel zu einem StackSet Vorgang migrieren können. StackSets erweitert die Funktionalität von Stacks, sodass Sie Stacks für mehrere Konten und Regionen mit einem einzigen Vorgang erstellen, aktualisieren oder löschen können.

Überlegungen zu vom Service verwalteten Stack-Importen

- Für den Stack-Import ist ein Verwaltungskonto oder ein delegiertes Administratorkonto erforderlich, mit dem Sie die zugehörigen Daten verwalten können, AWS Organizations z. B. um den vertrauenswürdigen Zugriff zu aktivieren. StackSets
- Bei den Zielkonten muss es sich um Mitglieder des vom Verwaltungskonto AWS Organizations verwalteten Kontos oder des delegierten Administratorkontos handeln.
- Der Zielstapel ist in einem der Ziele OUs vorhanden.
- Das Zielkonto sollte Mitglied von sein AWS Organizations.
- AWS Organizations Der Zugriff sollte im `ACTIVATED` Bundesstaat für die Organizations erfolgen.
- Stacks, die importiert werden, sollten in einem der Mitgliedskonten vorhanden sein und nicht im Verwaltungskonto.

Themen

- Importieren Sie einen vom Service verwalteten Stack in eine neue StackSet (Konsole)
- Erstellen Sie einen vom Service verwalteten Stack und importieren Sie ihn in einen vorhandenen Stack StackSet (Konsole)
- Importieren Sie einen vom Service verwalteten Stack in einen vorhandenen Stack StackSet (Konsole)
- Einen vom Service verwalteten Stack in einen StackSet () importieren AWS CLI

Importieren Sie einen vom Service verwalteten Stack in eine neue StackSet (Konsole)

Importieren Sie einen Stack in einen neuen StackSet mit dem AWS Management Console

Um einen neuen Stack in einen zu importieren StackSet, identifizieren Sie einen Stack, der die Ressource enthält, die Sie importieren möchten.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. StackSets.
3. Wählen Sie oben auf der StackSetsSeite Create aus. StackSet
4. Gehen Sie auf der Seite Vorlage auswählen wie folgt vor:

   a. Wählen Sie als StackSet Berechtigungsmodell vom Dienst verwaltete Berechtigungen aus.

   b. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorlage ist bereit aus und wählen Sie Ihre Vorlage mithilfe einer der folgenden Optionen aus:

      - Geben Sie für Amazon S3 S3-URL Ihre Amazon S3 S3-URL in das Feld Amazon S3 S3-URL ein.
      - Wählen Sie für „Eine Vorlagendatei hochladen“ eine CloudFormation Vorlage auf Ihrem lokalen Computer aus.

   Übernehmen Sie Ihre Einstellungen und wählen Sie Next (Weiter).
5. Gehen Sie auf der Seite „ StackSet Details angeben“ wie folgt vor:

   a. Geben Sie einen StackSet Namen in das StackSet Namensfeld ein.

   b. (Optional) Geben Sie im Beschreibungsbereich eine StackSet Beschreibung ein.

   Überprüfen Sie auf der Seite „ StackSet Optionen konfigurieren“ Ihre Auswahl und wählen Sie Weiter.
6. Gehen Sie auf der Seite Bereitstellungsoptionen festlegen wie folgt vor:

   a. Wählen Sie für Stacks zum Stack-Set hinzufügen die Option Stacks zum Stack-Set importieren aus.

   b. Wählen Sie für Stacks to import (Zu importierende Stacks) Ihre Stack-Import-Methode aus.

      i. Geben Sie für Stack-ID Ihre Stack ID ein.

      ii. Geben Sie für Stack-URL die Amazon S3 S3-URL ein.
7. Gehen Sie unter Organisationseinheiten zuordnen wie folgt vor:

   a. Wählen Sie Mit Organisation verknüpfen aus, um die Stammorganisationseinheit zu verwenden.

b. Wählen Sie Mit Organisationseinheiten verknüpfen (OUs), um die übergeordnete Organisationseinheit IDs für die zu importierenden Stapel einzugeben. Wenn zum Beispiel `Stack 1` und `Stack 2` unter `OU1` und `Stack 3` unter `OU2` liegen, geben Sie `OU1` und `OU2` ein.

Übernehmen Sie Ihre Einstellungen und wählen Sie Next (Weiter).

8. Überprüfen Sie Ihre Einstellungen auf der Seite Review (Überprüfen) und wählen Sie Submit (Absenden).

Erstellen Sie einen vom Service verwalteten Stack und importieren Sie ihn in einen vorhandenen Stack StackSet (Konsole)

Um einen vorhandenen Stack in einen neuen zu importieren StackSet, identifizieren Sie einen Stack, der die Ressource enthält, die Sie importieren möchten.

Um einen Stapel zu erstellen StackSet und zu importieren

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. StackSets.
3. Wählen Sie oben auf der StackSetsSeite Create aus. StackSet
4. Gehen Sie auf der Seite Vorlage auswählen wie folgt vor:

   a. Wählen Sie als StackSet Berechtigungsmodell vom Dienst verwaltete Berechtigungen aus.

   b. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorlage ist bereit aus und wählen Sie Ihre Vorlage mithilfe einer der folgenden Optionen aus:

      - Geben Sie für Amazon S3 S3-URL Ihre Amazon S3 S3-URL in das Feld Amazon S3 S3-URL ein.
      - Wählen Sie für „Eine Vorlagendatei hochladen“ eine CloudFormation Vorlage auf Ihrem lokalen Computer aus.

   Übernehmen Sie Ihre Einstellungen und wählen Sie Next (Weiter).

5. Gehen Sie auf der Seite „ StackSet Details angeben“ wie folgt vor:

   a. Geben Sie einen StackSet Namen in das StackSet Namensfeld ein.

b. (Optional) Geben Sie im Beschreibungsbereich eine StackSet Beschreibung ein.

Überprüfen Sie auf der Seite „ StackSet Optionen konfigurieren“ Ihre Auswahl und wählen Sie Weiter.

6. Gehen Sie auf der Seite Bereitstellungsoptionen festlegen wie folgt vor:

   - Wählen Sie für Add stacks to stack set (Stacks zum Stack-Set hinzufügen) Deploy new stacks (Neue Stacks bereitstellen).

7. Gehen Sie im Abschnitt Organisationseinheiten zuordnen wie folgt vor:

   a. Wählen Sie Mit Organisation verknüpfen, um die Stammorganisationseinheit zu verwenden.

   b. Wählen Sie Mit Organisationseinheiten verknüpfen (OUs), um die übergeordnete Organisationseinheit IDs für die zu importierenden Stapel einzugeben. Wenn zum Beispiel `Stack 1` und `Stack 2` unter `OU1` und `Stack 3` unter `OU2` liegen, geben Sie `OU1` und `OU2` ein.

8. Überprüfen Sie Ihre Auswahl für Specify regions (Regionen angeben) und Deployment options (Bereitstellungs-Optionen).

   Übernehmen Sie Ihre Einstellungen und wählen Sie Next (Weiter).

9. Überprüfen Sie Ihre Einstellungen auf der Seite Review (Überprüfen) und wählen Sie Submit (Absenden).

Importieren Sie einen vom Service verwalteten Stack in einen vorhandenen Stack StackSet (Konsole)

Wählen Sie Ihren StackSet und identifizieren Sie den Stack, den Sie importieren möchten.

Um einen Stapel in einen vorhandenen zu importieren StackSet

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. StackSets.
3. Wählen StackSet Sie den Stack aus, in den Sie einen Stack importieren möchten, und wählen Sie dann Stapel hinzufügen aus dem Drop-down-Menü StackSet Aktionen aus.
4. Gehen Sie auf der Seite „Bereitstellungsoptionen festlegen“ wie folgt vor:

a. Wählen Sie für Stacks zum Stack-Set hinzufügen die Option Stacks zum Stack-Set importieren aus.
b. Gehen Sie unter Zu importierende Stapel wie folgt vor
   i. Geben Sie für Stack-ID Ihre Stack-ID ein.
   ii. Geben Sie für Stack-URL die Amazon S3 S3-URL ein.
c. Gehen Sie unter Organisationseinheiten zuordnen wie folgt vor:
   i. Wählen Sie Mit Organisation verknüpfen aus, um die Stammorganisationseinheit zu verwenden.
   ii. Wählen Sie Mit Organisationseinheiten verknüpfen (OUs), um die übergeordnete Organisationseinheit IDs für die zu importierenden Stapel einzugeben. Wenn zum Beispiel `Stack 1` und `Stack 2` unter 0U1 und `Stack 3` unter 0U2 liegen, geben Sie 0U1 und 0U2 ein.

   Übernehmen Sie Ihre Einstellungen und wählen Sie Next (Weiter).

5. Überprüfen Sie die Seite Specify overrides (Überschreibungen angeben) und wählen Sie Next (Weiter).
6. Bestätigen und überprüfen Sie die Seite Review (Überprüfung) und wählen Sie Submit (Absenden).

Einen vom Service verwalteten Stack in einen StackSet () importieren AWS CLI

Sobald StackSet ein erstellt wurde, können Sie Ihre Stacks importieren, indem Sie die Stack-IDs der importierten Stacks übergeben. Sie können auch die Liste der OU-IDs übergeben, der Sie sie zuordnen möchten.

CloudFormation importiert vom Benutzer bereitgestellte Stacks innerhalb dieser OUs und verwendet diese OUs als Bereitstellungsziele für. StackSet Der in IDs der Eingabe angegebene Stapel wird der intern eingegebenen OU in der OU-ID-Liste der nächstgelegenen Organisationseinheit zugeordnet. Wenn ein Stapel nicht zu einer vorhandenen OU-ID in der Eingabeliste gehört, AWS CLI wird der `StackNotFoundException` Fehler zurückgegeben.

Der `import-stacks-to-stack-set`-Vorgang erstellt Stack-Instances für die Stacks in der OU-ID-Eingabe. In den folgenden AWS CLI Beispielen wird die `import-stacks-to-stack-set` Operation verwendet, um einen Stapel in einen zu importieren StackSet.

- Um den `import-stacks-to-stack-sets`-Vorgang zu verwenden, geben Sie die `stack-ids` oder `stack-ids-url` an, die Sie in Ihr Stack-Set importieren möchten.

```
aws cloudformation import-stacks-to-stack-set \
  --stack-set-name ServiceMangedStackSet \
  --stack-ids "arn:123456789012:us-east-1:Stack1" \
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

```
aws cloudformation import-stacks-to-stack-set \
  --stack-set-name ServiceMangedStackSet \
  --stack-ids-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/file-
name.json \
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

> **Note**
>
> Für den `import-stacks-to-stack-sets`-Vorgang müssen Sie mindestens eine Organisationseinheits-ID (OU-ID) angeben, damit der zu importierende Stack dieser bestimmten OU zugeordnet werden kann. Bei diesem Vorgang werden keine Stack-Instanzen für andere Mitgliedskonten in den zugehörigen Konten erstellt OUs. Verwenden Sie `create-stack-instances` oder, um Mitgliedskonten für die OUs verknüpften Konten zu aktualisierenupdate-stack-instances.

`create-stack-set`erstellt Stack-Instances für alle Konten unter der Vorlage OUs mit einer vom Benutzer bereitgestellten Vorlage, entweder aus direktem Upload oder aus Amazon S3. In den folgenden AWS CLI Beispielen wird die `create-stack-set` Operation verwendet, um einen Stack in einen neuen zu importieren StackSet.

- Um die `create-stack-set` Operation zu verwenden, geben Sie Ihren StackSet Namen an und importieren Sie einen Stack in einen neu erstellten StackSet.

```
aws cloudformation create-stack-set \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/file-
name.json \
  --permission-model SERVICE_MANAGED \
  --auto-deployment Enabled=true
```

Stack-Importe rückgängig machen in AWS CloudFormation StackSets

Wenn Sie unerwünschte Änderungen an Ihrer Stack-Instance vorgenommen haben, können Sie den Stack-Instance-Import rückgängig machen.

Wenn Sie einen Stack-Instance-Import rückgängig machen, CloudFormation wird die Stack-Instance aus dem gelöscht, die Ressourcen des Stacks bleiben StackSet jedoch erhalten.

Gehen Sie wie folgt vor, um einen Stack-Importvorgang rückgängig zu machen.

1. Geben Sie ein `DeletionPolicy` Attribut von `Retain` für jede Ressource an, die Sie behalten möchten, nachdem die Stack-Instance gelöscht wurde. Weitere Informationen finden Sie unter Wiederherstellen eines Importvorgangs.
2. Löschen Sie Stack-Instances von Ihrem StackSet. Weitere Informationen finden Sie unter Stapel löschen von AWS CloudFormation StackSets .
3. Lösche deine StackSet. Weitere Informationen finden Sie unter Löschen AWS CloudFormation StackSets.

Bewährte Methoden für die Verwendung AWS CloudFormation StackSets

In diesem Abschnitt werden die bewährten Methoden zum Definieren einer StackSet Vorlage, zum Erstellen oder Hinzufügen von Stacks zu einer Vorlage oder zum Aktualisieren einer StackSet Vorlage beschrieben. StackSet

Falls Sie noch nicht damit vertraut sind CloudFormation, finden Sie im CloudFormation bewährte Verfahren Thema weitere Empfehlungen, die Ihnen helfen können, diese CloudFormation effektiver und sicherer zu nutzen.

Themen

- Definieren der Vorlage
- Erstellen oder Hinzufügen von Stacks zum StackSet
- Aktualisierung von Stacks in einem StackSet

Definieren der Vorlage

- Definieren Sie die Vorlage, die in mehreren Konten in mehreren Regionen standardisiert werden soll.
- Wenn Sie die Vorlage erstellen, achten Sie darauf, dass bei globalen Ressourcen (z. B. IAM-Rollen und Amazon S3-Buckets) keine Namenskonflikte auftreten, wenn Sie in mehreren Regionen im selben Konto erstellt werden.
- A StackSet hat eine einzige Vorlage und einen einzigen Parametersatz. Derselbe Stapel wird in allen Konten erstellt, die mit a verknüpft sind StackSet. Wenn Sie Vorlagen schreiben, sollten Sie sie so abgestuft gestalten, dass eine ausgeglichene Kontrolle und Standardisierung möglich ist.
- Wir empfehlen, Ihre Vorlage in einem Amazon S3-Bucket zu speichern.

Erstellen oder Hinzufügen von Stacks zum StackSet

- Stellen Sie sicher, dass das Hinzufügen von Stack-Instances zu Ihren ersten Instanzen StackSet funktioniert, bevor Sie eine größere Anzahl von Stack-Instances zu Ihrer StackSet hinzufügen.
- Wählen Sie die Bereitstellungs-(Rollout-)Optionen, die für Ihren Anwendungsfall am besten geeignet sind.
  - Für eine konservativere Bereitstellung legen Sie Maximum Concurrent Accounts auf 1 und Failure Tolerance auf 0 fest. Setzen Sie die Region mit den geringsten Auswirkungen an die erste Stelle in der Liste Region Order. Beginnen Sie mit einer Region.
  - Für eine schnellere Bereitstellung können Sie die Werte Maximum Concurrent Accounts und Failure Tolerance nach Bedarf erhöhen.
- Die laufenden Operationen StackSets hängen davon ab, wie viele Stack-Instances beteiligt sind, und können viel Zeit in Anspruch nehmen.

Aktualisierung von Stacks in einem StackSet

- Standardmäßig werden beim Aktualisieren von alle Stack-Instanzen StackSet aktualisiert. Wenn Sie jeweils 20 Konten in zwei Regionen haben, haben Sie 40 Stack-Instances, und alle werden aktualisiert, wenn Sie die aktualisieren StackSet.

  StackSets Bei einer großen Anzahl von Stack-Instances empfehlen wir, dass Sie zum Testen der aktualisierten Version einer Vorlage die Stack-Instances in einigen Testkonten selektiv aktualisieren, bevor Sie alle Stack-Instances aktualisieren.

- Um eine genauere Kontrolle über die Aktualisierung einzelner Stacks in Ihrem zu erhalten StackSet, planen Sie, mehrere zu erstellen. StackSets
- Das Aktualisieren einer Datei StackSet , die eine große Anzahl von Stacks enthält, kann viel Zeit in Anspruch nehmen. In dieser Version ist jeweils nur ein Vorgang an einem zulässig. StackSet Planen Sie Ihre Updates so, dass Sie nicht daran gehindert werden, andere Operationen auf dem durchzuführen StackSet.

AWS CloudFormation StackSets Beispielvorlagen

Dieser Abschnitt enthält Links zu einigen CloudFormation Beispielvorlagen, die Ihnen bei der Verwendung AWS CloudFormation StackSets in Ihrem Unternehmen helfen können. Die in diesem Abschnitt aufgeführten Vorlagen aktivieren AWS CloudTrail oder AWS Config und darin enthaltene Regeln.

> **Important**
>
> Als bewährte Sicherheitsmethode beim Zulassen des AWS Config Zugriffs auf einen Amazon S3 S3-Bucket empfehlen wir dringend, den Zugriff in der Bucket-Richtlinie mit der `AWS:SourceAccount` Bedingung einzuschränken. Neue Vorlagen wurden aktualisiert und verfügen nun über `AWS:SourceAccount`. Wenn Ihre vorhandene Bucket-Richtlinie dieser bewährten Sicherheitsmethode nicht entspricht, sollten Sie die Bucket-Richtlinie unbedingt so bearbeiten, dass sie diesen Schutz enthält. Dadurch AWS Config wird sichergestellt, dass der Zugriff nur für die erwarteten Benutzer gewährt wird.

Beschreibung	S3-Link
Aktivieren AWS CloudTrail	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudtrail.yml aktivieren
Aktivieren AWS Config	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSConfig.yml aktivieren

Beschreibung	S3-Link
AWS Config Mit zentraler Protokollierung aktivieren	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSConfigForOrganizations.yml aktivieren
Aktivieren Sie die Standardrichtlinien von Amazon Data Lifecycle Manager für eine AWS Organisation oder AWS für bestimmte Konten	• Standardrichtlinie für EBS-Snapshots — .yaml https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ DataLifecycleManager EBSSnapshot DefaultPolicy • Standardrichtlinie für EBS-gestützt — Policy.yaml AMIs https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ DataLifecycleManager AMIDefault https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml
Konfigurieren Sie eine AWS Config Regel, um festzustellen, ob sie CloudTrail aktiviert ist	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleCloudtrailEnabled.yml
Konfigurieren Sie eine AWS Config Regel, um festzustellen, ob Root-MFA aktiviert ist	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleRootAccountMFAEnabled.yml
Konfigurieren Sie eine AWS Config Regel, um festzustellen, ob sie angehängt EIPs sind	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleEipAttached.yml
Konfigurieren Sie eine AWS Config Regel, um festzustellen, ob EBS-Volumes verschlüsselt sind	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml

Problembehebung AWS CloudFormation StackSets

Dieses Thema enthält einige häufig auftretende StackSets Probleme und Lösungsvorschläge für diese Probleme.

Themen

- Häufige Gründe für Störungen der Stack-Operationen
- Wiederholen fehlgeschlagenen Operationen zum Erstellen und Aktualisieren von Stacks
- Löschen von Stack-Instances schlägt fehl
- Stack-Importvorgang schlägt fehl
- Anzahl der Stack-Instance-Fehler für StackSets Operationen

Häufige Gründe für Störungen der Stack-Operationen

Problem: Eine Stack-Operation ist fehlgeschlagen und der Instance-Status ist `OUTDATED`.

Ursache: Es kann verschiedene häufige Ursachen für die Störung der Stack-Operation geben.

- Nicht ausreichende Berechtigungen eines Zielkontos für die Erstellung von Ressourcen, die in Ihrer Vorlage angegeben sind.
- Die CloudFormation Vorlage enthält möglicherweise Fehler. Überprüfen Sie die Vorlage CloudFormation und beheben Sie Fehler, bevor Sie versuchen, Ihre zu erstellen StackSet.
- Die Vorlage könnte versuchen, globale Ressourcen zu erstellen, die eindeutig sein müssen, es aber nicht sind, wie z. B. S3-Buckets.
- Eine angegebene Ziel-Kontonummer ist nicht vorhanden. Prüfen Sie die Ziel-Kontonummern, die Sie auf der Seite Set deployment options des Assistenten angegeben haben.
- Das Administratorkonto hat keine Vertrauensbeziehung mit dem Zielkonto.
- Die maximale Anzahl an einer Ressource, die in Ihrer Vorlage angegeben ist, ist in Ihrem Zielkonto bereits vorhanden. Sie können beispielsweise die Grenze der zulässigen IAM-Rollen in einem Zielkonto erreicht haben, aber die Vorlage erstellt weitere IAM-Rollen.
- Sie haben die maximale Anzahl von Stapeln erreicht, die in a StackSet zulässig sind. Die maximale Anzahl von Stacks pro Stapel finden Sie StackSet unter. CloudFormation Kontingente verstehen

Lösung: Weitere Informationen zu den Berechtigungen, die für Ziel- und Administratorkonten erforderlich sind, bevor Sie ein Konto erstellen können StackSets, finden Sie unterErteilen Sie allen Benutzern des Administratorkontos die Berechtigung, Stacks in allen Zielkonten zu verwalten.

Wiederholen fehlgeschlagenen Operationen zum Erstellen und Aktualisieren von Stacks

Problem: Eine Erstellen oder Aktualisieren eines Stacks ist fehlgeschlagen und der Instance-Status ist `OUTDATED`. Um zu beheben, warum die Erstellung oder Aktualisierung eines Stacks fehlgeschlagen ist, öffnen Sie die CloudFormation Konsole und sehen Sie sich die Ereignisse für den Stack an, die den Status `DELETED` (für fehlgeschlagene Erstellungsvorgänge) oder `FAILED` (für fehlgeschlagene Aktualisierungsvorgänge) haben. Durchsuchen Sie die Stack-Ereignissen und zeigen Sie die Spalte Statusgrund an. Der Wert vom Statusgrund erklärt, warum die Stack-Operation fehlgeschlagen ist.

Nachdem Sie die Ursache für den Fehler bei der Erstellung des Stacks behoben haben, können Sie das Erstellen der Stacks wiederholen. Dazu führen Sie die folgenden Schritte aus.

Lösung: Führen Sie die folgenden Schritte aus, um Ihre Stack-Operation zu wiederholen.

1. Wählen Sie in der Konsole den aus StackSet , der den Stack enthält, auf dem der Vorgang fehlgeschlagen ist.
2. Wählen Sie im Menü Aktionen die Option StackSetDetails bearbeiten aus, um erneut zu versuchen, Stacks zu erstellen oder zu aktualisieren.
3. Verwenden Sie auf der Seite Specify template (Vorlage angeben) dieselbe CloudFormation-Vorlage und behalten Sie die Standardoption, Current template (Aktuelle Vorlage), bei. Wenn Ihre Stack-Operation fehlgeschlagen ist, weil in der Vorlage Änderungen erforderlich waren, und Sie eine überarbeitete Vorlage hochladen wollen, wählen Sie stattdessen Upload a template to Amazon S3 und wählen dann Browse, um Ihre aktualisierte Vorlage auszuwählen. Nachdem Sie Ihre geänderte Vorlage hochgeladen haben, wählen Sie Next.
4. Wenn Sie auf der Seite „Stack-Details angeben“ keine für Ihre Vorlage spezifischen Parameter ändern, wählen Sie Weiter.
5. Ändern Sie auf der Seite Set deployment options die Standardwerte für Maximum concurrent accounts und gegebenenfalls für Failure tolerance. Weitere Informationen zu diesen Einstellungen finden Sie unter StackSet Operationsoptionen.
6. Überprüfen Sie auf der Seite Review Ihre Auswahlen und markieren Sie das Kontrollkästchen, um die erforderlichen IAM-Kapazitäten zu bestätigen. Wählen Sie Absenden aus.

7. Wenn Ihr Stack nicht erfolgreich aktualisiert wurde, wiederholen Sie dieses Verfahren, nachdem Sie die Probleme bei der Stack-Erstellung gelöst haben.

Löschen von Stack-Instances schlägt fehl

Problem: Das Löschen eines Stacks ist fehlgeschlagen.

Ursache: Das Löschen eines Stacks schlägt bei allen Stacks fehl, deren Beendigungsschutz aktiviert wurde.

Lösung: Ermitteln Sie, ob der Beendigungsschutz für den Stack aktiviert wurde. Wenn dies der Fall ist, deaktivieren Sie den Beendigungsschutz und wiederholen Sie dann das Löschen der Stack-Instance.

Stack-Importvorgang schlägt fehl

Problem: Bei einem Stack-Importvorgang können vorhandene Stacks nicht in neue oder bestehende StackSets importiert werden. Die Stack-Instance befindet sich in einem `INOPERABLE` Status.

Lösung: Setzten Sie den Stack-Importvorgang zurück, indem Sie die folgenden Aufgaben ausführen.

1. Verwenden Sie die StackSets Option Stacks löschen aus und aktivieren Sie sie RetainStackswährend der Konfiguration. Fahren Sie dann mit dem Löschen von Stack-Instances von Ihrem fort. StackSet Weitere Informationen finden Sie unterStapel löschen von AWS CloudFormation StackSets .
2. Sie werden sehen, dass die Stack-Instances von aktualisiert StackSet werden, um die `INOPERABLE` Stack-Instance zu entfernen.
3. Korrigieren Sie die Stack-Instances entsprechend dem Importfehler und wiederholen Sie den Stack-Importvorgang.

Anzahl der Stack-Instance-Fehler für StackSets Operationen

Die Anzahl der Stack-Instance-Fehler warnt Sie, wenn Stack-Instances nicht bereitgestellt oder aktualisiert werden können. Diese Stack-Instances konnten aus einem oder mehreren der folgenden Gründe nicht bereitgestellt werden:

- Bestehende Ressource(n) mit ähnlicher Konfiguration

- Fehlende Abhängigkeiten, wie z. B. AWS Identity and Access Management (IAM-) Rollen
- Andere widersprüchliche Faktoren

Wenn Sie mit maximaler Gleichzeitigkeit bereitstellen möchten, ist die Anzahl der maximalen Gleichzeitigkeit höchstens um eins höher als die Fehlertoleranz. Wenn die Anzahl der Fehlertoleranzen beispielsweise 9 beträgt, darf die maximale Anzahl der Parallelitäten nicht mehr als 10 betragen. Dadurch wird der Vorgang auch dann `SUCCEEDED` zurückgeben, wenn einige Stack-Instances nicht aktualisiert werden können. Mit der neuen Anzahl der Stack-Instances können Sie feststellen, ob der Vorgang nur bedingt erfolgreich war, da die Fehlertoleranzzahl so eingestellt ist, dass alle Fehler zugelassen werden.

Sie können das AWS SDK oder verwenden AWS Management Console, AWS CLI um die Anzahl der Fehler abzurufen und Stack-Instances zu filtern, um zu ermitteln, welche Instanzen erneut bereitgestellt werden müssen.

Verwenden der Konsole

So zeigen Sie die Gesamtzahl der ausgefallenen Stack-Instances an:

1. Öffnen Sie die CloudFormation-Konsole, und wählen Sie StackSets aus.
2. Wählen Sie Ihre StackSet aus und wählen Sie dann die Registerkarte Operationen.
3. Wählen Sie in der Spalte Status (Status) einen Status aus, um die Statusdetails anzuzeigen. Die Anzahl der ausgefallenen Stack-Instances für einen bestimmten Vorgang finden Sie in den Statusdetails.

Um das Konto, die Region und den Status der Stack-Instances für den Vorgang einzusehen, gehen Sie wie folgt vor:

1. Wählen Sie in den Statusdetails die Anzahl der ausgefallenen Stack-Instances aus. Beispiel: Stack Instances: **`<number of failed stack instances>`** (Stack-Instances).
2. Erweitern Sie den Seitenbereich, indem Sie den Header des Bereichs auswählen. Die Ergebnisse im Seitenbereich sind der Status der Stack-Instances nach Abschluss des ausgewählten Vorgangs.

Um die aktuellen Stack-Instance-Details für einen Vorgang anzuzeigen:

1. Wählen Sie die Registerkarte Stack Instances (Stack-Instances) aus.

2. Filtern Sie nach der Last operation ID (ID des letzten Vorgangs). Die Ergebnisse sind der aktuelle Status und die Statusgründe des letzten Vorgangs zur Änderung der Instance. Sie können diesen Filter in Kombination mit AWS -Konto, AWS -Region, Detaillierter Status und Abweichungsstatus verwenden, um Ihre Suchergebnisse weiter zu verfeinern.

Verwenden Sie den AWS CLI

Um die Anzahl der ausgefallenen Stack-Instances zu ermitteln, rufen Sie `describe-stack-set-operation` oder `list-stack-set-operations` auf und sehen Sie unter `StatusDetails` nach.

```
aws cloudformation describe-stack-set-operation --stack-set-name ss1 \
    --operation-id 5550e62f-c822-4331-88fa-21c1d7bafc60
```

```
{
    "StackSetOperation": {
        "OperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60",
        "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
        "Action": "CREATE",
        "Status": "SUCCEEDED",
        "OperationPreferences": {
            "RegionOrder": [],
            "FailureToleranceCount": 10,
            "MaxConcurrentCount": 10
        },
        "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
        "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
        "CreationTimestamp": "2022-10-26T17:18:53.947000+00:00",
        "EndTimestamp": "2022-10-26T17:19:35.304000+00:00",
        "StatusDetails": {
            "FailedStackInstancesCount": 3
        }
    }
}
```

```
aws cloudformation list-stack-set-operations --stack-set-name ss1
```

```
{
    "Summaries": [
        {
```

```
            "OperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60",
            "Action": "CREATE",
            "Status": "SUCCEEDED",
            "CreationTimestamp": "2022-10-26T17:18:53.947000+00:00",
            "EndTimestamp": "2022-10-26T17:19:35.304000+00:00",
            "StatusDetails": {
                "FailedStackInstancesCount": 3
            },
            "OperationPreferences": {
                "RegionOrder": [],
                "FailureToleranceCount": 10,
                "MaxConcurrentCount": 10
            }
        }
    ]
}
```

Um einen historischen Überblick über einen bestimmten Vorgang zu erhalten, verwenden Sie `list-stack-set-operation-results`, um den Status und den Statusgrund für jede Stack-Instance nach Abschluss des Vorgangs anzuzeigen. Im folgenden Beispiel finden `Status` und `StatusReason`:

```
aws cloudformation list-stack-set-operation-results --stack-set-name ss1 \
  --operation-id 5550e62f-c822-4331-88fa-21c1d7bafc60 --
filters Name=OPERATION_RESULT_STATUS,Values=FAILED
```

```
{
    "Summaries": [
        {
            "Account": "123456789012",
            "Region": "us-west-2",
            "Status": "FAILED",
            "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'.",
            "AccountGateResult": {
                "Status": "SKIPPED",
                "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'."
            },
            "OrganizationalUnitId": ""
```

```
        },
        {
            "Account": "123456789012",
            "Region": "us-west-1",
            "Status": "FAILED",
            "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'.",
            "AccountGateResult": {
                "Status": "SKIPPED",
                "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'."
            },
            "OrganizationalUnitId": ""
        },
        {
            "Account": "123456789012",
            "Region": "us-east-1",
            "Status": "FAILED",
            "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'.",
            "AccountGateResult": {
                "Status": "SKIPPED",
                "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'."
            },
            "OrganizationalUnitId": ""
        }
    ]
}
```

Verwenden Sie `list-stack-instances` mit den `DETAILED_STATUS`- und `LAST_OPERATION_ID`-Filtern, um eine Liste der Stack-Instances abzurufen, die bei dem letzten Vorgang, bei dem versucht wurde, die Stack-Instance bereitzustellen, fehlgeschlagen sind. Sehen Sie sich das `--filters`-Flag im Beispiel mit `DETAILED_STATUS` und `LAST_OPERATION_ID` an:

```
aws cloudformation list-stack-instances --stack-set-name ss1 \
  --filters Name=DETAILED_STATUS,Values=FAILED Name=LAST_OPERATION_ID,Values=5550e62f-
c822-4331-88fa-21c1d7bafc60
```

```
{
    "Summaries": [
        {
            "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
            "Region": "us-east-1",
            "Account": "123456789012",
            "Status": "OUTDATED",
            "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'.",
            "StackInstanceStatus": {
                "DetailedStatus": "FAILED"
            },
            "OrganizationalUnitId": "",
            "DriftStatus": "NOT_CHECKED",
            "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
        },
        {
            "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
            "Region": "us-west-1",
            "Account": "123456789012",
            "Status": "OUTDATED",
            "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'.",
            "StackInstanceStatus": {
                "DetailedStatus": "FAILED"
            },
            "OrganizationalUnitId": "",
            "DriftStatus": "NOT_CHECKED",
            "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
        },
        {
            "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
            "Region": "us-west-2",
            "Account": "123456789012",
            "Status": "OUTDATED",
            "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'.",
            "StackInstanceStatus": {
                "DetailedStatus": "FAILED"
            },
```

```
            "OrganizationalUnitId": "",
            "DriftStatus": "NOT_CHECKED",
            "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
        }
    ]
}
```

Um nach der letzten Vorgangs-ID zum Ändern einer Stack-Instance zu suchen, rufen Sie mithilfe von `list-stack-instances` oder `describe-stack-instance` die `LastOperationId` ab:

```
aws cloudformation describe-stack-instance --stack-set-name ss1 \
  --stack-instance-account 123456789012 --stack-instance-region us-east-2
```

```
{
    "StackInstance": {
        "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
        "Region": "us-west-2",
        "Account": "123456789012",
        "ParameterOverrides": [],
        "Status": "OUTDATED",
        "StackInstanceStatus": {
            "DetailedStatus": "FAILED"
        },
        "StatusReason": "Account 123456789012 should have
 'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
 'AWSCloudFormationStackSetAdministrationRole'.",
        "OrganizationalUnitId": "",
        "DriftStatus": "NOT_CHECKED",
        "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
    }
}
```

Synchronisieren von Stacks mit Quellcode, der in einem Git-Repository gespeichert ist, mit Git Sync

Mit Git Sync kannst du deine CloudFormation Stacks mit Quellcodeverwaltung verwalten. Sie tun dies, indem Sie CloudFormation die Überwachung eines Git-Repositorys konfigurieren. Das Repository wird auf Änderungen an zwei Dateien überwacht:

- Eine CloudFormation Vorlagendatei, die einen Stack definiert
- Eine Stack-Bereitstellungsdatei, die Parameter zur Konfiguration des Stacks enthält

Mit Git Sync kannst du Pull-Requests und Versionsverfolgung verwenden, um deine CloudFormation Stacks von einem zentralen Ort aus zu konfigurieren, bereitzustellen und zu aktualisieren. Wenn du Änderungen an der Vorlage oder der Deployment-Datei festschreibst, CloudFormation wird der Stack automatisch aktualisiert. Wenn Sie Pull-Requests verwenden, CloudFormation können Sie einen Kommentar zu der Pull-Anfrage hinterlassen, in dem erklärt wird, welche Änderungen an Ihrem Stack vorgenommen werden, bevor Sie ihn tatsächlich aktualisieren. Sie müssen diese Funktion jedoch zuerst aktivieren.

Git Sync bietet eine Konsolenschnittstelle, mit der du eine Verbindung zu einem Repository herstellen, eine Stack-Deployment-Datei generieren, eine CloudFormation Vorlage aktualisieren und eine Pull-Anfrage an dein Repository senden kannst. Git Sync bietet auch ein Status-Dashboard, mit dem Sie aktive Git-Sync-Stack-Bereitstellungen überwachen, bearbeiten und Fehler darin beheben können. Auf Git Sync wird über die CloudFormation Konsole zugegriffen, wenn Sie einen Stack erstellen. Sie können auf Git Sync auch über zugreifen CodeConnections. Weitere Informationen finden Sie unter Arbeiten mit Synchronisierungskonfigurationen für verknüpfte Repositorys im Developer Tools Console-Benutzerhandbuch.

Git Sync unterstützt GitHub Enterprise GitHub- GitLab, Bitbucket - und GitLab selbstverwaltete Repositorys.

> **Note**
>
> Git Sync ist in den Regionen USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Nordkalifornien), USA West (Oregon), Kanada (Zentral), Asien-Pazifik (Mumbai), Asien-Pazifik (Tokio), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney),

Europa (Irland), Europa (London), Europa (Paris), Europa (Stockholm), Europa (Frankfurt), Europa (Mailand) und Südamerika (São Paulo) verfügbar.

Informationen zur Verwendung von Git Sync mit einer Strategie für mehrere Konten finden Sie im folgenden Blogbeitrag Verwenden Sie AWS CloudFormation Git Sync, um Ressourcen in Kundenkonten zu konfigurieren.

Themen

- So funktioniert Git Sync mit CloudFormation
- Voraussetzungen für die Synchronisierung von Stacks mit einem Git-Repository mithilfe von Git Sync
- Erstellen Sie mit Git Sync einen Stack aus dem Repository-Quellcode
- Ermöglicht CloudFormation das Posten einer Zusammenfassung der Stack-Änderungen in Pull-Requests
- Status-Dashboard für Git-Synchronisierung

So funktioniert Git Sync mit CloudFormation

In diesem Thema wird beschrieben, wie Git Sync funktioniert, und es werden die wichtigsten Konzepte vorgestellt, die für die Arbeit mit Git erforderlich sind.

Themen

- So funktioniert Git Sync
- Kommentare zu Pull-Anforderungen
- Stack-Bereitstellungsdatei
- CloudFormation Vorlagendatei
- Vorlagendefinitions-Repository

So funktioniert Git Sync

Um Git Sync zu verwenden, müssen Sie zunächst einen Git-Anbieter CloudFormation mit der Nutzung des CodeConnectionsDienstes verbinden. In den Verfahren in diesem Handbuch wird die Verbindung über die CodeConnections Konsole hergestellt. Alternativ können Sie die Verbindung mit dem herstellen AWS CLI. Sie können einen der folgenden Git-Anbieter verwenden:

- GitHub
- GitHub Unternehmen
- GitLab
- Bitbucket
- GitLabselbst verwaltet

Als Nächstes erstellen Sie eine CloudFormation Vorlage, die Ihren Stack definiert, und fügen ihn Ihrem Repository hinzu. Diese Vorlagendatei wird überwacht. CloudFormation aktualisiert den Stack automatisch, wenn Änderungen an ihm festgeschrieben werden.

In der CloudFormation Konsole erstellst du einen neuen Stack und wählst Sync from Git, um anzugeben, dass CloudFormation du Git Sync verwenden möchtest. Sie geben das Repository und den Branch an, den Sie überwachen CloudFormation möchten, und geben die CloudFormation Vorlage in Ihrem Repository an, die den Stack definiert.

Während der Konfiguration können Sie entweder Ihre eigene Stack-Bereitstellungsdatei aus Ihrem Repository bereitstellen oder Git Sync eine für Sie generieren lassen. Die Stack-Bereitstellungsdatei enthält Parameter und Werte, die die Ressourcen in Ihrem Stack konfigurieren. Diese Stack-Bereitstellungsdatei wird überwacht. CloudFormation aktualisiert den Stack automatisch, wenn Änderungen daran vorgenommen werden.

Git Sync erstellt eine Pull-Anfrage in deinem Repository, um deinen Stack mit der CloudFormation Vorlagendatei und der Stack-Deployment-Datei zu synchronisieren. Wenn Git Sync die Stack-Bereitstellungsdatei für Sie generiert, wird sie von Git Sync an Ihr Repository gesendet.

Anschließend fügst du die Pull-Anfrage mit deinem Repository zusammen, sodass CloudFormation der Stack bereitgestellt, mit deinen Bereitstellungsparametern konfiguriert wird und dein Repository auf Änderungen überwacht wird.

Ab diesem Zeitpunkt werden die Änderungen automatisch erkannt, wann immer Sie Änderungen an Ihrer Vorlagendatei oder Stack-Bereitstellungsdatei vornehmen und diese in Ihr Repository übernehmen. CloudFormation Wenn dein Team Pull-Requests verwendet, können deine Teammitglieder die Änderungen überprüfen und genehmigen, bevor sie bereitgestellt werden. Sobald die Pull-Anfrage akzeptiert wurde, CloudFormation werden Ihre Änderungen bereitgestellt.

Du kannst den Status deiner Git-Sync-Konfiguration für den Stack überwachen und dir in der CloudFormation Konsole eine Historie der auf den Stack angewendeten Commits ansehen. Die

Konsole bietet auch Tools zur Neukonfiguration der Git-Synchronisierung und zur Behebung von Problemen.

Kommentare zu Pull-Anforderungen

Du kannst festlegen, dass eine Zusammenfassung der Codeänderungen in Pull-Requests über den CodeConnections Service CloudFormation erstellt wird, indem du in der Konsole die Option Kommentar bei Pull-Requests aktivieren aktivierst. Durch die Bereitstellung einer Zusammenfassung der Änderungen in Pull Requests können Teammitglieder die Auswirkungen der vorgeschlagenen Änderungen leicht überprüfen und verstehen, bevor sie den Pull Request zusammenführen. Weitere Informationen finden Sie unter Ermöglicht CloudFormation das Posten einer Zusammenfassung der Stack-Änderungen in Pull-Requests.

Stack-Bereitstellungsdatei

Eine Stack-Bereitstellungsdatei ist eine Datei im JSON-Format (JavaScript Object Notation) oder YAML-Standardformat, die Parameter und Werte zur Verwaltung Ihres Stacks enthält. CloudFormation Sie wird auf Änderungen überwacht. Wenn Änderungen an der Datei in das Repository übernommen werden, wird der zugehörige Stack automatisch aktualisiert.

Die Stack-Bereitstellungsdatei enthält ein Schlüssel-Wert-Paar und zwei Wörterbücher:

- `template-file-path`

  Dies ist der vollständige Repository-Pfad für die CloudFormation Vorlagendatei. Die Vorlagendatei deklariert die Ressourcen für den CloudFormation Stack, der dieser Bereitstellungsdatei zugeordnet ist.
- `parameters`

  Das Parameterwörterbuch enthält Schlüssel-Wert-Paare, die die Ressourcen im Stack konfigurieren. Eine Stack-Bereitstellungsdatei kann bis zu 50 Parameter haben.
- `tags`

  Das Tags-Wörterbuch enthält optionale Schlüssel-Wert-Paare, mit denen Sie Ressourcen im Stack identifizieren und kategorisieren können. Eine Stack-Bereitstellungsdatei kann bis zu 50 Tags haben.

Sie können Ihre eigene Stack-Bereitstellungsdatei bereitstellen oder Git Sync eine für Sie erstellen lassen und automatisch eine Pull-Anforderung an Ihr Repository senden. Sie können die Parameter

und Tags verwalten, indem Sie die Stack-Bereitstellungsdatei bearbeiten und die Änderungen an das Repository übergeben.

Im Folgenden finden Sie ein Beispiel einer Git-Sync-Stack-Bereitstellungsdatei:

```
template-file-path: fargate-srvc/my-stack-template.yaml

parameters:
    image: public.ecr.aws/lts/nginx:latest
    task_size: x-small
    max_capacity: 5
    port: 8080
    env: production
tags:
    cost-center: '123456'
    org: 'AWS'
```

CloudFormation Vorlagendatei

Eine Vorlagendatei enthält eine Deklaration der AWS Ressourcen, aus denen ein CloudFormation Stapel besteht. Mit Git Sync wird die Vorlagendatei in Ihrem Git-Repository gespeichert und von der Stack-Bereitstellungsdatei referenziert. Sie können den Stack verwalten, indem Sie die Vorlagendatei bearbeiten und Änderungen an das Repository übertragen.

Weitere Informationen finden Sie unter Mit CloudFormation Vorlagen arbeiten.

Vorlagendefinitions-Repository

Das Template-Definition-Repository ist das Git-Repository, mit dem CloudFormation über Git Sync verknüpft ist. Das Repository wird auf Änderungen an der CloudFormation Vorlagen- und Stack-Bereitstellungsdatei überwacht. Wenn Sie Änderungen an der Datei vornehmen, wird der zugehörige Stack automatisch aktualisiert.

> **Important**
>
> Wenn Sie das Vorlagendefinition-Repository in der Git-Sync-Konsole konfigurieren, wählen Sie das richtige Repository und die richtige Verzweigung aus der Git-Verbindung aus. Git Sync überwacht nur das konfigurierte Repository und den Branch auf Änderungen an der CloudFormation Vorlage und der Stack-Deployment-Datei.

Voraussetzungen für die Synchronisierung von Stacks mit einem Git-Repository mithilfe von Git Sync

Bevor Sie einen CloudFormation Stack mit Ihrem Git-Repository synchronisieren, stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind.

Themen

- Git-Repository
- CloudFormation Vorlage
- Git-Sync-Dienstrolle
- IAM-Berechtigungen für Konsolenbenutzer

Git-Repository

Sie müssen ein Git-Repository auf einer der folgenden Plattformen gehostet haben.

- GitHub
- GitHub Unternehmen
- GitLab
- Bitbucket
- GitLabselbst verwaltet

Das Repository kann öffentlich oder privat sein. Sie müssen dieses Git-Repository über die CloudFormation Connections-Konsole verbinden.

CloudFormation Vorlage

Ihr Git-Repository muss eine CloudFormation Vorlagendatei enthalten, die in den Branch eingecheckt ist, den Sie mit Git Sync verbinden möchten. Auf diese Vorlage wird in der Stack-Deployment-Datei verwiesen.

Git-Sync-Dienstrolle

Git Sync erfordert eine IAM-Rolle. Du kannst wählen, ob bei der Konfiguration von Git Sync eine IAM-Rolle für deinen Stack erstellt werden soll, oder du kannst eine bestehende Rolle verwenden.

> **Note**
>
> Eine automatisch generierte IAM-Rolle wendet Berechtigungen nur auf den Stack an, für den die Rolle generiert wurde. Um eine automatisch generierte IAM-Rolle wiederzuverwenden, müssen Sie die Rolle für den neuen Stack bearbeiten.

Erforderliche Berechtigungen für die Git-Sync-Dienstrolle

Die IAM-Rolle, die Sie für Git Sync bereitstellen, erfordert die folgenden Berechtigungen.

- `cloudformation:CreateChangeSet`
- `cloudformation:DeleteChangeSet`
- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:ListChangeSets`
- `cloudformation:ValidateTemplate`
- `events:PutRule`
- `events:PutTargets`

> **Note**
>
> Die oben genannten erforderlichen Berechtigungen werden automatisch zu den von Git Sync generierten IAM-Rollen hinzugefügt.

Die folgende Beispiel-IAM-Rolle enthält die erforderlichen Berechtigungen für Git Sync.

JSON

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

```
            "Sid": "SyncToCloudFormation",
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateChangeSet",
                "cloudformation:DeleteChangeSet",
                "cloudformation:DescribeChangeSet",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStacks",
                "cloudformation:ExecuteChangeSet",
                "cloudformation:GetTemplate",
                "cloudformation:ListChangeSets",
                "cloudformation:ListStacks",
                "cloudformation:ValidateTemplate"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PolicyForManagedRules",
            "Effect": "Allow",
            "Action": [
                "events:PutRule",
                "events:PutTargets"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                "events:ManagedBy":
 ["cloudformation.sync.codeconnections.amazonaws.com"]
                }
            }
        },
        {
            "Sid": "PolicyForDescribingRule",
            "Effect": "Allow",
            "Action": "events:DescribeRule",
            "Resource": "*"
        }
    ]
}
```

Vertrauensrichtlinie

Sie müssen die folgende Vertrauensrichtlinie angeben, wenn Sie die Rolle zur Definition der Vertrauensbeziehung erstellen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CfnGitSyncTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.sync.codeconnections.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Wir empfehlen Ihnen, die Tasten `aws:SourceArn` und `aws:SourceAccount` Condition zu verwenden, um sich vor dem Problem mit dem verwirrten Stellvertreter zu schützen. Das Quellkonto ist Ihre Konto-ID und der Quell-ARN ist der ARN der Verbindung im CodeConnectionsDienst, mit der Sie eine Verbindung CloudFormation zu Ihrem Git-Repository herstellen können.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CfnGitSyncTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.sync.codeconnections.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
```

```
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/EXAMPLE64-8aad-4d5d-8878-dfcab0bc441f"
        }
      }
    }
  ]
}
```

Weitere Informationen zum Confused-Deputy-Problem finden Sie Serviceübergreifende Confused-Deputy-Prävention.

IAM-Berechtigungen für Konsolenbenutzer

Um Git Sync erfolgreich über die CloudFormation Konsole einzurichten, müssen Endbenutzern auch Berechtigungen über IAM erteilt werden.

Die folgenden `codeconnections` Berechtigungen sind erforderlich, um die Verbindung zu Ihrem Git Repository herzustellen und zu verwalten.

- `codeconnections:CreateRepositoryLink`
- `codeconnections:CreateSyncConfiguration`
- `codeconnections:DeleteRepositoryLink`
- `codeconnections:DeleteSyncConfiguration`
- `codeconnections:GetRepositoryLink`
- `codeconnections:GetSyncConfiguration`
- `codeconnections:ListRepositoryLinks`
- `codeconnections:ListSyncConfigurations`
- `codeconnections:ListTagsForResource`
- `codeconnections:TagResource`
- `codeconnections:UntagResource`
- `codeconnections:UpdateRepositoryLink`
- `codeconnections:UpdateSyncBlocker`
- `codeconnections:UpdateSyncConfiguration`

- `codeconnections:UseConnection`

Konsolenbenutzer müssen außerdem über die folgenden `cloudformation` Berechtigungen verfügen, um Stacks während des Git-Sync-Einrichtungsprozesses anzeigen und verwalten zu können.

- `cloudformation:CreateChangeSet`
- `cloudformation:DeleteChangeSet`
- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:GetTemplate`
- `cloudformation:ListChangeSets`
- `cloudformation:ListStacks`
- `cloudformation:ValidateTemplate`

> **Note**
>
> Für die ausschließliche Nutzung auf der Konsole sind Berechtigungen für Änderungssätze (`cloudformation:CreateChangeSetcloudformation:DeleteChangeSetcloudformation:D` möglicherweise nicht unbedingt erforderlich, es wird jedoch empfohlen, die Funktionen zur vollständigen Überprüfung und Verwaltung des Stacks zu aktivieren.

Die folgende Beispiel-IAM-Richtlinie umfasst die Benutzerberechtigungen, die für die Einrichtung von Git Sync über die Konsole erforderlich sind.

JSON

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

```
            "Effect": "Allow",
            "Action": [
                "codeconnections:CreateRepositoryLink",
                "codeconnections:CreateSyncConfiguration",
                "codeconnections:DeleteRepositoryLink",
                "codeconnections:DeleteSyncConfiguration",
                "codeconnections:GetRepositoryLink",
                "codeconnections:GetSyncConfiguration",
                "codeconnections:ListRepositoryLinks",
                "codeconnections:ListSyncConfigurations",
                "codeconnections:ListTagsForResource",
                "codeconnections:TagResource",
                "codeconnections:UntagResource",
                "codeconnections:UpdateRepositoryLink",
                "codeconnections:UpdateSyncBlocker",
                "codeconnections:UpdateSyncConfiguration",
                "codeconnections:UseConnection",
                "codeconnections:CreateForcedTargetSync",
                "codeconnections:CreatePullRequestForResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudFormationConsolePermissions",
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateChangeSet",
                "cloudformation:DeleteChangeSet",
                "cloudformation:DescribeChangeSet",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStacks",
                "cloudformation:ExecuteChangeSet",
                "cloudformation:GetTemplate",
                "cloudformation:ListChangeSets",
                "cloudformation:ListStacks",
                "cloudformation:ValidateTemplate"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Wenn Sie eine IAM-Richtlinie erstellen, die die Berechtigungen `codeconnections:CreateForcedTargetSync` und enthält`codeconnections:CreatePullRequestForResource`, wird in der IAM-Konsole möglicherweise eine Warnung angezeigt, dass diese Aktionen nicht existieren. Diese Warnung kann ignoriert werden, und die Richtlinie wird trotzdem erfolgreich erstellt. Diese Berechtigungen sind für bestimmte Git-Synchronisierungsvorgänge erforderlich, obwohl sie von der IAM-Konsole nicht erkannt werden.

Erstellen Sie mit Git Sync einen Stack aus dem Repository-Quellcode

In diesem Thema wird erklärt, wie Sie mit Git Sync einen CloudFormation Stack erstellen, der mit einem Git-Repository synchronisiert wird.

> **Important**
>
> Bevor Sie fortfahren, müssen Sie alle im vorherigen Abschnitt genannten Voraussetzungen erfüllen.

Erstellen Sie einen Stack aus dem Quellcode des Repositorys

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, in der der Stack erstellt werden AWS-Region soll.
3. Wählen Sie auf der Seite „Stapel“ die Option „Stapel erstellen“ und dann „Mit neuen Ressourcen (Standard)“ aus.
4. Gehen Sie auf der Seite Create stack (Stack erstellen) wie folgt vor:
   a. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorhandene Vorlage auswählen aus.
   b. Wählen Sie für Vorlage angeben die Option Sync from Git und dann Weiter aus.

5. Geben Sie auf der Seite „Stack-Details angeben“ unter Stack-Name einen Namen für Ihren Stack ein. Stack-Namen können Buchstaben (A–Z und a–z), Ziffern (0–9) und Bindestriche (-) enthalten.
6. Für die Stack-Bereitstellungsdatei, Erstellung der Bereitstellungsdatei:
   - Wenn Sie keine Stack-Deployment-Datei erstellt und zu Ihrem Repository hinzugefügt haben, wählen Sie Create the file using the following parameters und platzieren Sie sie in meinem Repository.
   - Wenn Sie eine Stack-Bereitstellungsdatei in Ihrem Repository haben, wählen Sie Ich stelle meine eigene Datei in meinem Repository bereit.
7. Wählen Sie für Repository mit Vorlagendefinitionen die Option Ein verknüpftes Git-Repository auswählen, um ein Git-Repository auszuwählen, mit dem bereits verknüpft ist CloudFormation, oder Ein Git-Repository verknüpfen, um ein neues zu verknüpfen. Wenn Sie Link a Git repository wählen, gehen Sie wie folgt vor:
   a. Wählen Sie unter Repository-Anbieter auswählen eine der folgenden Optionen aus:
      - GitHub
      - GitHub Unternehmensserver
      - GitLab
      - Bitbucket
      - GitLab selbst verwaltet
   b. Wählen Sie für Verbindung eine Verbindung aus der Liste aus. Wenn in der Verbindungsliste keine Optionen angezeigt werden, wählen Sie Neue Verbindung hinzufügen, um zur Verbindungskonsole zu wechseln und eine Verbindung zu Ihrem Repository herzustellen.
8. Wählen Sie in der Repository-Liste das Git-Repository aus, das Ihre Stack-Vorlagendatei enthält.
9. Wähle in der Liste Verzweigung die Verzweigung aus, die Git Sync überwachen soll.

> **Note**
>
> Git Sync überwacht nur den ausgewählten Branch auf Änderungen an den CloudFormation Vorlagen- und Stack-Bereitstellungsdateien. Alle Änderungen, die Sie auf Ihren Stack anwenden möchten, müssen für diese Verzweigung übernommen werden.

10. Geben Sie für den Pfad der Bereitstellungsdatei den vollständigen Pfad einschließlich des Namens der Stack-Bereitstellungsdatei aus dem Stammverzeichnis Ihrer Repository-Verzweigung an.

    Wenn die Datei für dich generiert CloudFormation wird, wird die Datei hier in deinem Repository festgeschrieben. Wenn Sie die Datei bereitstellen, ist dies der Speicherort der Datei in Ihrem Repository.

11. Fügen Sie eine IAM-Rolle hinzu. Die IAM-Rolle beinhaltet Berechtigungen, die für CloudFormation die Synchronisierung des Stacks aus Ihrem Git-Repository erforderlich sind. Du kannst Neue IAM-Rolle wählen, um eine neue Rolle zu generieren, oder Bestehende IAM-Rolle wählen, um eine bestehende Rolle aus deiner auszuwählen. AWS-Konto Wenn Sie sich dafür entscheiden, eine neue Rolle zu generieren, sind die erforderlichen Berechtigungen in der Rolle enthalten.

12. Kommentare bei Pull-Requests aktivieren oder deaktivieren:

    - Um Informationen nach der Änderung in Pull-Requests für Stack-Updates festzulegen, lassen Sie den Schalter Kommentar bei Pull-Request aktivieren aktiviert. CloudFormation
    - Wenn du diesen Schalter ausschaltest, CloudFormation werden die Unterschiede zwischen der aktuellen Stack-Konfiguration und den vorgeschlagenen Änderungen an Pull-Requests nicht beschrieben, wenn die Repo-Dateien aktualisiert werden.

13. Geben Sie für den Pfad der Vorlagendatei den vollständigen Pfad aus dem Stammverzeichnis Ihres Repositorys für die Stack-Vorlagendatei an.

14. (Optional) Um die Stack-Parameter anzugeben, wählen Sie Parameter hinzufügen, geben Sie einen Schlüssel und einen Wert für jeden Parameter ein und wählen Sie dann Weiter. Weitere Informationen finden Sie unter Stack-Bereitstellungsdatei.

    Gehen Sie beispielsweise wie folgt vor, um einen **`port=8080`** Parameter in Ihrer Stack-Bereitstellungsdatei anzugeben:

    a. Wählen Sie Parameter hinzufügen aus.

    b. Geben Sie für Key (Schlüssel) **`port`** ein.

    c. Geben Sie für Wert **`8080`** ein.

15. (Optional) Um Stack-Tags anzugeben, wählen Sie Neues Tag hinzufügen aus, geben Sie einen Tag-Schlüssel und einen Tag-Wert für jedes Tag ein und wählen Sie dann Weiter. Weitere Informationen finden Sie unter Stack-Bereitstellungsdatei.

16. Wählen Sie Weiter, um mit der Konfiguration der Stack-Optionen fortzufahren. Informationen zum Konfigurieren von Stack-Optionen finden Sie unter Konfigurieren Sie die Stack-Optionen.

    Wenn Sie Ihre Stack-Konfiguration abgeschlossen haben, wählen Sie Weiter, um fortzufahren.
17. Überprüfen Sie Ihre Stack-Einstellungen und bestätigen Sie Folgendes:
    - Die Stack-Vorlage ist korrekt konfiguriert und auf Von Git synchronisieren gesetzt.
    - Die Bereitstellungsdatei ist korrekt konfiguriert.
    - Das Repository mit der Vorlagendefinition ist korrekt konfiguriert, insbesondere, das richtige Repository und der richtige Verzweigungsname wurden ausgewählt.
    - Die Vorschau der Bereitstellungsdatei ist korrekt und enthält die erwarteten Parameter und Werte.
18. Wählen Sie Absenden, um den Stack zu erstellen.

    Nachdem du Submit ausgewählt hast, wird automatisch eine Pull-Anfrage in deinem Git-Repository erstellt. Du musst diesen Pull-Request mit deinem Git-Repository zusammenführen, um deinen Stack zu erstellen. CloudFormation Überwacht dein Git-Repository nach der Erstellung des Stacks auf Änderungen.

Aktualisiere deinen Stack aus deinem Git-Repository

Um den Stack zu aktualisieren, nimm Änderungen direkt an deiner Vorlagendatei oder Stack-Bereitstellungsdatei in deinem Git-Repository vor. Nachdem Sie Ihre Änderungen in den überwachten Branch übernommen haben, CloudFormation wird der Stack automatisch aktualisiert. Wenn du Pull-Requests verwendest, wird automatisch eine Pull-Anfrage in deinem Git-Repository erstellt, bevor der Stack aktualisiert wird. Du musst diesen Pull-Request mit deinem Git-Repository zusammenführen, um deinen Stack zu aktualisieren.

In der CloudFormation Konsole kannst du den Stack auswählen und den Tab Git Sync wählen, um Informationen zum Status des Stacks und zu Synchronisierungsereignissen anzuzeigen. Weitere Informationen finden Sie unter Status-Dashboard für Git-Synchronisierung.

Ermöglicht CloudFormation das Posten einer Zusammenfassung der Stack-Änderungen in Pull-Requests

In diesem Thema erfahren Sie, wie Sie das Posten einer Zusammenfassung der Stack-Änderungen in Pull-Requests in Ihrem Git-Repository aktivieren CloudFormation .

Indem Sie die Funktion „Kommentare CloudFormation zu Pull-Anfragen“ aktivieren, können Sie einen Kommentar posten, der die Unterschiede zwischen der aktuellen Stack-Konfiguration und den vorgeschlagenen Änderungen beschreibt, wenn die Repo-Dateien aktualisiert werden. Dieser Kommentar bietet eine Zusammenfassung der Ressourcen, die hinzugefügt, geändert oder gelöscht werden, sodass Sie vor dem Zusammenführen der Pull-Anfrage eine gründliche Codeüberprüfung durchführen können.

Um Kommentare zu Pull-Requests für einen neuen Stack (Konsole) zu aktivieren

Wenn Sie den Stack erstellen, stellen Sie auf der Seite Stack-Details angeben unter Repository für Vorlagendefinitionen sicher, dass die Option Kommentar bei Pull-Request aktivieren aktiviert ist. Dies ist die Standardeinstellung für neue Stacks.

Um Kommentare zu Pull-Requests für einen vorhandenen Stack (Konsole) zu aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, in der AWS-Region Sie Ihren Stack erstellt haben.
3. Wählen Sie auf der Seite Stacks den laufenden Stack aus, den Sie aktualisieren möchten.
4. Wähle den Tab Git Sync und dann Bearbeiten.
5. Aktiviere auf der Seite Git-Sync-Einstellungen bearbeiten unter Template-Definition-Repository die Option Kommentar bei Pull-Anfrage aktivieren.
6. Wählen Sie Update configuration (Konfiguration aktualisieren).

Status-Dashboard für Git-Synchronisierung

Um den Status einer AWS CloudFormation Git-Sync-Bereitstellung anzuzeigen, wählen Sie den Stack in der CloudFormation Konsole aus und wählen Sie den Tab Git Sync.

Der Git-Sync-Registerkarte ist in zwei Bereiche unterteilt: Git-Sync-Status und Letzte Synchronisierungsereignisse.

Git-Sync-Status

Im oberen Bereich finden Sie die folgenden Informationen zur Git-Sync-Konfiguration für den Stack.

Repository

Ein Link zum Repository, das mit Git Sync verbunden ist

Repository-Anbieter

Der Name des Repository-Anbieters.

Verzweigung

Der Name der Verzweigung, die Git Sync überwacht

Pfad der Bereitstellungsdatei

Der vollständige Pfad zur Stack-Bereitstellungsdatei für diesen Stack

Synchronisierungsstatus des Repository

Der Status des letzten Synchronisierungsvorgangs

Statusmeldung zur Repository-Synchronisierung

Die Statusmeldung des letzten Synchronisierungsvorgangs

Git-Sync-Status

Der Status der Git-Synchronisierung für diesen Stack

Bereitstellungsstatus

Der Status des Bereitstellungsvorgangs

Verwenden Sie in der oberen rechten Ecke des Bereics die folgenden Schaltflächen, um Git Sync zu ändern oder zu aktualisieren:

- Bearbeiten – Bearbeiten Sie die Git-Sync-Konfiguration.
- Letztes Commit wiederholen – Aktualisieren Sie den Stack entsprechend dem letzten Commit im Repository.
- Verbindung trennen – Trennen Sie Git Sync vom Stack.

- Aktualisieren – Aktualisieren Sie das Status-Menü von Git Sync.

Letzte Synchronisierungsereignisse

Im Bereich Letzte Synchronisierungsereignisse wird eine Tabelle mit Commits angezeigt, die auf den Stack angewendet wurden.

Sie können die Tabelle mithilfe der Pfeile in der Kopfzeile jeder Spalte sortieren. Die Tabelle kann wie folgt in aufsteigender oder absteigender Reihenfolge sortiert werden:

- Date (Datum)
- Commit-ID
- Veranstaltung
- Date (Datum)
- Ereignistyp

Unterstützte Stack-Status

Git Sync kann nur für einen Stack konfiguriert werden, wenn sich der Stack in einem der folgenden unterstützten Status befindet:

- `CREATE_COMPLETE`
- `UPDATE_COMPLETE`
- `UPDATE_ROLLBACK_COMPLETE`
- `IMPORT_COMPLETE`
- `IMPORT_ROLLBACK_COMPLETE`

Die folgende Tabelle enthält eine vollständige Liste der Stack-Statuscodes mit Beschreibungen:

Stack-Status und optional detaillierter Status	Beschreibung
`CREATE_COMPLETE`	Erfolgreiche Erstellung eines oder mehrerer Stacks.
`CREATE_IN_PROGRESS`	Laufende Erstellung eines oder mehrerer Stacks.

Stack-Status und optional detaillierter Status	Beschreibung
`CREATE_FAILED`	Fehlgeschlagene Erstellung eines oder mehrerer Stacks. Untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen. Mögliche Gründe für eine fehlgesch lagene Erstellung sind unzureichende Berechtigungen für die Arbeit mit allen Ressourcen im Stack, von einem AWS Dienst abgelehnte Parameterwerte oder ein Timeout bei der Ressourcenerstellung.
`DELETE_COMPLETE`	Erfolgreiche Löschung eines oder mehrerer Stacks. Gelöschte Stacks bleiben 90 Tage lang erhalten und anzeigbar.
`DELETE_FAILED`	Fehlgeschlagene Löschung eines oder mehrerer Stacks. Da der Löschvorgang fehlgeschlagen ist, werden möglicherweise einige Ressourcen noch ausgeführt. Sie können jedoch nicht mit dem Stack arbeiten oder ihn aktualisieren. Löschen Sie den Stack erneut oder untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen.
`DELETE_IN_PROGRESS`	Laufende Entfernung eines oder mehrerer Stacks.
`REVIEW_IN_PROGRESS`	Laufende Erstellung eines oder mehrerer Stacks mit einer erwarteten `StackId`, aber ohne Vorlagen oder Ressourcen. **Important** Ein Stack mit diesem Statuscode wird auf die maximal mögliche Anzahl von Stacks angerechnet.

Stack-Status und optional detaillierter Status	Beschreibung
`ROLLBACK_COMPLETE`	Erfolgreiche Entfernung eines oder mehrerer Stacks nach einer fehlgeschlagenen Stack-Erstellung oder nach einer explizit abgebrochenen Stack-Erstellung. Der Stack wird in einen früheren funktionsfähigen Zustand zurückgesetzt. Alle Ressourcen, die während des Create Stack-Vorgangs erstellt wurden, werden gelöscht. Dieser Status existiert nur, nachdem das Erstellen eines Stacks fehlgeschlagen ist. Es bedeutet, dass alle Vorgänge aus dem teilweise erstellten Stack entsprechend bereinigt wurden. In diesem Status kann nur einen Löschbefehl ausgeführt werden.
`ROLLBACK_FAILED`	Fehlgeschlagene Entfernung eines oder mehrerer Stacks nach einer fehlgeschlagenen Stack-Erstellung oder nach einer explizit abgebrochenen Stack-Erstellung. Löschen Sie den Stack oder untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen.
`ROLLBACK_IN_PROGRESS`	Laufende Entfernung eines oder mehrerer Stacks nach einer fehlgeschlagenen Stack-Erstellung oder nach einer explizit abgebrochenen Stack-Erstellung.
`UPDATE_COMPLETE`	Erfolgreiche Aktualisierung eines oder mehrerer Stacks.
`UPDATE_COMPLETE_CLEANUP_IN_PROGRESS`	Laufende Entfernung alter Ressourcen für einen oder mehrere Stacks nach einem erfolgreichen Stack-Update. Bei Stack-Updates, bei denen Ressourcen ersetzt werden müssen, CloudFormation erstellt zuerst die neuen Ressourcen und löscht dann die alten Ressourcen, um Unterbrechungen im Stack zu vermeiden. In diesem Zustand wurde der Stack aktualisiert und ist nutzbar, löscht aber CloudFormation immer noch die alten Ressourcen.

Stack-Status und optional detaillierter Status	Beschreibung
`UPDATE_FAILED`	Nicht erfolgreiche Aktualisierung eines oder mehrerer Stacks. Untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen.
`UPDATE_IN_PROGRESS`	Laufende Aktualisierung eines oder mehrerer Stacks.
`UPDATE_ROLLBACK_COMPLETE`	Erfolgreiches Zurücksetzen eines oder mehrerer Stacks in einen früheren funktionsfähigen Zustand nach einer fehlgeschlagenen Stack-Aktualisierung.
`UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS`	Laufende Entfernung neuer Ressourcen für einen oder mehrere Stacks nach einem fehlgeschlagenen Stack-Update. In diesem Zustand wurde der Stack in seinen vorherigen Betriebszustand zurückgesetzt und ist nutzbar, löscht aber CloudFormation immer noch alle neuen Ressourcen, die er während der Stack-Aktualisierung erstellt hat.
`UPDATE_ROLLBACK_FAILED`	Fehlgeschlagenes Zurücksetzen eines oder mehrerer Stacks in einen früheren funktionsfähigen Zustand nach einer fehlgeschlagenen Stack-Aktualisierung. In diesem Status können Sie den Stack löschen oder das Rollback fortsetzen. Möglicherweise müssen Sie Fehler beseitigen, um den Stack wieder in einen arbeitsfähigen Zustand zu versetzen. Oder Sie können Kontakt aufnehmen, Support um den Stack wieder in einen verwendbaren Zustand zu versetzen.
`UPDATE_ROLLBACK_IN_PROGRESS`	Laufendes Zurücksetzen eines oder mehrerer Stacks in den früheren funktionsfähigen Zustand nach einer fehlgeschlagenen Stack-Aktualisierung.
`IMPORT_IN_PROGRESS`	Der Importvorgang wird derzeit ausgeführt.
`IMPORT_COMPLETE`	Der Importvorgang wurde für alle Ressourcen im Stack erfolgreich abgeschlossen, die `resource import` unterstützen.

Stack-Status und optional detaillierter Status	Beschreibung
`IMPORT_ROLLBACK_IN_PROGRESS`	Der Import wird zur vorherigen Vorlagenkonfiguration zurückgesetzt.
`IMPORT_ROLLBACK_FAILED`	Der Import-Rollback-Vorgang ist für mindestens eine Ressource im Stack fehlgeschlagen. Die Ergebnisse werden für die CloudFormation erfolgreich importierten Ressourcen verfügbar sein.
`IMPORT_ROLLBACK_COMPLETE`	Der Import wurde erfolgreich auf die vorherige Vorlagenk onfiguration zurückgesetzt.

Erweiterungen mit der CloudFormation Registry verwalten

Die AWS CloudFormation Registrierung dient als zentraler Knotenpunkt für die Verwaltung von Erweiterungen, die in die CloudFormation Vorlagen in Ihrem integriert werden können AWS-Konto. Zu den Erweiterungen gehören Ressourcentypen, Module und Hooks von AWS Herausgebern und Drittanbietern sowie Ihre eigenen benutzerdefinierten Erweiterungen. Die Registrierung erleichtert das Auffinden und Bereitstellen von Erweiterungen in Ihren CloudFormation Vorlagen auf die gleiche Weise, wie Sie AWS bereitgestellte Ressourcen verwenden.

In diesem Abschnitt wird beschrieben, wie Sie mithilfe der CloudFormation Registrierung Erweiterungen von Drittanbietern in Ihrem Konto aktivieren, darunter:

- Aktivieren von öffentlichen Erweiterungen
- Registrierung und Aktivierung von privaten Erweiterungen

Themen

- Verwandte Dokumentation
- CloudFormation Konzepte der Registrierung
- Sehen Sie sich die verfügbaren und aktivierten Erweiterungen in der CloudFormation Registrierung an
- Verwenden Sie öffentliche Erweiterungen von Drittanbietern aus der CloudFormation Registrierung
- Verwenden Sie private Erweiterungen von Drittanbietern, die mit Ihnen geteilt wurden
- Bearbeiten Sie die Konfigurationsdaten für Erweiterungen in Ihrem Konto
- Ressourcentypen aufzeichnen in AWS Config

Verwandte Dokumentation

Wenn Sie ein Entwickler sind, der daran interessiert ist, eigene Erweiterungen zu erstellen, lesen Sie die folgende Dokumentation:

- Entwickeln von Modulen mit der CloudFormation CLI im Benutzerhandbuch für die CloudFormation Befehlszeilenschnittstelle
- Erstellen von Ressourcentypen mit der CloudFormation CLI im CloudFormationCommand Line Interface User Guide

- Entwicklung benutzerdefinierter Hooks mit der CloudFormation CLI im AWS CloudFormation Hooks User Guide

Darüber hinaus können alle in der CloudFormation Registrierung verfügbaren bereitstellbaren AWS Ressourcentypen mit dem verwendet werden AWS -Cloud-Control- API, wobei ihre Attribute und Eigenschaften in einem Standard-JSON-Schema definiert sind. Weitere Informationen finden Sie im Cloud Control API-Benutzerhandbuch. Wenn Sie die Cloud Control API verwenden, um CRUDL-Operationen (Create, Read, Update, Delete, List) für AWS Ressourcen durchzuführen, können Sie dies nur für Ressourcen tun, die sich in Ihren eigenen AWS AWS-Konto Ressourcen befinden.

CloudFormation Konzepte der Registrierung

In diesem Thema werden die wichtigsten Konzepte erläutert, die Ihnen helfen sollen, die CloudFormation Registrierung zu verstehen und damit zu beginnen, sie zu verwenden.

Arten von Erweiterungen

Die CloudFormation Registrierung bietet die folgenden Erweiterungstypen:

Haken

Hooks sind Validierungsprüfungen, die deine Stacks oder bestimmte Ressourcen überprüfen, bevor sie CloudFormation erstellt, aktualisiert oder gelöscht werden. Hooks können auch bei Vorgängen zum Erstellen von Änderungssätzen aufgerufen werden. Sie bieten einen Mechanismus zur Durchsetzung organisatorischer Standards und bewährter Verfahren, indem sie Ressourcenkonfigurationen anhand bestimmter Anforderungen validieren. Wenn ein Hook Konfigurationen erkennt, die seiner Logik nicht entsprechen, kann er entweder eine Warnung ausgeben oder den Bereitstellungsprozess fehlschlagen, um zu verhindern, dass nicht konforme Ressourcen bereitgestellt werden. Weitere Informationen finden Sie im AWS CloudFormation Hooks User Guide.

Spezifische Dokumentation zu Guard- und Lambda-Hooks finden Sie in den folgenden Abschnitten des AWS CloudFormation Hooks User Guide:

- Guard Hooks
- Lambda-Haken

Module

Module sind wiederverwendbare Ressourcenkonfigurationen, die in mehreren CloudFormation Stack-Vorlagen enthalten sein können. Sie vereinfachen die Erstellung und Pflege von CloudFormation Vorlagen, indem sie komplexe oder häufig verwendete Ressourcenkonfigurationen in wiederverwendbare Komponenten kapseln. Dies fördert die Konsistenz und Standardisierung aller Infrastrukturbereitstellungen Ihres Unternehmens.

Ressourcentypen

Mit Ressourcentypen können Sie Ressourcen von Drittanbietern oder benutzerdefinierte Ressourcen modellieren und automatisieren, die nicht nativ von unterstützt werden. CloudFormation Durch die Entwicklung von Ressourcentypen können Sie die Funktionen zur Bereitstellung und Verwaltung von Ressourcen aus verschiedenen Drittanbieterdiensten erweitern CloudFormation.

Öffentliche Erweiterungen

Öffentliche Erweiterungen sind CloudFormation Erweiterungen, die öffentlich in der Registrierung veröffentlicht werden und von allen verwendet werden können AWS-Konten. Dazu zählen:

- AWS Erweiterungen — Erweiterungen, die von veröffentlicht werden, AWS sind immer öffentlich und standardmäßig aktiviert, sodass Sie keine Maßnahmen ergreifen müssen, bevor Sie sie in Ihrem Konto verwenden. AWS steuert die Versionierung dieser Erweiterungen, sodass Sie immer die neueste verfügbare Version verwenden.
- Erweiterungen von Drittanbietern — Diese Erweiterungen werden von anderen Herausgebern zur allgemeinen Verwendung zur Verfügung gestellt als AWS. Um eine öffentliche Erweiterung eines Drittanbieters verwenden zu können, müssen Sie sie zunächst in Ihrem Konto und in Ihrer Region aktivieren.

Aktivierte Erweiterungen

Die CloudFormation Registrierung in Ihrem speziellen Verzeichnis AWS-Konto umfasst drei Arten von aktivierten Erweiterungen:

- AWS Erweiterungen — Alle AWS öffentlichen Erweiterungen werden automatisch aktiviert.
- Aktivierte Drittanbieter — Dies sind lokale Kopien von öffentlichen Erweiterungen von Drittanbietern, die Sie ausdrücklich für Ihr Konto und Ihre Region aktiviert haben. Wenn Sie eine

öffentliche Erweiterung eines Drittanbieters aktivieren, CloudFormation wird eine lokale Kopie in der Registrierung Ihres Kontos erstellt. Weitere Informationen finden Sie unter Verwenden Sie öffentliche Erweiterungen von Drittanbietern aus der CloudFormation Registrierung.

- Privat registriert — Dies sind private Erweiterungen, die nicht im öffentlichen CloudFormation Register aufgeführt sind. Dabei kann es sich um Erweiterungen handeln, die Sie selbst erstellt haben, oder um Erweiterungen, die von Ihrer Organisation oder anderen Dritten mit Ihnen geteilt wurden. Um eine solche private Erweiterung in Ihrem Konto zu verwenden, müssen Sie sie zuerst registrieren. Durch die Registrierung der Erweiterung wird eine Kopie in die CloudFormation Registrierung in Ihrem Konto hochgeladen und aktiviert. Weitere Informationen finden Sie unter Verwenden Sie private Erweiterungen von Drittanbietern, die mit Ihnen geteilt wurden.

Die Verwendung privat registrierter Erweiterungen und aktivierter öffentlicher Erweiterungen von Drittanbietern in Ihrem Konto entspricht der Verwendung in einer Sandbox-Umgebung. Erweiterungen werden mit Versionskontrolle verwaltet, sodass ihr Bereitstellungsverhalten an eine bestimmte Version gebunden ist. Daher funktionieren diese Erweiterungen genauso wie öffentliche Erweiterungen und halten sich an dieselben versionsspezifischen Regeln.

> **Note**
>
> Privat registrierte Erweiterungen und aktivierte öffentliche Erweiterungen von Drittanbietern können Event-Handler implementieren, die bei Erstellungs-, Lese-, Aktualisierungs-, Listen- und Löschvorgängen ausgeführt werden. Wenn Sie diese Erweiterungen in Ihren CloudFormation Stacks verwenden, können zusätzlich zu den Gebühren für die erstellten Ressourcen Gebühren für Ihr Konto anfallen. Weitere Informationen finden Sie unter AWS CloudFormation Preise.

Sehen Sie sich die verfügbaren und aktivierten Erweiterungen in der CloudFormation Registrierung an

Um die verfügbaren und aktivierten Erweiterungen in der CloudFormation Registrierung anzuzeigen, können Sie die AWS Management Console oder die verwenden AWS CLI.

Um die verfügbaren und aktivierten Erweiterungen anzuzeigen (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie im Navigationsbereich unter Registrierung aus, welche Erweiterungskategorie Sie anzeigen möchten:
   - Öffentliche Erweiterungen zeigt die öffentlichen Erweiterungen an, die in Ihrem Konto verfügbar sind.
     1. Wählen Sie unter Filter, Erweiterungstyp Ihren Erweiterungstyp aus: Ressourcentypen, Module oder Hooks.
     2. Wählen Sie für Filter, Herausgeber einen Herausgeber aus: AWSoder Drittanbieter.
   - Aktivierte Erweiterungen zeigt die Erweiterungen an, die in Ihrem Konto aktiviert sind.
     1. Wählen Sie Ihren Erweiterungstyp aus: Ressourcentypen,Module, oder Hooks-Erweiterungen.
     2. Verwenden Sie das Dropdownmenü Filter, um die anzuzeigenden Erweiterungen weiter auszuwählen:
        - AWS— listet Erweiterungen auf, die von veröffentlicht wurden AWS. Erweiterungen, die von veröffentlicht wurden, AWS sind standardmäßig aktiviert.
        - Drittanbieter — listet alle öffentlichen Erweiterungen von anderen Herausgebern auf AWS , als denen, die Sie in diesem Konto aktiviert haben.
        - Registriet – listet alle privaten Erweiterungen auf, die Sie in diesem Konto aktiviert haben.
   - Publisher zeigt alle öffentlichen Erweiterungen an, die Sie mit diesem Konto veröffentlicht haben.
4. Suchen Sie nach dem Namen der Erweiterung oder wählen Sie ihn aus, um die Details der Erweiterung anzuzeigen.

Um die verfügbaren und aktivierten Erweiterungen anzuzeigen (AWS CLI)

Verwenden Sie den list-types-Befehl.

Verwenden Sie öffentliche Erweiterungen von Drittanbietern aus der CloudFormation Registrierung

Um eine öffentliche Erweiterung eines Drittanbieters in Ihrer Vorlage zu verwenden, müssen Sie zuerst die Erweiterung für das Konto und die Region aktivieren, in der Sie sie verwenden möchten.

Wenn Sie eine Erweiterung aktivieren, kann sie für Stack-Operationen in dem Konto und der Region verwendet werden, in der sie aktiviert ist.

Wenn Sie eine öffentliche Erweiterung eines Drittanbieters aktivieren, CloudFormation wird in der Erweiterungsregistrierung Ihres Kontos ein Eintrag für die aktivierte Erweiterung als private Erweiterung erstellt. Auf diese Weise können Sie alle Konfigurationseigenschaften festlegen, die die Erweiterung enthält. Die Konfigurationseigenschaften definieren, wie die Erweiterung für eine bestimmte AWS-Konto Region konfiguriert wird.

Neben dem Festlegen der Konfigurationseigenschaften können Sie die Erweiterung auch auf folgende Weise anpassen:

- Geben Sie die Ausführungsrolle an, die zur Aktivierung der Erweiterung CloudFormation verwendet wird, und konfigurieren Sie zusätzlich die Protokollierung für die Erweiterung.
- Geben Sie an, ob die Erweiterung automatisch aktualisiert wird, wenn eine neue Nebenversion oder Patch-Version verfügbar wird.
- Geben Sie einen Alias an, der anstelle des Namens der öffentlichen Erweiterung eines Drittanbieters verwendet werden soll. Dadurch können Namenskollisionen zwischen Erweiterungen von Drittanbietern vermieden werden.

Themen

- Konfigurieren Sie eine Ausführungsrolle mit IAM-Berechtigungen und einer Vertrauensrichtlinie für den Zugriff auf öffentliche Erweiterungen
- Automatisch neue Versionen von Erweiterungen verwenden
- Verwenden Sie Aliase, um auf Erweiterungen zu verweisen
- Häufig verwendete AWS CLI Befehle für die Arbeit mit öffentlichen Erweiterungen
- Aktiviere eine öffentliche Erweiterung eines Drittanbieters in deinem Konto
- Aktualisieren Sie eine öffentliche Drittanbieter-Erweiterung in Ihrem Konto
- Deaktivieren Sie öffentliche Erweiterungen von Drittanbietern in Ihrem Konto

Konfigurieren Sie eine Ausführungsrolle mit IAM-Berechtigungen und einer Vertrauensrichtlinie für den Zugriff auf öffentliche Erweiterungen

Wenn Sie eine öffentliche Erweiterung in der CloudFormation Registrierung aktivieren, können Sie eine Ausführungsrolle angeben, die CloudFormation die erforderlichen Berechtigungen zum Aufrufen dieser Erweiterung in Ihrer Region und Ihrer AWS-Konto Region erteilt.

Die für die Ausführungsrolle erforderlichen Berechtigungen sind im Handler-Abschnitt des Erweiterungsschemas definiert. Sie müssen eine IAM-Richtlinie erstellen, die die spezifischen Berechtigungen gewährt, die für die Erweiterung benötigt werden, und sie der Ausführungsrolle zuordnen.

Zusätzlich zur Berechtigungsrichtlinie muss die Ausführungsrolle auch über eine Vertrauensrichtlinie verfügen, die es ermöglicht, die Rolle CloudFormation zu übernehmen. Folgen Sie den Anweisungen unter Erstellen einer Rolle mithilfe benutzerdefinierter Vertrauensrichtlinien im IAM-Benutzerhandbuch, um eine Rolle mit einer benutzerdefinierten Vertrauensrichtlinie zu erstellen.

Vertrauensstellung

Im Folgenden finden Sie Beispiele für Vertrauensrichtlinien, die Sie verwenden können.

Sie können optional den Umfang der Berechtigung für die dienstübergreifende Verhinderung von verwirrten Stellvertretern einschränken, indem Sie einen oder mehrere globale Bedingungskontextschlüssel für das `Condition` Feld verwenden. Weitere Informationen finden Sie unter Serviceübergreifende Confused-Deputy-Prävention.

- Legen Sie den Wert `aws:SourceAccount` auf Ihre Konto-ID fest.
- Setzen Sie den `aws:SourceArn` Wert auf den ARN Ihrer Erweiterung.

Beispiel für Vertrauensrichtlinie 1

Im Folgenden finden Sie ein Beispiel für eine Vertrauensrichtlinie für eine IAM-Rolle für eine Ressourcentyperweiterung.

JSON

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "resources."
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "StringLike": {
                    "aws:SourceArn": "arn:aws:cloudformation:us-
west-2:123456789012:type/resource/Organization-Service-Resource/*"
                }
            }
        }
    ]
}
```

Beispiel für eine Vertrauensrichtlinie 2

Im Folgenden finden Sie ein Beispiel für eine Vertrauensrichtlinie für eine IAM-Rolle für eine Hook-Erweiterung.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "resources.",
                    "hooks."
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
```

```
                },
                "StringLike": {
                    "aws:SourceArn": "arn:aws:cloudformation:us-
west-2:123456789012:type/hook/Organization-Service-Hook/*"
                }
            }
        }
    ]
}
```

Automatisch neue Versionen von Erweiterungen verwenden

Wenn Sie eine Erweiterung aktivieren, können Sie auch den Erweiterungstyp angeben, um die neueste Nebenversion zu verwenden. Ihr Erweiterungstyp aktualisiert die Nebenversion, wenn der Publisher eine neue Version für Ihre aktivierte Erweiterung veröffentlicht.

Wenn Sie beispielsweise das nächste Mal einen Stack-Vorgang ausführen, z. B. einen Stack erstellen oder aktualisieren, und dabei eine Vorlage verwenden, die diese Erweiterung enthält, CloudFormation wird die neue Nebenversion verwendet.

Die Aktualisierung auf eine neue Erweiterungs-Version, entweder automatisch oder manuell, wirkt sich nicht auf Erweiterungs-Instances aus, die bereits in Stacks bereitgestellt wurden.

CloudFormation behandelt Hauptversionsupdates von Erweiterungen so, dass sie potenziell wichtige Änderungen enthalten, und erfordert daher, dass Sie manuell auf eine neue Hauptversion einer Erweiterung aktualisieren.

Erweiterungen, AWS die von veröffentlicht wurden, sind standardmäßig für alle Konten und Regionen aktiviert, in denen sie verfügbar sind, und verwenden immer die neueste Version, die in jeder Version verfügbar ist AWS-Region.

> **Important**
>
> Da Sie kontrollieren, ob und wann Erweiterungen in Ihrem Konto auf die neueste Version aktualisiert werden, kann es sein, dass unterschiedliche Versionen derselben Erweiterung in verschiedenen Konten und Regionen bereitgestellt werden.
> Dies kann möglicherweise zu unerwarteten Ergebnissen führen, wenn dieselbe Vorlage, die diese Erweiterung enthält, für alle Konten und Regionen verwendet wird.

Verwenden Sie Aliase, um auf Erweiterungen zu verweisen

Sie können nicht mehr als eine Erweiterung mit einem bestimmten Namen in einer bestimmten AWS-Konto Region aktivieren. Da verschiedene Herausgeber möglicherweise öffentliche Erweiterungen mit demselben Erweiterungsnamen anbieten, CloudFormation können Sie einen Alias für jede öffentliche Erweiterung eines Drittanbieters angeben, die Sie aktivieren.

Wenn Sie einen Alias für die Erweiterung angeben, wird der Alias als Name des Erweiterungstyps innerhalb des Kontos und der Region CloudFormation behandelt. Sie müssen den Alias verwenden, um in Ihren Vorlagen, API-Aufrufen und in der CloudFormation Konsole auf die Erweiterung zu verweisen.

Ein Erweiterungs-Alias muss innerhalb eines bestimmten Kontos und einer bestimmten Region eindeutig sein. Sie können dieselbe öffentliche Ressource mehrmals in demselben Konto und derselben Region aktivieren, indem Sie Aliase für unterschiedliche Typnamen verwenden.

> **Important**
>
> Die Aliase für Erweiterungen müssen zwar nur für ein bestimmtes Konto und eine bestimmte Region eindeutig sein, wir empfehlen jedoch dringend, dass Benutzer nicht denselben Alias verschiedenen öffentlichen Erweiterungen von Drittanbietern für Konten und Regionen zuweisen. Dies könnte zu unerwarteten Ergebnissen führen, wenn eine Vorlage verwendet wird, die den Alias der Erweiterung für mehrere Konten oder Regionen enthält.

Häufig verwendete AWS CLI Befehle für die Arbeit mit öffentlichen Erweiterungen

Zu den häufig verwendeten Befehlen für die Arbeit mit öffentlichen Erweiterungen gehören:

- activate-typeum ein öffentliches Modul oder einen Ressourcentyp eines öffentlichen Drittanbieters in Ihrem Konto zu aktivieren.
- set-type-configurationum die Konfigurationsdaten für eine Erweiterung in Ihrem Konto anzugeben und Hooks zu deaktivieren und zu aktivieren.
- list-typesum die Erweiterungen in Ihrem Konto aufzulisten.
- describe-typeum detaillierte Informationen über eine bestimmte Erweiterung oder eine bestimmte Erweiterungsversion, einschließlich aktueller Konfigurationsdaten, zurückzugeben.

- set-type-default-versionum anzugeben, welche Version einer Erweiterung die Standardversion ist.
- deactivate-typeum ein öffentliches Modul oder einen Ressourcentyp eines Drittanbieters zu deaktivieren, der zuvor in Ihrem Konto aktiviert wurde.

Aktiviere eine öffentliche Erweiterung eines Drittanbieters in deinem Konto

Im folgenden Thema erfahren Sie, wie Sie eine öffentliche Erweiterung eines Drittanbieters in Ihrem Konto aktivieren, sodass sie in dem Konto und der Region, in der sie aktiviert wurde, verwendet werden kann.

> **Note**
>
> Bevor Sie fortfahren, vergewissern Sie sich, dass Sie die erstellt habenIAM-Rolle, die Sie mit dieser Erweiterung verwenden werden.

Themen

- Aktivieren Sie eine öffentliche Erweiterung (Konsole)
- Aktivieren Sie eine öffentliche Erweiterung ()AWS CLI

Aktivieren Sie eine öffentliche Erweiterung (Konsole)

Befolgen Sie die Schritte in diesem Abschnitt, um die Konsole für Folgendes zu verwenden:

- Aktivieren Sie eine öffentliche Erweiterung eines Drittanbieters
- Geben Sie zusätzliche Erweiterungskonfigurationsdaten für Ihr Konto an

Um eine öffentliche Erweiterung zur Verwendung in Ihrem Konto zu aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie im Navigationsbereich unter Registrierung die Option Öffentliche Erweiterungen aus.
4. Verwenden Sie den Filter, um den Erweiterungstyp auszuwählen, und wählen Sie Drittanbieter aus. (Erweiterungen, die von AWS veröffentlicht werden, sind standardmäßig aktiviert.)

5. Wählen Sie die Erweiterung und anschließend Aktivieren aus.

   Wenn mehrere Versionen einer Erweiterung verfügbar sind, können Sie im Menü Version die Version der Erweiterung auswählen, die Sie aktivieren möchten. Die Standardversion ist die neueste Version.

6. Für den Namen der Erweiterung können Sie entweder die Option Standard verwenden ausgewählt lassen oder Standard überschreiben wählen und dann den Alias für den Erweiterungstyp eingeben, den Sie mit dieser Erweiterung verwenden möchten. Der Alias muss dem für den Erweiterungstyp empfohlenen Format entsprechen. Weitere Informationen finden Sie unter Verwenden Sie Aliase, um auf Erweiterungen zu verweisen.

7. Wenn es sich bei der Erweiterung, die Sie aktivieren, um einen Hook- oder Ressourcentyp handelt, geben Sie für die Ausführungsrolle ARN die IAM-Rolle an, die beim Aufrufen der Erweiterung übernommen werden CloudFormation soll. Weitere Informationen finden Sie unter Konfigurieren Sie eine Ausführungsrolle mit IAM-Berechtigungen und einer Vertrauensrichtlinie für den Zugriff auf öffentliche Erweiterungen.

8. Geben Sie für die Logging-Konfiguration, falls gewünscht, die Logging-Konfigurationsinformationen für eine Erweiterung an. Zum Beispiel:

```
{
    "logRoleArn": "arn:aws:iam::account:role/rolename",
    "logGroupName": "log-group-name"
}
```

   Die Protokollierung der Konfigurationsinformationen ist nicht erforderlich, wird aber für Debugging-Zwecke empfohlen. Um die Protokollierungskonfiguration mit Hooks zu verwenden, fügen Sie dieselbe Vertrauensrichtlinie wie für die angegebene Ausführungsrolle hinzu, sodass die Protokollrolle Protokolle in Ihre Protokollgruppe schreiben kann.

   `logRoleArn`und bei `logGroupName` Schlüsselnamen wird zwischen Groß- und Kleinschreibung unterschieden.

9. Wählen Sie unter Versionierung und Automatische Updates aus, wie Sie Updates erhalten möchten.

   - Ein – Wird automatisch auf die neueste Nebenversion aktualisiert. Hauptversionen werden manuell aktualisiert.
   - Aus – Wird niemals automatisch auf die neueste Version aktualisiert. Alle Versionen werden manuell aktualisiert.

Weitere Informationen finden Sie unter Automatisch neue Versionen von Erweiterungen verwenden.

Wenn für die Erweiterung eine zusätzliche Konfiguration erforderlich ist, haben Sie die Möglichkeit, die Konfigurationsdaten jetzt oder nach der Aktivierung der Erweiterung anzugeben.

> **Important**
>
> Wenn es sich bei der Erweiterung, die Sie aktivieren, um einen Hook handelt, ist dieser Schritt erforderlich. Sie müssen `ENABLED` für die `HookInvocationStatus`-Eigenschaft angeben. Dieser Vorgang aktiviert die Eigenschaften des Hooks, die im `properties` Schemabereich des Hooks definiert sind. Weitere Informationen finden Sie in der Syntaxreferenz zum Hook-Konfigurationsschema im AWS CloudFormation Hooks User Guide.

Um die Konfigurationsdaten anzugeben

1. Wählen Sie unter Konfiguration die Option Jetzt konfigurieren und anschließend Erweiterung aktivieren aus.

   CloudFormation zeigt die Seite „Erweiterung konfigurieren“ an. Um das aktuelle Konfigurationsschema für die Erweiterung anzuzeigen, stellen Sie sicher, dass View configuration schema (Konfigurationsschema anzeigen) aktiviert ist.
2. Geben Sie im Textfeld Configuration JSON eine JSON-Zeichenfolge ein, die die Konfigurationsdaten darstellt, die Sie für diese Erweiterung angeben möchten. Das von Ihnen angegebene JSON muss anhand des Konfigurationsschemas der Erweiterung validiert werden.
3. Wählen Sie Erweiterung konfigurieren aus.

Wenn Sie die Erweiterung lieber nach der Aktivierung konfigurieren möchten, können Sie diesen Schritt überspringen und die Konfigurationsdaten zu einem späteren Zeitpunkt angeben.

1. Wählen Sie unter Konfiguration die Option Später konfigurieren und anschließend Erweiterung aktivieren aus.

2. Nachdem die Erweiterung aktiviert wurde, können Sie sie konfigurieren, indem Sie von der Seite mit den aktivierten Erweiterungen zu der Erweiterung navigieren und die Konfigurationsdaten angeben.

Aktivieren Sie eine öffentliche Erweiterung ()AWS CLI

Folgen Sie den Schritten in diesem Abschnitt, um Folgendes AWS CLI zu verwenden:

- Aktivieren Sie eine öffentliche Erweiterung eines Drittanbieters
- Geben Sie zusätzliche Erweiterungskonfigurationsdaten für Ihr Konto an

Aktiviere öffentliche Hooks

Indem Sie Hooks in Ihrem Konto aktivieren, autorisieren Sie einen Hook, definierte Berechtigungen von Ihnen AWS-Konto zu verwenden. CloudFormation entfernt nicht benötigte Berechtigungen, bevor du deine Berechtigungen an den Hook weitergibst. CloudFormationempfiehlt Kunden oder Hook-Benutzern, die Hook-Berechtigungen zu überprüfen und sich darüber im Klaren zu sein, welche Berechtigungen die Hooks haben dürfen, bevor Sie Hooks in Ihrem Konto aktivieren.

Um einen öffentlichen Hook zur Verwendung in deinem Konto zu aktivieren (AWS CLI)

1. Holen Sie sich den ARN für Ihren Hook und speichern Sie ihn. Sie können den ARN eines Hooks mit dem AWS Management Console oder abrufen AWS CLI. Weitere Informationen finden Sie unter Sehen Sie sich die verfügbaren und aktivierten Erweiterungen in der CloudFormation Registrierung an.

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/
Organization-Service-Hook/"
```

2. Verwenden Sie den set-type-configurationBefehl, um die Konfigurationsdaten anzugeben. Das JSON, das Sie übergeben, `--configuration` muss anhand des Konfigurationsschemas des Hook validiert werden. Um den Hook für alle Stack-Operationen zu aktivieren, müssen Sie die `HookInvocationStatus` Eigenschaft `ENABLED` im `HookConfiguration` Abschnitt auf setzen.

```
aws cloudformation set-type-configuration \
  --configuration "{"CloudFormationConfiguration":{"HookConfiguration":
{"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}" \
```

```
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

Weitere Informationen zu den `HookConfiguration` Konfigurationsoptionen finden Sie in der Syntaxreferenz für das Hooks-Konfigurationsschema im AWS CloudFormation Hooks User Guide.

Aktivieren Sie öffentliche Module und Ressourcentypen

Um eine öffentliche Erweiterung zur Verwendung in Ihrem Konto zu aktivieren (AWS CLI)

- Verwenden Sie den activate-typeBefehl, um die Erweiterung zu aktivieren, und geben Sie an, ob die Erweiterung auto aktualisiert werden soll, wenn eine neue Nebenversion der Erweiterung veröffentlicht wird.

  Im folgenden Beispiel wird der öffentliche Amazon-Ressourcenname (ARN) einer öffentlichen Erweiterung angegeben, die für dieses Konto aktiviert werden soll. Darüber hinaus gibt es an, dass die Erweiterung CloudFormation aktualisiert wird, wenn eine neue Nebenversion veröffentlicht wird.

  ```
  aws cloudformation activate-type \
    --public-type-arn public_extension_ARN \
    --execution-role-arn arn:aws:iam::123456789012:role/my-execution-role \
    --auto-update true --region us-west-2
  ```

  Dieser Befehl gibt einen ARN der aktivierten Erweiterung zurück.

  ```
  {
      "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-
  Resource-Example"
  }
  ```

Aktualisieren Sie die Version einer öffentlichen Erweiterung (AWS CLI)

Verwenden Sie activate-type, um die Erweiterung erneut zu aktivieren.

Verwenden Sie die `--version-bump` Option, um anzugeben, ob die Erweiterung auf die neueste `MAJOR` Version oder die neueste Version aktualisiert werden soll. `MINOR`

```
aws cloudformation activate-type --type RESOURCE \
```

```
  --type-name Example::Test::1234567890abcdef0 \
  --type-name-alias Example::Test::Alias \
  --version-bump MAJOR --region us-west-2
```

Aktualisieren Sie eine öffentliche Drittanbieter-Erweiterung in Ihrem Konto

Nachdem Sie eine öffentliche Erweiterung eines Drittanbieters aktiviert haben, können Sie die meisten Erweiterungsdetails von Ihrem Konto aus aktualisieren.

Um eine öffentliche Erweiterung in Ihrem Konto (Konsole) zu aktualisieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie im Navigationsbereich unter Registrierung die Option Aktivierte Erweiterungen aus.
4. Suchen Sie die Erweiterung, die Sie aktualisieren möchten, und wählen Sie sie aus. Weitere Informationen finden Sie unter Sehen Sie sich die verfügbaren und aktivierten Erweiterungen in der CloudFormation Registrierung an.
5. Wählen Sie im Menü Aktionen die Option Bearbeiten und dann die entsprechende Bearbeitungsoption aus:
   - Informationen zum Aktualisieren des Konfigurationsschemas finden Sie unterBearbeiten Sie die Konfigurationsdaten für Erweiterungen in Ihrem Konto.
   - So aktivieren oder deaktivieren Sie automatische Updates:
     1. Wählen Sie Automatische Updates bearbeiten aus.
     2. Wählen Sie „Ein“ oder „Aus“ und anschließend „Speichern“. Weitere Informationen finden Sie unter Automatisch neue Versionen von Erweiterungen verwenden.
   - So aktualisieren Sie die Ausführungsrolle:
     1. Wählen Sie „Ausführungsrolle bearbeiten“.
     2. Geben Sie den ARN der IAM-Rolle CloudFormation an, die Sie beim Aufrufen dieser Erweiterung verwenden möchten, und wählen Sie dann Speichern. Weitere Informationen finden Sie unter Konfigurieren Sie eine Ausführungsrolle mit IAM-Berechtigungen und einer Vertrauensrichtlinie für den Zugriff auf öffentliche Erweiterungen.
   - So aktualisieren Sie die Protokollierungskonfiguration:
     1. Wählen Sie Logging-Konfiguration bearbeiten aus.

2. Bearbeiten Sie die JSON der Protokollierungskonfiguration und wählen Sie dann Speichern aus.

Deaktivieren Sie öffentliche Erweiterungen von Drittanbietern in Ihrem Konto

Wenn Sie eine aktivierte öffentliche Erweiterung eines Drittanbieters nicht mehr benötigen, gehen Sie wie folgt vor, um sie in Ihrem Konto zu deaktivieren.

Themen

- Deaktiviere eine öffentliche Erweiterung in deinem Konto (Konsole)
- Deaktiviere eine öffentliche Erweiterung in deinem Konto (AWS CLI)
- Deaktiviere einen Hook in deinem Konto (AWS CLI)

Deaktiviere eine öffentliche Erweiterung in deinem Konto (Konsole)

Um eine öffentliche Erweiterung in Ihrem Konto zu deaktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie im Navigationsbereich unter Registrierung die Option Aktivierte Erweiterungen aus.
4. Suchen Sie die Erweiterung, die Sie deaktivieren möchten, und wählen Sie sie aus. Weitere Informationen finden Sie unter Sehen Sie sich die verfügbaren und aktivierten Erweiterungen in der CloudFormation Registrierung an.
5. Wählen Sie im Menü Aktionen die Option Deaktivieren.
6. Wählen Sie Deaktivieren aus.

Deaktiviere eine öffentliche Erweiterung in deinem Konto (AWS CLI)

Verwenden Sie den folgenden deactivate-type-Befehl.

```
aws cloudformation deactivate-type --type MODULE \
  --type-name Example::Test::Type::MODULE \
```

```
  --region us-west-2
```

Deaktiviere einen Hook in deinem Konto (AWS CLI)

Wenn Sie einen Hook deaktivieren, wird verhindert, dass der Hook in Ihrem ausgeführt wird, AWS-Konto ohne ihn zu entfernen.

Verwenden Sie den set-type-configurationBefehl und geben Sie an`DISABLED`, `HookInvocationStatus` wie ein Hook deaktiviert werden soll.

Das folgende Beispiel spezifiziert den AWS-Region und den Amazon-Ressourcennamen (ARN) des Hooks, der deaktiviert wird.

```
aws cloudformation set-type-configuration \
  --configuration "{"CloudFormationConfiguration":{"HookConfiguration":
{"HookInvocationStatus": "DISABLED", "FailureMode": "FAIL", "Properties":{}}}}" \
  --type-arn "arn:aws:cloudformation:us-west-2:123456789012:type/hook/MyTestHook" --
region us-west-2
```

Weitere Informationen finden Sie unter Hooks deaktivieren und aktivieren im AWS CloudFormationAWS CloudFormation Hooks User Guide.

Verwenden Sie private Erweiterungen von Drittanbietern, die mit Ihnen geteilt wurden

Um private Erweiterungen von Drittanbietern verwenden zu können, die mit Ihnen geteilt wurden, müssen Sie diese zunächst in den Konten und Regionen registrieren CloudFormation, in denen Sie sie verwenden möchten. Durch die Registrierung der Erweiterung wird eine Kopie der Erweiterung in die CloudFormation Registrierung in Ihrem Konto hochgeladen und aktiviert. Sobald Sie eine private Erweiterung registriert haben, wird sie in der CloudFormation Registrierung für diese AWS-Konto und Region angezeigt, und Sie können sie in Ihren Stack-Vorlagen verwenden.

Themen

- IAM-Berechtigungen für die Registrierung einer privaten Erweiterung eines Drittanbieters
- Häufig verwendete AWS CLI Befehle für die Arbeit mit privaten Erweiterungen
- Registrieren Sie eine private Erweiterung eines Drittanbieters in Ihrem Konto
- Private Erweiterungen von Drittanbietern aus deinem Konto entfernen

IAM-Berechtigungen für die Registrierung einer privaten Erweiterung eines Drittanbieters

Im Rahmen der Registrierung einer privaten Erweiterung können Sie einen Amazon S3 S3-Bucket angeben, der das Erweiterungsprojektpaket enthält. Dieses Paket enthält alle Quelldateien, die für die Erweiterung erforderlich sind, die Sie registrieren möchten. Der Benutzer, der die Erweiterung registriert, muss auf das Projektpaket in diesem Amazon-S3-Bucket zugreifen können. Dazu muss der Benutzer über GetObjectBerechtigungen für das Erweiterungspaket verfügen.

Dies gilt unabhängig davon, ob Sie entweder den register-typeBefehl der AWS CLI oder den submitBefehl der CloudFormation CLI verwenden.

Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Häufig verwendete AWS CLI Befehle für die Arbeit mit privaten Erweiterungen

Zu den häufig verwendeten Befehlen für die Arbeit mit privaten Erweiterungen gehören:

- register-typeum eine private Erweiterung in Ihrem Konto zu registrieren.
- describe-type-registrationum den aktuellen Status einer Registrierungsanfrage zurückzugeben.
- list-typesum die Erweiterungen in Ihrem Konto aufzulisten.
- describe-typeum detaillierte Informationen über eine bestimmte Erweiterung oder eine bestimmte Erweiterungsversion, einschließlich aktueller Konfigurationsdaten, zurückzugeben.
- set-type-configurationum die Konfigurationsdaten für eine Erweiterung in Ihrem Konto anzugeben und Hooks zu deaktivieren und zu aktivieren.
- set-type-default-versionum anzugeben, welche Version einer Erweiterung die Standardversion ist.
- deregister-typeum eine private Erweiterung oder Erweiterungsversion aus Ihrem Konto zu entfernen.

Registrieren Sie eine private Erweiterung eines Drittanbieters in Ihrem Konto

In diesem Thema werden die Schritte zur Registrierung einer privaten Erweiterung eines Drittanbieters beschrieben, die mit Ihnen geteilt wurde, sodass sie in Ihrem Konto verwendet werden kann.

> **Note**
>
> Bevor Sie fortfahren, vergewissern Sie sich, dass Sie über die erforderlichen Voraussetzungen IAM-Berechtigungen für die Registrierung einer privaten Erweiterung verfügen.

Um eine private Erweiterung zu registrieren, die mit Ihnen geteilt wurde (AWS CLI)

1. Suchen Sie den Amazon S3 S3-Bucket, der das Projektpaket für die private Erweiterung enthält, die Sie in Ihrem Konto registrieren möchten.
2. Verwenden Sie den register-typeBefehl, um die private Erweiterung in Ihrem Konto zu registrieren.

   Mit dem folgenden Befehl wird beispielsweise der `My::Resource::Example` Ressourcentyp im angegebenen Verzeichnis registriert AWS-Konto.

   ```
   aws cloudformation register-type --type RESOURCE \
     --type-name My::Resource::Example \
     --schema-handler-package [s3 object path] --region us-west-2
   ```

   `RegisterType` ist ein asynchroner Vorgang und gibt ein Registrierungs-Token zurück, mit dem Sie den Fortschritt Ihrer Registrierungs-Anforderung verfolgen können.

   ```
   {
       "RegistrationToken": "f5525280-104e-4d35-bef5-8f1fexample"
   }
   ```

   Wenn Ihre Erweiterung im AWS APIs Rahmen ihrer Funktionalität aufgerufen wird, müssen Sie eine IAM-Ausführungsrolle erstellen, die die erforderlichen Berechtigungen für deren Aufruf enthält AWS APIs, und diese Ausführungsrolle in Ihrem Konto bereitstellen. Sie können diese

Ausführungsrolle dann mithilfe der `--execution-role-arn` Option angeben. CloudFormation geht dann davon aus, dass diese Ausführungsrolle Ihrem Ressourcentyp die entsprechenden Anmeldeinformationen zur Verfügung stellt.

```
--execution-role-arn arn:aws:iam::123456789012:role/MyIAMRole
```

3. (Optional) Verwenden Sie das Registrierungstoken zusammen mit dem describe-type-registrationBefehl, um den Fortschritt Ihrer Registrierungsanfrage zu verfolgen.

   Wenn die Registrierungsanfrage CloudFormation abgeschlossen ist, wird der Fortschrittsstatus der Anfrage auf festgelegt`COMPLETE`.

   Im folgenden Beispiel wird das vom oben genannten `describe-type-registration` Befehl zurückgegebene Registrierungs-Token verwendet, um Registrierungs-Statusinformationen zurückzugeben.

```
aws cloudformation describe-type-registration \
  --registration-token f5525280-104e-4d35-bef5-8f1fexample \
  --region us-west-2
```

   Der Befehl gibt die folgende Ausgabe zurück.

```
{
    "ProgressStatus": "COMPLETE",
    "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-
Resource-Example",
    "Description": "Deployment is currently in DEPLOY_STAGE of status COMPLETED; ",
    "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
My-Resource-Example/00000001"
}
```

> **Important**
>
> Wenn es sich bei der Erweiterung, die Sie registrieren, um einen Hook handelt, ist dieser nächste Schritt erforderlich. Sie müssen `ENABLED` für die `HookInvocationStatus`-Eigenschaft angeben. Dieser Vorgang aktiviert die Eigenschaften des Hooks, die im `properties` Schemabereich des Hooks definiert sind. Weitere Informationen finden Sie in

der Syntaxreferenz zum Hook-Konfigurationsschema im AWS CloudFormation Hooks User Guide.

Um die Konfigurationsdaten für einen Hook (AWS CLI) anzugeben

1. Holen Sie sich den ARN für Ihren Hook und speichern Sie ihn. Sie können den ARN eines Hooks mit dem AWS Management Console oder abrufen AWS CLI. Weitere Informationen finden Sie unter Sehen Sie sich die verfügbaren und aktivierten Erweiterungen in der CloudFormation Registrierung an.

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/
Organization-Service-Hook/"
```

2. Verwenden Sie den set-type-configurationBefehl, um die Konfigurationsdaten anzugeben. Das JSON, das Sie übergeben, `--configuration` muss anhand des Konfigurationsschemas des Hook validiert werden. Um den Hook zu aktivieren, müssen Sie die `HookInvocationStatus` Eigenschaft `ENABLED` im `HookConfiguration` Abschnitt auf setzen.

```
aws cloudformation set-type-configuration \
  --configuration "{"CloudFormationConfiguration":{"HookConfiguration":
{"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}" \
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

   Weitere Informationen finden Sie in der Syntaxreferenz zum Hook-Konfigurationsschema im AWS CloudFormation Hooks User Guide.

Private Erweiterungen von Drittanbietern aus deinem Konto entfernen

Verwenden Sie den Befehl, um eine private Erweiterung oder eine Erweiterungsversion eines Drittanbieters zu entfernen. deregister-type

Sie können die Registrierung einer bestimmten Erweiterungsversion oder der Erweiterung als Ganzes aufheben. Um die Registrierung einer Erweiterung aufzuheben, müssen Sie alle registrierten Versionen dieser Erweiterung einzeln abmelden. Wenn eine Erweiterung nur eine einzige registrierte Version hat, führt das Aufheben der Registrierung dieser Version dazu, dass die Erweiterung selbst deregistriert wird. Sie können die Standardversion einer Erweiterung nicht deregistrieren, es sei denn,

es handelt sich um die einzige registrierte Version dieser Erweiterung. In diesem Fall wird auch die Erweiterung selbst deregistriert.

> **Note**
>
> Bevor Sie die Registrierung einer Erweiterung aufheben, empfiehlt es sich, den Befehl describe-type zu verwenden, um zu überprüfen, ob die Erweiterung vorhanden ist. Überprüfen Sie dann mit den Befehlen list-stacks und get-template, ob Stacks die Erweiterung verwenden, bevor Sie sie deregistrieren.

Beispiele für Befehle zur Aufhebung der Registrierung von Erweiterungen

Dieser Abschnitt enthält Beispiele, die die verschiedenen Möglichkeiten zur Deregistrierung von privaten Erweiterungen zeigen.

Melden Sie sich nach Typname ab

Verwenden Sie den deregister-typeBefehl with `--type` und `--type-name` options, um Ihre Erweiterung abzumelden.

```
aws cloudformation deregister-type \
  --type MODULE \
  --type-name My::S3::SampleBucket::MODULE
```

Melden Sie sich nach Typname und Version ab

Um die Registrierung einer bestimmten Version Ihrer Erweiterung aufzuheben, geben Sie die `--version-id` Option im Befehl an.

```
aws cloudformation deregister-type \
  --type MODULE \
  --type-name My::S3::SampleBucket::MODULE \
  --version-id 00000001
```

> **Tip**
>
> Um zuerst eine andere Version der Erweiterung als Standard festzulegen, verwenden Sie den set-type-default-versionBefehl.

Per ARN abmelden

Verwenden Sie die `--arn` Option und geben Sie den ARN Ihrer Erweiterung an, um die Registrierung aufzuheben.

```
aws cloudformation deregister-type \
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Organization-
Service-Resource
```

Bearbeiten Sie die Konfigurationsdaten für Erweiterungen in Ihrem Konto

Dieses Thema enthält Anleitungen zur Bearbeitung von Konfigurationsdaten für Erweiterungen in Ihrem Konto in einer bestimmten Region. Erweiterungen können Konfigurationseigenschaften enthalten, die für alle Instanzen der Erweiterung für ein bestimmtes Konto und eine bestimmte Region gelten. Diese werden vom Autor der Erweiterung in der Konfigurationsdefinition der Erweiterung definiert. Wenn die Konfigurations-Definition der Erweiterung erforderliche Eigenschaften enthält, müssen Sie diese Eigenschaften angeben, bevor Sie die Erweiterung in Ihrem Konto und Ihrer Region verwenden können.

Weitere Informationen darüber, wie Konfigurationsdefinitionen bei der Entwicklung einer Erweiterung definiert werden, finden Sie in der folgenden Dokumentation.

- Syntaxreferenz für das Hook-Konfigurationsschema
- Definieren der Konfiguration einer Erweiterung auf Konto-Ebene

Themen

- Für die Verwendung dynamischer Verweise sind Berechtigungen erforderlich
- Bearbeiten Sie die Konfigurationsdaten für eine Erweiterung (Konsole)
- Bearbeiten Sie die Konfigurationsdaten für eine Erweiterung (AWS CLI)

Für die Verwendung dynamischer Verweise sind Berechtigungen erforderlich

Wenn Ihre Konfigurationsdaten dynamische Verweise auf Werte enthalten, die in AWS Systems Manager oder gespeichert sind AWS Secrets Manager, muss jede Rolle, die zur Bereitstellung des

Typs verwendet wird (z. B. beim Erstellen oder Aktualisieren eines Stacks), über die entsprechenden Berechtigungen zum Abrufen dieses Werts verfügen. Das heißt:

- Wenn die Konfigurationsdaten einen im AWS Systems Manager Parameter Store gespeicherten Parameter enthalten, muss der Benutzer oder die Rolle, mit der der Typ bereitgestellt wurde, über Aufrufberechtigungen verfügen GetParameter.
- Wenn die Konfigurationsdaten einen geheimen Schlüssel enthalten, der in gespeichert ist AWS Secrets Manager, muss der Benutzer oder die Rolle, mit der der Typ bereitgestellt wurde, über Aufrufberechtigungen verfügen GetSecretValue.

Weitere Informationen finden Sie unter Ruft mithilfe dynamischer Verweise Werte ab, die in anderen Diensten gespeichert sind.

Bearbeiten Sie die Konfigurationsdaten für eine Erweiterung (Konsole)

Befolgen Sie die Schritte in diesem Abschnitt, um die Konsole für Folgendes zu verwenden:

- Die aktuellen Konfigurationsdaten für eine Erweiterung anzeigen
- Aktualisieren Sie die Erweiterungskonfigurationsdaten für Ihr Konto

Um die aktuellen Konfigurationsdaten für eine Erweiterung anzuzeigen

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie im Navigationsbereich unter Registrierung die Option Aktivierte Erweiterungen aus.
4. Suchen Sie die Erweiterung, die Sie sich ansehen möchten. Weitere Informationen finden Sie unter Sehen Sie sich die verfügbaren und aktivierten Erweiterungen in der CloudFormation Registrierung an.
5. Wählen Sie die Erweiterung aus, um die Details der Erweiterung anzuzeigen.
6. Wählen Sie auf der Seite zu den Erweiterungsdetails die Registerkarte Configuration (Konfiguration).
7. Erweitern Sie die Registerkarte Configuration schema (Schema der Konfiguration), um das für die Erweiterung definierte Konfigurationsschema anzuzeigen.
8. Erweitern Sie die Registerkarte Configuration (Konfiguration), um die aktuelle Konfiguration anzuzeigen, die Sie für diese Erweiterung festgelegt haben.

Um die Konfigurationsdaten für eine Erweiterung zu aktualisieren

1. Wählen Sie auf der Seite mit den Erweiterungsdetails auf der Registerkarte Konfiguration die Option Konfiguration bearbeiten aus.

   Sie können auch unter Aktionen die Option Bearbeiten und dann Konfiguration bearbeiten auswählen.

   CloudFormation zeigt die Seite „Erweiterung konfigurieren“ an. Stellen Sie sicher, dass View configuration schema (Konfigurationsschema anzeigen) aktiviert ist, um das aktuelle Konfigurations-Definitionsschema der Erweiterung anzuzeigen.
2. Geben Sie im Textfeld Configuration JSON eine JSON-Zeichenfolge ein, die das Konfigurationsschema darstellt, das Sie für diese Erweiterung festlegen möchten. Es muss mit dem Schema validiert werden, das in Configuration schema (Konfigurationsschema) angegeben ist.
3. Wählen Sie Erweiterung konfigurieren.

Bearbeiten Sie die Konfigurationsdaten für eine Erweiterung (AWS CLI)

Folgen Sie den Schritten in diesem Abschnitt, um Folgendes AWS CLI zu verwenden:

- Die aktuellen Konfigurationsdaten für eine Erweiterung anzeigen
- Aktualisieren Sie die Erweiterungskonfigurationsdaten für Ihr Konto

Um die aktuellen Konfigurationsdaten für eine Erweiterung anzuzeigen

- Verwenden Sie den describe-typeBefehl, um detaillierte Informationen über die Erweiterung zurückzugeben. Das `ConfigurationSchema`-Element der Ausgabe enthält die aktuelle Konfigurations-Definition der Erweiterung in einer bestimmten Region.

  Als alternative Vorgehensweise verwenden Sie den Befehl batch-describe-type-configurations, um Konfigurationsdaten über mehrere Erweiterungen zurückzugeben.

Um die Konfigurationsdaten für eine Erweiterung zu aktualisieren

- Verwenden Sie den set-type-configurationBefehl, um die Konfigurationsdaten anzugeben. Das JSON, das Sie übergeben, `--configuration` muss anhand des Konfigurationsschemas der Erweiterung validiert werden.

  Im folgenden Beispiel gibt der set-type-configuration Befehl die Konfigurationsdaten *`"{"CredentialKey": "testUserCredential"}"`* für die `--configuration` Option an.

  ```
  aws cloudformation set-type-configuration --type RESOURCE \
    --type-name My::Resource::Example \
    --configuration-alias default \
    --configuration "{"CredentialKey": "testUserCredential"}" \
    --region us-west-2
  ```

Ressourcentypen aufzeichnen in AWS Config

Sie können festlegen, dass Ihre privaten Ressourcentypen AWS Config automatisch nachverfolgt und Änderungen an diesen Ressourcen als Konfigurationselemente aufgezeichnet werden. Auf diese Weise können Sie den Konfigurationsverlauf für diese privaten Ressourcentypen sowie AWS-Config-Regeln Schreibregeln zur Überprüfung der bewährten Konfigurationsmethoden anzeigen. AWS Config ist für die Hook-Erweiterung erforderlich.

Um Ihre privaten Ressourcentypen AWS Config automatisch verfolgen zu lassen:

- Verwalten Sie die Ressourcen über CloudFormation. Dazu gehört das Durchführen aller Vorgänge zum Erstellen, Aktualisieren und Löschen von Ressourcen durch CloudFormation.

  > **Note**
  >
  > Wenn Sie eine IAM-Rolle verwenden, um Ihre Stack-Operationen auszuführen, muss diese IAM-Rolle über die Berechtigung verfügen, die folgenden AWS Config Aktionen aufzurufen:
  >
  > - PutResourceConfig
  > - DeleteResourceConfig

- Konfigurieren Sie AWS Config es so, dass alle Ressourcentypen aufgezeichnet werden. Weitere Informationen finden Sie unter Aufzeichnen von Konfigurationen für Ressourcen von Drittanbietern mithilfe von AWS CLI im AWS Config Entwicklerhandbuch.

> **Note**
>
> AWS Config unterstützt nicht die Aufzeichnung von privaten Ressourcen, deren Eigenschaften sowohl als erforderlich als auch als schreibgeschützt definiert sind. Standardmäßig werden Ressourceneigenschaften, die als schreibgeschützt definiert sind, in dem Schema, das zur Erstellung des Konfigurationselements verwendet wurde, nicht zurückgegeben. AWS Config Aus diesem Grund führt das Einschließen einer Eigenschaft, die sowohl als lesegeschützt als auch als erforderlich definiert ist, dazu, dass die Erstellung des Konfigurationselements fehlschlägt, da keine erforderliche Eigenschaft vorhanden sein wird. Um das Schema anzuzeigen, das zur Erstellung des Konfigurationselements verwendet wird, können Sie die `schema` Eigenschaft der DescribeTypeAktion überprüfen.

Weitere Informationen zu Konfigurationselementen finden Sie unter Konfigurationselemente im AWS Config -Entwicklerhandbuch.

Verhindern, dass vertrauliche Eigenschaften in einem Konfigurationselement aufgezeichnet werden

Ihr Ressourcentyp kann Eigenschaften enthalten, die Sie als vertrauliche Informationen betrachten, z. B. Passwörter, Secrets oder andere vertrauliche Daten, die nicht als Teil des Konfigurationselements aufgezeichnet werden sollen. Um zu verhindern, dass eine Eigenschaft im Konfigurationselement aufgezeichnet wird, können Sie diese Eigenschaft in die `writeOnlyproperties`-Liste in Ihrem Ressourcentypschema aufnehmen. Als `writeOnlyproperties` aufgelistete Ressourceneigenschaften können vom Benutzer angegeben werden, werden aber nicht von einer `read` oder `list`-Anforderung zurückgegeben.

Weitere Informationen finden Sie writeOnlyPropertiesim CloudFormation CLI-Benutzerhandbuch.

Kontinuierliche Lieferung mit CodePipeline

Die fortlaufende Lieferung ist eine Freigabepraxis, bei der Codeänderungen automatisch erstellt, getestet und für die Produktionsfreigabe vorbereitet werden. Mit CloudFormation und können Sie Continuous Delivery verwenden CodePipeline, um Änderungen an Ihren CloudFormation Vorlagen automatisch zu erstellen und zu testen, bevor Sie sie in Produktions-Stacks hochstufen. Mit diesem Release-Prozess können Sie schnell und zuverlässig Änderungen an Ihrer AWS Infrastruktur vornehmen.

Sie können beispielsweise einen Workflow erstellen, der automatisch einen Test-Stack erstellt, wenn Sie eine aktualisierte Vorlage an ein Code-Repository absenden. Nachdem Sie den Test-Stack CloudFormation erstellt haben, können Sie ihn testen und dann entscheiden, ob Sie die Änderungen auf einen Produktions-Stack übertragen möchten. Weitere Informationen zum Nutzen von Continuous Delivery finden Sie unter Was ist fortlaufende Lieferung?.

Wird verwendet CodePipeline , um einen Continuous-Delivery-Workflow aufzubauen, indem eine Pipeline für CloudFormation Stacks erstellt wird. CodePipeline verfügt über eine integrierte Integration mit CloudFormation, sodass Sie CloudFormation spezifische Aktionen innerhalb einer Pipeline angeben können, z. B. das Erstellen, Aktualisieren oder Löschen eines Stacks. Weitere Informationen zu CodePipeline finden Sie im AWS CodePipeline Benutzerhandbuch.

Themen

- Walkthrough: Erstellen einer Pipeline für Test- und Produktions-Stacks
- AWS CloudFormation Referenz zu den Konfigurationseigenschaften
- AWS CloudFormation Artefakte
- Verwenden von Funktionen zum Überschreiben von Parametern mit CodePipeline Pipelines

Walkthrough: Erstellen einer Pipeline für Test- und Produktions-Stacks

Stellen Sie sich einen Release-Prozess vor, bei dem Sie eine AWS CloudFormation Vorlage einreichen, anhand derer CloudFormation dann automatisch ein Test-Stack erstellt wird. Nachdem Sie die Test-Stack geprüft haben, können Sie in einer Vorschau betrachten, wie sich Änderungen auf den Produktions-Stack auswirken, und dann entscheiden, ob Sie die Änderungen implementieren. Um diesen Workflow durchzuführen, könnten Sie CloudFormation Ihren Test-Stack erstellen,

den Test-Stack löschen, einen Änderungssatz erstellen und dann den Änderungssatz ausführen. Bei jeder Aktion müssen Sie allerdings manuell mit CloudFormation interagieren. In dieser exemplarischen Vorgehensweise erstellen wir eine CodePipeline Pipeline, die viele dieser Aktionen automatisiert und Ihnen hilft, mit Ihren CloudFormation Stacks einen Continuous-Delivery-Workflow zu erreichen.

Voraussetzungen

In dieser exemplarischen Vorgehensweise wird davon ausgegangen, dass Sie CodePipeline und verwendet haben und wissen CloudFormation, wie Pipelines, Vorlagen und AWS CloudFormation Stacks funktionieren. Weitere Informationen zu CodePipeline finden Sie im AWS CodePipeline Benutzerhandbuch. Sie benötigen auch einen Amazon S3 S3-Bucket in demselben, in dem AWS-Region Sie Ihre Pipeline erstellen.

> **Important**
>
> Die WordPress Beispielvorlage erstellt eine EC2 Instance, für die eine Verbindung zum Internet erforderlich ist. Stellen Sie sicher, dass Sie über ein Standard-VPC und Subnetz verfügen, die Datenverkehr ins Internet zulassen.

Walkthrough-Übersicht

In dieser exemplarischen Vorgehensweise wird eine Pipeline für eine WordPress Beispielsite in einem Stapel erstellt. Die Pipeline ist in drei Phasen unterteilt. Jede Phase enthält mindestens eine Aktion, wobei es sich um eine in der Pipeline mit Ihren Artefakten (Ihre Eingabe) ausgeführte Aufgabe handelt. Eine Phase ist eine Organisationseinheit von Aktionen in einer Pipeline. CodePipeline muss alle Aktionen einer Phase abschließen, bevor in der Phase neue Artefakte verarbeitet werden können, z. B. wenn Sie neue Eingaben zum erneuten Ausführen der Pipeline abgesendet haben.

Am Ende dieser Anleitung haben Sie eine Pipeline, in welcher der folgende Workflow ausgeführt wird:

1. In der ersten Phase der Pipeline wird ein Quellartefakt (eine CloudFormation Vorlage und ihre Konfigurationsdateien) aus einem Repository abgerufen.

   Sie bereiten ein Artefakt vor, das eine WordPress Beispielvorlage enthält, und laden es in einen S3-Bucket hoch.
2. In der zweiten Phase erstellt die Pipeline einen Test-Stack und wartet auf Ihre Genehmigung.

Nachdem Sie den Test-Stack geprüft haben, können Sie mit der ursprünglichen Pipeline fortfahren oder ein weiteres Artefakt erstellen und absenden, um Änderungen vorzunehmen. Wenn Sie den Stack genehmigen, wird er in dieser Phase gelöscht, und die Pipeline fährt mit der nächsten Phase fort.

3. In der dritten Phase erstellt die Pipeline einen Änderungssatz für einen Produktions-Stack und wartet dann auf Ihre Genehmigung.

   In der ersten Ausführung ist kein Produktions-Stack vorhanden. Das Änderungssatz zeigt Ihnen alle Ressourcen, die erstellt CloudFormation werden. Wenn Sie es genehmigen, wird der Änderungssatz in dieser Phase ausgeführt und der Produktions-Stack erstellt.

> **Note**
>
> CloudFormation ist ein kostenloser Service. Ihnen werden jedoch die AWS Ressourcen, z. B. die EC2 Instanz, die Sie in Ihren Stack aufnehmen, zum jeweils aktuellen Tarif in Rechnung gestellt. Weitere Informationen zur AWS Preisgestaltung finden Sie auf der Detailseite für jedes Produkt unter http://aws.amazon.com.

Schritt 1: Artefakt bearbeiten und in einen S3-Bucket hochladen

Bevor Sie Ihre Pipeline erstellen, müssen Sie Ihr Quell-Repository und Ihre Dateien einrichten. CodePipelinekopiert diese Quelldateien in den Artefaktspeicher Ihrer Pipeline und verwendet sie dann, um Aktionen in Ihrer Pipeline auszuführen, wie z. B. das Erstellen eines CloudFormation Stacks.

Wenn Sie Amazon Simple Storage Service (Amazon S3) als Quell-Repository verwenden, CodePipeline müssen Sie Ihre Quelldateien komprimieren, bevor Sie sie in einen S3-Bucket hochladen. Die komprimierte Datei ist ein CodePipeline Artefakt, das eine CloudFormation Vorlage, eine Vorlagen-Konfigurationsdatei oder beides enthalten kann. Wir stellen ein Artefakt bereit, das eine WordPress Beispielvorlage und zwei Vorlagenkonfigurationsdateien enthält. Die beiden Konfigurationsdateien spezifizieren Parameterwerte für die WordPress Vorlage. CodePipeline verwendet sie, wenn es die WordPress Stapel erstellt. Eine Datei enthält die Parameterwerte für einen Test-Stack und die andere die für einen Produktions-Stack. Sie müssen die Konfigurationsdateien bearbeiten, um beispielsweise einen vorhandenen EC2 Schlüsselpaarnamen

anzugeben, dessen Eigentümer Sie sind. Weitere Informationen zu Artefakten finden Sie unter AWS CloudFormation Artefakte.

Nachdem das Artefakt erstellt worden ist, können Sie es in einen S3-Bucket hochladen.

So können Sie das Artefakt bearbeiten und hochladen

1. Laden Sie das Beispielartefakt herunter und öffnen Sie es: https://s3.amazonaws.com/cloudformation-examples/user - -single-instance.zip. guide/continuous-deployment/wordpress

   Das Artefakt enthält drei Dateien:

   - Die WordPress Beispielvorlage: `wordpress-single-instance.yaml`
   - Die Vorlagenkonfigurationsdatei für den Test-Stack: `test-stack-configuration.json`
   - Die Vorlagenkonfigurationsdatei für den Produktions-Stack: `prod-stack-configuration.json`
2. Extrahieren Sie alle Dateien und ändern Sie dann die Vorlagenkonfigurationsdateien mit einem beliebigen Texteditor.

   Öffnen Sie die Konfigurationsdateien, um zu sehen, dass sie Schlüssel-Wert-Paare enthalten, die den Parametern der WordPress Vorlage zugeordnet sind. Die Konfigurationsdateien legen die Parameterwerte fest, die in der Pipeline zum Erstellen der Test- und Produktions-Stacks verwendet werden.

   Bearbeiten Sie die Datei `test-stack-configuration.json`, um Parameterwerte für den Test-Stack anzugeben, und die Datei `prod-stack-configuration.json` für den Produktions-Stack.

   - Ändern Sie die Werte der `DBRootPassword` Schlüssel `DBPassword` und in Kennwörter, mit denen Sie sich bei Ihrer WordPress Datenbank anmelden können. Wie in der WordPress Vorlage definiert, dürfen die Parameterwerte nur alphanumerische Zeichen enthalten.
   - Ändern Sie den Wert des `KeyName` Schlüssels in einen vorhandenen EC2 Schlüsselpaarnamen in der Region, in der Sie Ihre Pipeline erstellen werden.
3. Fügen Sie die geänderten Konfigurationsdateien der ursprünglichen Artefaktdatei (`.zip`) hinzu, wobei Sie doppelte Dateien ersetzen.

   Sie verfügen nun über ein benutzerdefiniertes Artefakt, das Sie in einen S3-Bucket hochladen können.

4. Laden Sie das Artefakt in einen S3-Bucket hoch, dessen Eigentümer Sie sind.

   Merken Sie sich den Speicherort der Datei. Sie geben den Speicherort dieser Datei beim Erstellen der Pipeline an.

   Hinweise zu Artefakt und S3-Bucket:

   - Verwenden Sie einen Bucket, der sich in derselben AWS Region befindet, in der Sie Ihre Pipeline erstellen werden.
   - CodePipeline setzt voraus, dass für den Bucket die Versionierung aktiviert ist.
   - Sie können auch Dienste verwenden, bei denen Sie Ihre Dateien nicht komprimieren müssen, bevor Sie sie hochladen, z. B. GitHub oder CodeCommit, für Ihr Quell-Repository.
   - Artefakte können vertrauliche Informationen enthalten, z. B. Kennwörter. Beschränken Sie den Zugriff, sodass nur autorisierte Benutzer die Datei anzeigen können. Wenn Sie das tun, stellen Sie sicher, dass Sie weiterhin auf die Datei zugreifen CodePipeline können. Wenn Sie beispielsweise Ihr Artefakt in einen S3-Bucket hochladen, verwenden Sie S3-Bucket-Richtlinien oder Benutzerrichtlinien zum Einschränken des Zugriffs.

Sie haben jetzt ein Artefakt, das in Ihre Pipeline übernommen werden CodePipeline kann. Im nächsten Schritt geben Sie den Standort des Artefakts an und erstellen die WordPress Pipeline.

Schritt 2: Pipeline-Stack erstellen

Um die WordPress Pipeline zu erstellen, verwenden Sie eine CloudFormation Beispielvorlage. Zusätzlich zum Aufbau der Pipeline richtet die Vorlage (IAM) -Servicerollen AWS Identity and Access Management (IAM) für CodePipeline und CloudFormation, einen S3-Bucket für den CodePipeline Artefaktspeicher und ein Amazon Simple Notification Service (Amazon SNS) -Thema ein, an das die Pipeline Benachrichtigungen sendet, z. B. Benachrichtigungen über Rezensionen. Die Beispielvorlage macht es einfach, diese Ressourcen in einem einzigen Stack bereitzustellen und zu konfigurieren. CloudFormation

Weitere Informationen zur Konfiguration der Pipeline finden Sie unter Aufgaben der Pipeline.

> **Important**
>
> Die WordPress Beispielvorlage erstellt eine EC2 Instanz, für die eine Verbindung zum Internet erforderlich ist. Stellen Sie sicher, dass Ihr Standard-VPC und Subnetz Datenverkehr ins Internet zulassen.

So erstellen Sie den Pipeline-Stack

1. Laden Sie die Beispielvorlage unter https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/basic -pipeline.yml herunter. Speichern Sie die Datei auf dem Computer.
2. Öffnen Sie die Konsole unter. CloudFormation https://console.aws.amazon.com/cloudformation/
3. Wählen Sie eine AWS Region, die CodePipeline und unterstützt CloudFormation.

   Weitere Informationen finden Sie unter CodePipeline Endpunkte und Kontingente und AWS CloudFormation Endpunkte und Kontingente in der. Allgemeine AWS-Referenz
4. Wählen Sie Stack erstellen aus.
5. Wählen Sie unter Specify template (Vorlage angeben) die Option Upload a template file (Eine Vorlagendatei hochladen) und dann die Vorlage aus, die Sie gerade heruntergeladen haben, `basic-pipeline.yml`.
6. Wählen Sie Weiter aus.
7. Geben Sie für Stack name `sample-WordPress-pipeline` ein.
8. Geben Sie im Abschnitt Parameters die folgenden Parameterwerte an und wählen Sie dann Next. Wenn Sie beim Festlegen von Stack-Parametern dieselben Namen für die WordPress Vorlage und ihre Konfigurationsdateien beibehalten haben, können Sie die Standardwerte verwenden. Falls nicht, geben Sie die Dateinamen an, die Sie verwendet haben.

   PipelineName

   Der Name der Pipeline, z. B. `WordPress-test-pipeline`.

   S3 Bucket

   Der Name des S3-Bucket, in dem Sie Ihr Artefakt (Datei `.zip`) gespeichert haben.

SourceS3Key

Der Dateiname Ihres Artefakts. Wenn Sie das Artefakt in einem Ordner gespeichert haben, geben Sie diesen zusammen mit dem Dateinamen an, z. B. `folder/subfolder/wordpress-single-instance.zip`.

E-Mail

Die E-Mail-Adresse, an die Pipeline-Benachrichtigungen CodePipeline gesendet werden, z. `myemail@example.com` B.

9. Für diese Anleitung müssen Sie keine Tags hinzufügen oder erweiterte Einstellungen angeben, wählen Sie daher Weiter aus.
10. Stellen Sie sicher, dass der Stack-Name und die Vorlagen-URL richtig sind und wählen Sie Create stack (Stack erstellen) aus.
11. Um zu bestätigen, dass Sie wissen, dass CloudFormation dadurch IAM-Ressourcen erstellt werden könnten, aktivieren Sie das Kontrollkästchen.

Es kann mehrere Minuten dauern, AWS CloudFormation bis Ihr Stack erstellt ist. Betrachten Sie die Stack-Ereignisse, um den Fortschritt zu überwachen. Weitere Informationen finden Sie unter Stack-Fortschritt überwachen.

Nachdem Ihr Stack erstellt wurde, CodePipeline startet Ihre neue Pipeline. Informationen zum Anzeigen deren Status finden Sie unter -CodePipeline Konsole. Wählen Sie in der Liste der Pipelines die Option WordPress-test-pipeline aus.

Aufgaben der Pipeline

In diesem Abschnitt werden die drei Phasen der Pipeline anhand von Ausschnitten aus der Beispiel-Pipeline-Vorlage erklärt. WordPress

Phase 1: Quelle

Die erste Phase der Pipeline ist die Quellphase, in der Sie den Speicherort Ihres Quellcodes angeben. Jedes Mal, wenn Sie eine Revision an diesen Speicherort übertragen, wird Ihre Pipeline CodePipeline erneut ausgeführt.

Der Quellcode befindet sich in einem S3-Bucket und wird durch seinen Dateinamen identifiziert. Sie haben diese Werte als Eingabeparameterwerte beim Erstellen des Pipeline-Stacks angegeben. Damit das Quell-Artefakt auch in nachfolgenden Phasen verwendet werden kann, wird im Codebeispiel

die `OutputArtifacts`-Eigenschaft mit dem Namen `TemplateSource` angegeben. Damit dieses Artefakt in späteren Phasen verwendet werden kann, legen Sie `TemplateSource` als Eingabe-Artefakt fest.

```
- Name: S3Source
  Actions:
    - Name: TemplateSource
      ActionTypeId:
        Category: Source
        Owner: AWS
        Provider: S3
        Version: '1'
      Configuration:
        S3Bucket: !Ref 'S3Bucket'
        S3ObjectKey: !Ref 'SourceS3Key'
      OutputArtifacts:
        - Name: TemplateSource
```

Stufe 2: TestStage

In der Phase `TestStage` erstellt die Pipeline den Test-Stack, wartet auf Ihre Genehmigung und löscht dann den Test-Stack.

Für die `CreateStack` Aktion verwendet die Pipeline die Testkonfigurationsdatei und die WordPress Vorlage, um den Teststapel zu erstellen. Beide Dateien sind in dem Eingabe-Artefakt `TemplateSource` enthalten, das aus der Quellphase übernommen wurde. Im Codefragment wird der Aktionsmodus `REPLACE_ON_FAILURE` verwendet. Wenn bei der Stack-Erstellung ein Fehler auftritt, ersetzt die Pipeline den Stack, sodass Sie den Stack nicht bereinigen oder beheben müssen, bevor Sie die Pipeline erneut ausführen. Der Aktionsmodus ist für die schnelle Iteration über Test-Stacks nützlich. Für die `RoleArn` Eigenschaft ist der Wert eine AWS CloudFormation Servicerolle, die an anderer Stelle in der Vorlage deklariert ist.

Die Aktion `ApproveTestStack` hält die Pipeline zeitweilig an und sendet eine Benachrichtigung an die E-Mail-Adresse, die Sie beim Erstellen des Pipeline-Stacks angegeben haben. Während die Pipeline angehalten ist, können Sie den WordPress Teststapel und seine Ressourcen überprüfen. Wird verwendet CodePipeline , um diese Aktion zu genehmigen oder abzulehnen. Die `CustomData`-Eigenschaft enthält eine Beschreibung der Aktion, die Sie genehmigen. Die Pipeline fügt diese der Benachrichtigungs-E-Mail hinzu.

Nachdem Sie diese Aktion genehmigt haben, CodePipeline geht es zur `DeleteTestStack` Aktion über und löscht den WordPress Teststapel und die zugehörigen Ressourcen.

```
- Name: TestStage
  Actions:
    - Name: CreateStack
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      InputArtifacts:
        - Name: TemplateSource
      Configuration:
        ActionMode: REPLACE_ON_FAILURE
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: !Ref TestStackName
        TemplateConfiguration: !Sub "TemplateSource::${TestStackConfig}"
        TemplatePath: !Sub "TemplateSource::${TemplateFileName}"
      RunOrder: '1'
    - Name: ApproveTestStack
      ActionTypeId:
        Category: Approval
        Owner: AWS
        Provider: Manual
        Version: '1'
      Configuration:
        NotificationArn: !Ref CodePipelineSNSTopic
        CustomData: !Sub 'Do you want to create a change set against the production
 stack and delete the ${TestStackName} stack?'
      RunOrder: '2'
    - Name: DeleteTestStack
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      Configuration:
        ActionMode: DELETE_ONLY
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: !Ref TestStackName
      RunOrder: '3'
```

Stufe 3: ProdStage

In der Phase `ProdStage` der Pipeline wird ein Änderungssatz für den bestehenden Produktions-Stack erstellt, auf die Genehmigung gewartet und der Änderungssatz anschließend ausgeführt.

Ein Änderungssatz bietet eine Vorschau aller Änderungen, die Sie an Ihrem Produktions-Stack vornehmen AWS CloudFormation werden, bevor Sie sie implementieren. Während der ersten Ausführung der Pipeline ist kein laufender Produktions-Stack vorhanden. Das Änderungssatz zeigt die Aktionen, die bei der Erstellung des Teststapels AWS CloudFormation ausgeführt wurden. Um den Änderungssatz zu erstellen, verwendet die `CreateChangeSet` Aktion die WordPress Beispielvorlage und die Produktionsvorlagenkonfiguration aus dem `TemplateSource` Eingabeartefakt.

Ähnlich wie in der vorherigen Phase, hält die Aktion `ApproveChangeSet` die Pipeline an und sendet eine E-Mail-Benachrichtigung. Während die Pipeline angehalten ist, können Sie sich den Änderungssatz ansehen, um alle vorgeschlagenen Änderungen am Produktions-Stack zu überprüfen. WordPress Verwenden Sie diese CodePipeline Option, um diese Aktion zu genehmigen oder abzulehnen, um die Pipeline fortzusetzen bzw. zu beenden.

Nachdem Sie diese Aktion genehmigt haben, führt die `ExecuteChangeSet` Aktion den Änderungssatz AWS CloudFormation aus, sodass alle im Änderungssatz beschriebenen Aktionen ausgeführt werden. Für den ersten Lauf AWS CloudFormation wird der WordPress Produktionsstapel erstellt. In jeder nachfolgenden Ausführung der Aktion aktualisiert AWS CloudFormation den Stack.

```
- Name: ProdStage
  Actions:
    - Name: CreateChangeSet
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      InputArtifacts:
        - Name: TemplateSource
      Configuration:
        ActionMode: CHANGE_SET_REPLACE
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: !Ref ProdStackName
        ChangeSetName: !Ref ChangeSetName
        TemplateConfiguration: !Sub "TemplateSource::${ProdStackConfig}"
        TemplatePath: !Sub "TemplateSource::${TemplateFileName}"
```

```
      RunOrder: '1'
    - Name: ApproveChangeSet
      ActionTypeId:
        Category: Approval
        Owner: AWS
        Provider: Manual
        Version: '1'
      Configuration:
        NotificationArn: !Ref CodePipelineSNSTopic
        CustomData: !Sub 'A new change set was created for the ${ProdStackName} stack.
 Do you want to implement the changes?'
      RunOrder: '2'
    - Name: ExecuteChangeSet
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      Configuration:
        ActionMode: CHANGE_SET_EXECUTE
        ChangeSetName: !Ref ChangeSetName
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: !Ref ProdStackName
      RunOrder: '3'
```

Schritt 3: Sehen Sie sich den WordPress Stapel an

Beim CodePipeline Durchlaufen der Pipeline werden Test- und Produktionsstapel erstellt. CloudFormation Verwenden Sie die Konsole, um den Status dieser Stacks und ihre Ausgabe zu sehen. CloudFormation

So zeigen Sie einen Stack an

1. Öffnen Sie die CloudFormation Konsole unter. https://console.aws.amazon.com/cloudformation/
2. Je nachdem, ob sich die Pipeline in der Test- oder Produktionsphase befindet, wählen Sie den `Test-MyWordPressSite`- oder den `Prod-MyWordPressSite`-Stack.
3. Wenn Sie den Status des Stack überprüfen möchten, zeigen Sie die Ereignisse des Stack an.

Wenn sich der Stack in einem fehlerhaften Zustand befindet, zeigen Sie den Status an, um die Ursache des Stack-Fehlers zu finden. Korrigieren Sie den Fehler und führen Sie die Pipeline erneut

aus. Wenn sich der Stack im `CREATE_COMPLETE` Status befindet, sehen Sie sich seine Ausgaben an, um die URL Ihrer WordPress Site abzurufen.

Sie haben erfolgreich CodePipeline damit einen Workflow für die kontinuierliche Bereitstellung für eine WordPress Beispiel-Website erstellt. Wenn Sie Änderungen an den S3-Bucket senden, CodePipeline wird automatisch eine neue Version erkannt und Ihre Pipeline erneut ausgeführt. Dieser Workflow vereinfacht es, Änderungen zu senden und zu testen, bevor Sie die Produktions-Website verändern.

Schritt 4: Bereinigen von Ressourcen

Um sicherzustellen, dass Ihnen keine Gebühren für ungewollte Dienste berechnet werden, sollten Sie Ihre Ressourcen löschen.

> **Important**
>
> Löschen Sie die Test- und WordPress Produktionsstapel, bevor Sie den Pipeline-Stack löschen. Der Pipeline-Stack enthält eine Servicerolle, die zum Löschen der WordPress Stacks erforderlich ist. Wenn Sie zuerst den Pipeline-Stack gelöscht haben, können Sie den WordPress Stacks eine weitere Servicerolle — Amazon Resource Name (ARN) — zuordnen und sie dann löschen.

So löschen Sie Objekte im Artefaktspeicher

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie den S3-Bucket aus, der als Artefaktspeicher Ihrer Pipeline CodePipeline verwendet wurde.

   Der Bucket-Name hat folgendes Format: *`stackname-artifactstorebucket-id`*. Wenn Sie diese schrittweise Anleitung befolgt haben, dann ähnelt der Bucket-Name dem folgenden Beispiel: `sample-WordPress-pipeline-artifactstorebucket-12345abcd12345`.
3. Löschen Sie alle Objekte im S3-Bucket, der als Artefaktspeicher fungiert.

   Wenn Sie den Pipeline-Stack im nächsten Schritt löschen, muss dieser Bucket leer sein. Andernfalls kann der Bucket CloudFormation nicht gelöscht werden.

So löschen Sie Stacks

1. Wählen Sie in der CloudFormation Konsole den Stack aus, den Sie löschen möchten.

   Wenn die WordPress Stacks, die von der Pipeline erstellt wurden, noch laufen, wählen Sie sie zuerst aus. Standardmäßig lauten die Stack-Namen `Test-MyWordPressSite` und `Prod-MyWordPressSite`.

   Wenn Sie die Stapel bereits gelöscht haben, wählen Sie den WordPress `sample-WordPress-pipeline` Stapel aus.
2. Wählen Sie die Option Actions (Aktionen) und anschließend Delete Stack (Stack löschen) aus.
3. Wählen Sie im Bestätigungsdialogfeld Yes, Delete aus.

CloudFormation löscht aus dem Stack alle Ressourcen des Stacks, wie die EC2 Instanz, das Benachrichtigungsthema, die Servicerolle und die Pipeline.

Da Sie nun wissen, wie Sie einen grundlegenden CloudFormation Workflow erstellen CodePipeline, können Sie die Beispielvorlage und die Artefakte als Ausgangspunkt für die Erstellung Ihres eigenen Workflows verwenden.

Weitere Informationen finden Sie auch unter

Die folgenden verwandten Ressourcen unterstützen Sie bei der Arbeit mit diesen Parametern.

- Weitere Informationen zu den CloudFormation Aktionsparametern finden Sie in CodePipeline der Konfigurationsreferenz für AWS CloudFormation Bereitstellungsaktionen im AWS CodePipeline Benutzerhandbuch.
- Beispiels-Vorlagenwerte nach Aktionsanbietern, wie etwa für das `Owner`-Feld oder die `configuration`-Felder, finden Sie in der Referenz zur Aktionsstruktur im Benutzerhandbuch für AWS CodePipeline .
- Beispiele für Pipeline-Stack-Vorlagen im YAML- oder JSON-Format finden Sie AWS CloudFormation im AWS CodePipeline Benutzerhandbuch unter Tutorial: Eine Pipeline erstellen mit.

AWS CloudFormation Referenz zu den Konfigurationseigenschaften

Wenn Sie eine CodePipeline Pipeline erstellen, fügen Sie der Pipeline eine `Deploy` Aktion mit AWS CloudFormation als Anbieter hinzu. Anschließend müssen Sie angeben, welche AWS CloudFormation Aktion die Pipeline aufruft und welche Einstellungen die Aktion hat. In diesem Thema werden die AWS CloudFormation -Konfigurationseigenschaften beschrieben. Um Eigenschaften anzugeben, können Sie die CodePipeline Konsole verwenden oder ein JSON-Objekt erstellen, das für die AWS CLI, CodePipeline API oder AWS CloudFormation Vorlagen verwendet werden soll.

Themen

- Konfigurationseigenschaften (Konsole)
- Konfigurationseigenschaften (JSON-Objekt)
- Weitere Informationen finden Sie auch unter

Konfigurationseigenschaften (Konsole)

Die CodePipeline Konsole zeigt die Konfigurationseigenschaften an und gibt die Eigenschaften an, die je nach dem von Ihnen ausgewählten Aktionsmodus erforderlich sind.

> **Note**
>
> Wenn Sie eine Pipeline erstellen, können Sie nur die Aktionsmodi Create or update a stack (Einen Stack erstellen oder aktualisieren) oder Create or replace a change set (Einen Änderungssatz erstellen oder ersetzen) angeben. Die Eigenschaften im Advanced (Eweitert)-Abschnitt sind nur verfügbar, wenn Sie eine Pipeline bearbeiten.

Action-Modus

Die AWS CloudFormation Aktion, die bei CodePipeline der Verarbeitung der zugehörigen Phase aufgerufen wird. Wählen Sie einen der folgenden Aktionsmodi aus:

- Create or replace a change set erstellt den Änderungssatz, falls er noch nicht existiert, basierend auf dem Stack-Namen und der Vorlage, die Sie übermitteln. Wenn der Änderungssatz vorhanden ist, AWS CloudFormation wird er gelöscht und anschließend ein neuer erstellt.

- Create or update a stack erstellt oder aktualisiert einen Stack, wenn der angegebene Stack nicht vorhanden ist. Wenn der Stapel vorhanden ist, wird der Stapel AWS CloudFormation aktualisiert. Verwenden Sie diese Aktion, um bestehende Stacks zu aktualisieren. CodePipeline ersetzt den Stapel nicht.
- Delete a stack löscht einen Stack. Wenn Sie einen Stack angeben, der nicht vorhanden ist, wird die Aktion ohne Löschen eines Stacks erfolgreich abgeschlossen.
- Execute a change set führt einen Änderungssatz aus.
- Replace a failed stack erstellt oder aktualisiert einen Stack, wenn der angegebene Stack nicht vorhanden ist. Wenn der Stapel existiert und sich in einem ausgefallenen Zustand befindet (gemeldet als `ROLLBACK_COMPLETE` `ROLLBACK_FAILEDCREATE_FAILED`,`DELETE_FAILED`,, oder`UPDATE_ROLLBACK_FAILED`), AWS CloudFormation wird der Stapel gelöscht und anschließend ein neuer erstellt. Wenn sich der Stack nicht in einem ausgefallenen Zustand befindet, wird er AWS CloudFormation aktualisiert. Verwenden Sie diese Aktion, um fehlerhafte Stacks ohne Wiederherstellung oder Fehlerbehebung zu ersetzen. In der Regel wählen Sie diesen Modus für Tests.

Stack name

Der Name, der einem vorhandenen Stack oder einem Stack zugeordnet ist, den Sie erstellen möchten. Der Name muss in der AWS Region, in der Sie den Stack erstellen, eindeutig sein.

> **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.

Change set name (Name des Änderungssatzes)

Der Name eines vorhandenen Änderungssatzes oder eines neuen Änderungssatzes, den Sie für den angegebenen Stack erstellen möchten.

Vorlage

Der Speicherort einer AWS CloudFormation Vorlagendatei, die dem Format folgt`ArtifactName::TemplateFileName`.

Vorlagenkonfiguration

Der Speicherort einer Vorlagenkonfigurationsdatei im Format *`ArtifactName`*`::`*`TemplateConfigurationFileName`*. Die Vorlagenkonfigurationsdatei kann Vorlagenparameterwerte, eine Stack-Richtlinie und Tags enthalten. Wenn Sie sensible Daten wie z. B. Passwörter einschließen, sollten Sie den Zugriff auf diese Datei beschränken. Weitere Informationen finden Sie unter AWS CloudFormation Artefakte.

Funktionen

Bei Stacks, die bestimmte Ressourcen enthalten, ausdrückliche Bestätigung, dass diese Ressourcen AWS CloudFormation möglicherweise erstellt oder aktualisiert werden. Sie müssen beispielsweise `CAPABILITY_IAM` angeben, wenn die Stack-Vorlage AWS Identity and Access Management (IAM)-Ressourcen enthält. Weitere Informationen finden Sie unter Anforderungsparameter für CreateStackAPI-Operationen.

Wenn Sie IAM-Ressourcen in Ihrer Stack-Vorlage verwenden, müssen Sie diese Eigenschaft angeben.

Sie können mehr als eine Funktion angeben.

Name der Rolle

Der Name der IAM-Dienstrolle, die AWS CloudFormation annimmt, wenn sie mit Ressourcen im angegebenen Stack arbeitet.

Ausgabedateiname

Im Abschnitt „Erweitert“ können Sie einen Namen für die Ausgabedatei angeben, z. B. den Namen`CreateStackOutput.json`, der dem Ausgabeartefakt CodePipeline hinzugefügt wird, nachdem es die angegebene Aktion ausgeführt hat. Das Ausgabeartefakt enthält eine JSON-Datei mit dem Inhalt des `Outputs` Abschnitts der AWS CloudFormation Vorlage.

Wenn Sie keinen Namen angeben, wird CodePipeline kein Ausgabeartefakt generiert.

Parameter überschreiben

Parameter werden in Ihrer Vorlage definiert und ermöglichen Ihnen, benutzerdefinierte Werte einzugeben, wenn Sie einen Stack erstellen oder aktualisieren. Sie können ein JSON-Objekt angeben, das Parameterwerte für die Vorlage in der Vorlagenkonfigurationsdatei überschreibt. Alle Parameternamen müssen in der Stack-Vorlage vorhanden sein. Weitere Informationen finden Sie unter CloudFormation ParametersVorlagensyntax.

> **Note**
>
> Für das JSON-Objekt, das in der Eigenschaft `ParameterOverrides` gespeichert werden kann, gilt eine Maximalgröße von 1 Kilobyte.

Wir empfehlen, dass Sie die Vorlagenkonfigurationsdatei verwenden, um die meisten Ihrer Parameterwerte anzugeben. Verwenden Sie Parameterüberschreibungen, um nur dynamische Parameterwerte anzugeben. Dynamische Parameter sind unbekannt, bis Sie die Pipeline ausführen.

Das folgende Beispiel definiert einen Wert für den `ParameterName`-Parameter mithilfe einer Parameterüberschreibungsfunktion. Die Funktion ruft einen Wert aus einem CodePipeline Eingabeartefakt ab. Weitere Informationen zu Parameterüberschreibungsfunktionen finden Sie unter Verwenden von Funktionen zum Überschreiben von Parametern mit CodePipeline Pipelines.

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
 "ParamName"]}
}
```

Konfigurationseigenschaften (JSON-Objekt)

Wenn Sie `CloudFormation` als Anbieter für eine Phasenaktion angeben, definieren Sie die folgenden Eigenschaften in der `Configuration`-Eigenschaft. Verwenden Sie das JSON-Objekt für die AWS CLI, CodePipeline API oder AWS CloudFormation Vorlagen. Beispiele finden Sie unter Walkthrough: Erstellen einer Pipeline für Test- und Produktions-Stacks und AWS CloudFormation Referenz zu den Konfigurationseigenschaften.

`ActionMode`

Die AWS CloudFormation Aktion, die CodePipeline aufgerufen wird, wenn sie die zugehörige Phase verarbeitet. Geben Sie nur einen der folgenden Aktionsmodi an:

- `CHANGE_SET_EXECUTE` führt einen Änderungssatz aus.
- `CHANGE_SET_REPLACE` erstellt den Änderungssatz, falls er noch nicht existiert, basierend auf dem Stack-Namen und der Vorlage, die Sie übermitteln. Wenn der Änderungssatz vorhanden ist, AWS CloudFormation wird er gelöscht und anschließend ein neuer erstellt.

- `CREATE_UPDATE` erstellt den Stack, wenn der angegebene Stack noch nicht vorhanden ist. Wenn der Stack bereits vorhanden ist, wird er von AWS CloudFormation aktualisiert. Verwenden Sie diese Aktion, um bestehende Stapel zu aktualisieren. CodePipeline ersetzt den Stapel nicht.
- `DELETE_ONLY` löscht einen Stack. Wenn Sie einen Stack angeben, der nicht vorhanden ist, wird die Aktion ohne Löschen eines Stacks erfolgreich abgeschlossen.
- `REPLACE_ON_FAILURE` erstellt einen Stack, wenn der angegebene Stack noch nicht vorhanden ist. Wenn der Stapel existiert und sich in einem ausgefallenen Zustand befindet (gemeldet als `ROLLBACK_COMPLETEROLLBACK_FAILED`,`CREATE_FAILED`,`DELETE_FAILED`, oder`UPDATE_ROLLBACK_FAILED`), AWS CloudFormation wird der Stapel gelöscht und anschließend ein neuer Stapel erstellt. Wenn sich der Stack nicht in einem ausgefallenen Zustand befindet, wird er AWS CloudFormation aktualisiert. Verwenden Sie diese Aktion, um fehlerhafte Stacks automatisch ohne Wiederherstellung oder Fehlerbehebung zu ersetzen. In der Regel wählen Sie diesen Modus für Tests.

Diese Eigenschaft ist erforderlich.

`Capabilities`

Bei Stacks, die bestimmte Ressourcen enthalten, ausdrückliche Bestätigung, dass diese Ressourcen AWS CloudFormation möglicherweise erstellt oder aktualisiert werden. Sie müssen beispielsweise `CAPABILITY_IAM` angeben, wenn die Stack-Vorlage AWS Identity and Access Management (IAM)-Ressourcen enthält. Weitere Informationen finden Sie unter Anforderungsparameter für CreateStackAPI-Operationen.

Diese Eigenschaft ist bedingt erforderlich. Wenn Sie IAM-Ressourcen in Ihrer Stack-Vorlage verwenden, müssen Sie diese Eigenschaft angeben.

Sie können mehrere Funktionen angeben. Im folgenden Beispiel werden der Vorlage die `CAPABILITY_AUTO_EXPAND` Eigenschaften `CAPABILITY_IAM` und hinzugefügt:

YAML

```
configuration:
  ActionMode: CHANGE_SET_REPLACE
  Capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
  ChangeSetName: pipeline-changeset
  RoleArn: CloudFormation_Role_ARN
  StackName: my-pipeline-stack
  TemplateConfiguration: 'my-pipeline-stack::template-configuration.json'
```

```
  TemplatePath: 'my-pipeline-stack::template-export.yml'
```

JSON

```
 "configuration": {
        "ActionMode": "CHANGE_SET_REPLACE",
        "Capabilities": "CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND",
        "ChangeSetName": "pipeline-changeset",
        "RoleArn": "CloudFormation_Role_ARN",
        "StackName": "my-pipeline-stack",
        "TemplateConfiguration": "my-pipeline-stack::template-
configuration.json",
        "TemplatePath": "my-pipeline-stack::template-export.yml"
    }
```

`ChangeSetName`

Der Name eines vorhandenen Änderungssatzes oder eines neuen Änderungssatzes, den Sie für den angegebenen Stack erstellen möchten.

Die Eigenschaft ist für die folgenden Aktionsmodi erforderlich: `CHANGE_SET_REPLACE` und `CHANGE_SET_EXECUTE`. Bei allen anderen Aktionsmodi wird die Eigenschaft ignoriert.

`OutputFileName`

Ein Name für die Ausgabedatei, z. `CreateStackOutput.json` B. CodePipeline fügt die Datei dem Ausgabeartefakt hinzu, nachdem es die angegebene Aktion ausgeführt hat. Das Ausgabeartefakt enthält eine JSON-Datei mit dem Inhalt des `Outputs` Abschnitts der AWS CloudFormation Vorlage.

Diese Eigenschaft ist optional. Wenn Sie keinen Namen angeben, wird CodePipeline kein Ausgabeartefakt generiert.

`ParameterOverrides`

Parameter werden in Ihrer Vorlage definiert und ermöglichen Ihnen, benutzerdefinierte Werte einzugeben, wenn Sie einen Stack erstellen oder aktualisieren. Sie können ein JSON-Objekt angeben, das Parameterwerte für die Vorlage in der Vorlagenkonfigurationsdatei überschreibt. Alle Parameternamen müssen in der Stack-Vorlage vorhanden sein. Weitere Informationen finden Sie unter CloudFormation ParametersVorlagensyntax.

Im folgenden Beispiel werden die Parameterüberschreibungen `InstanceType` und `KeyName` zur Vorlage hinzugefügt:

YAML

```
configuration:
  ActionMode: CHANGE_SET_REPLACE
  Capabilities: CAPABILITY_NAMED_IAM
  ChangeSetName: pipeline-changeset
  ParameterOverrides: '{"InstanceType": "t2.small","KeyName": "my-keypair"}'
  RoleArn: CloudFormation_Role_ARN
  StackName: my-pipeline-stack
  TemplateConfiguration: 'my-pipeline-stack::template-configuration.json'
  TemplatePath: 'my-pipeline-stack::template-export.yml'
```

JSON

```
"configuration": {
        "ActionMode": "CHANGE_SET_REPLACE",
        "Capabilities": "CAPABILITY_NAMED_IAM",
        "ChangeSetName": "pipeline-changeset",
        "ParameterOverrides": "{\"InstanceType\": \"t2.small\",\"KeyName\": \"my-
keypair\"}",
        "RoleArn": "CloudFormation_Role_ARN",
        "StackName": "my-pipeline-stack",
        "TemplateConfiguration": "my-pipeline-stack::template-
configuration.json",
        "TemplatePath": "my-pipeline-stack::template-export.yml"
    }
```

> **Note**
>
> Die maximale Größe für das JSON-Objekt, das in der `ParameterOverrides`-Eigenschaft gespeichert werden kann, beträgt 1 Kilobyte.

Wir empfehlen, dass Sie die Vorlagenkonfigurationsdatei verwenden, um die meisten Ihrer Parameterwerte anzugeben. Verwenden Sie Parameterüberschreibungen, um nur dynamische Parameterwerte anzugeben. Dynamische Parameterwerte sind unbekannt, bis Sie die Pipeline ausführen.

Das folgende Beispiel definiert einen Wert für den `ParameterName`-Parameter mithilfe einer Parameterüberschreibungsfunktion. Die Funktion ruft einen Wert aus einem CodePipeline

Eingabeartefakt ab. Weitere Informationen zu Parameterüberschreibungsfunktionen finden Sie unter Verwenden von Funktionen zum Überschreiben von Parametern mit CodePipeline Pipelines.

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
 "ParamName"]}
}
```

Diese Eigenschaft ist optional.

`RoleArn`

Der Amazon-Ressourcenname (ARN) der IAM-Servicerolle, die AWS CloudFormation annimmt, wenn sie mit Ressourcen in einem Stack arbeitet.

Diese Eigenschaft ist für die folgenden Aktionsmodi erforderlich: `CREATE_UPDATE`, `REPLACE_ON_FAILURE`, `DELETE_ONLY` und `CHANGE_SET_REPLACE`. `RoleArn` wird nicht angewendet, wenn ein Änderungssatz ausgeführt wird. Wenn Sie den Änderungssatz nicht CodePipeline zur Erstellung verwenden, stellen Sie sicher, dass dem Änderungssatz oder Stack eine zugeordnete Rolle zugeordnet ist.

`StackName`

Der Name eines vorhandenen Stacks oder eines Stacks, den Sie erstellen möchten.

Die Eigenschaft ist für alle Aktionsmodi erforderlich.

`TemplateConfiguration`

`TemplateConfiguration` ist die Vorlagenkonfigurationsdatei. Sie schließen die Datei in ein Eingabeartefakt für diese Aktion ein. Der Dateiname der Vorlagenkonfiguration hat das folgende Format:

`Artifactname``::`*`TemplateConfigurationFileName`*

`Artifactname`ist der Name des Eingabeartefakts, wie er in CodePipeline erscheint. So erstellt eine Quellstufe mit dem Artefaktnamen `SourceArtifact` und dem Dateinamen `test-configuration.json` einen `TemplateConfiguration`-Namen, wie in diesem Beispiel gezeigt:

```
"TemplateConfiguration": "SourceArtifact::test-configuration.json"
```

Die Vorlagenkonfigurationsdatei kann Vorlagenparameterwerte und eine Stack-Richtlinie enthalten. Wenn Sie sensible Daten wie z. B. Passwörter einschließen, sollten Sie den Zugriff auf diese Datei beschränken. Für eine Beispiel-Vorlagenkonfigurationsdatei siehe AWS CloudFormation Artefakte.

Diese Eigenschaft ist optional.

`TemplatePath`

`TemplatePath`steht für die AWS CloudFormation Vorlagendatei. Sie schließen die Datei in ein Eingabeartefakt für diese Aktion ein. Der Dateiname folgt diesem Format:

`Artifactname``::`*`TemplateFileName`*

`Artifactname`ist der Name des Eingabeartefakts, wie er in CodePipeline erscheint. So erstellt eine Quellstufe mit dem Artefaktnamen `SourceArtifact` und dem Dateinamen `template.yaml` einen `TemplatePath`-Namen, wie in diesem Beispiel gezeigt:

```
"TemplatePath": "SourceArtifact::template.yaml"
```

Die Eigenschaft wird für die folgenden Aktionsmodi benötigt: `CREATE_UPDATE`, `REPLACE_ON_FAILURE` und `CHANGE_SET_REPLACE`. Bei allen anderen Aktionsmodi wird die Eigenschaft ignoriert.

Weitere Informationen finden Sie auch unter

Die folgenden verwandten Ressourcen unterstützen Sie bei der Arbeit mit diesen Parametern.

- Weitere Informationen zu den CloudFormation Aktionsparametern in CodePipeline finden Sie in der Konfigurationsreferenz für AWS CloudFormation Bereitstellungsaktionen im AWS CodePipeline Benutzerhandbuch.
- Beispiels-Vorlagenwerte nach Aktionsanbietern, wie etwa für das `Owner`-Feld oder die `configuration`-Felder, finden Sie in der Referenz zur Aktionsstruktur im Benutzerhandbuch für AWS CodePipeline .
- Beispiele für Pipeline-Stack-Vorlagen im YAML- oder JSON-Format finden Sie AWS CloudFormation im AWS CodePipeline Benutzerhandbuch unter Tutorial: Eine Pipeline erstellen mit.

AWS CloudFormation Artefakte

CodePipeline führt Aufgaben an Artefakten aus, während eine Pipeline CodePipeline ausgeführt wird. Denn AWS CloudFormation Artefakte können eine Stack-Vorlagendatei, eine Vorlagen-Konfigurationsdatei oder beides enthalten. CodePipeline verwendet diese Artefakte, um mit AWS CloudFormation Stacks und Änderungssätzen zu arbeiten.

Wenn Sie Amazon Simple Storage Service (Amazon S3) als Quell-Repository verwenden, müssen Sie die Vorlagen- und Vorlagenkonfigurationsdateien in einer einzigen ZIP-Datei komprimieren, bevor Sie sie in einen S3-Bucket hochladen. Laden Sie für andere Repositorys, wie z. B. GitHub und AWS CodeCommit, Artefakte hoch, ohne sie zu komprimieren. Weitere Informationen finden Sie im AWS CodePipeline Benutzerhandbuch unter Erstellen einer Pipeline, Phasen und Aktionen.

Sie können so viele Dateien wie nötig zu Ihrem Repository hinzufügen. So können Sie beispielsweise zwei verschiedene Konfigurationen für dieselbe Vorlage einschließen: eine für eine Testkonfiguration und eine andere für eine Produktionskonfiguration.

In diesem Thema werden die einzelnen Artefakttypen beschrieben.

Themen

- Stack-Vorlagendatei
- Vorlagenkonfigurationsdatei
- Weitere Informationen finden Sie auch unter

Stack-Vorlagendatei

Eine Stack-Vorlagendatei definiert die Ressourcen, die AWS CloudFormation bereitgestellt und konfiguriert werden. Bei diesen Dateien handelt es sich um dieselben Vorlagendateien, die Sie verwenden, wenn Sie Stacks mithilfe von erstellen oder aktualisieren. AWS CloudFormation Sie können mit YAML oder JSON formatierte Vorlagen verwenden. Weitere Informationen zu Vorlagen finden Sie unter CloudFormation Vorlagenabschnitte.

Vorlagenkonfigurationsdatei

Eine Vorlagenkonfigurationsdatei ist eine JSON-formatierte Textdatei, die Vorlagenparameterwerte, eine Stack-Richtlinie und Tags angeben kann. Verwenden Sie diese Konfigurationsdateien, um Parameterwerte oder eine Stack-Richtlinie für einen Stack anzugeben. Alle Parameterwerte, die Sie angeben, müssen in der zugehörigen Vorlage deklariert sein.

Wenn Sie sensible Daten wie z. B. Passwörter einschließen, sollten Sie den Zugriff auf diese Datei beschränken. Wenn Sie beispielsweise Ihr Artefakt in einen S3-Bucket hochladen, verwenden Sie S3-Bucket-Richtlinien oder Benutzerrichtlinien zum Einschränken des Zugriffs.

Verwenden Sie zum Erstellen einer Konfigurationsdatei das folgende Format:

```
{
  "Parameters" : {
    "NameOfTemplateParameter" : "ValueOfParameter",
    ...
  },
  "Tags" : {
    "TagKey" : "TagValue",
    ...
  },
  "StackPolicy" : {
    "Statement" : [
      StackPolicyStatement
    ]
  }
}
```

Im folgenden Beispiel wird `TestEC2Key` für den Parameter `KeyName` angegeben und ein `Department`-Tag mit dem Wert `Marketing` sowie eine Stack-Richtlinie hinzugefügt, die alle Aktualisierungsaktionen außer einer Aktualisierung zulässt, die eine Ressource löscht.

```
{
  "Parameters" : {
    "KeyName" : "TestEC2Key"
  },
  "Tags" : {
    "Department" : "Marketing"
  },
  "StackPolicy" : {
    "Statement" : [
      {
        "Effect" : "Allow",
        "NotAction" : "Update:Delete",
        "Principal": "*",
        "Resource" : "*"
      }
    ]
```

```
  }
}
```

Weitere Informationen finden Sie auch unter

Die folgenden verwandten Ressourcen unterstützen Sie bei der Arbeit mit diesen Parametern.

- Weitere Informationen zu den CloudFormation Aktionsparametern in CodePipeline finden Sie in der Konfigurationsreferenz für AWS CloudFormation Bereitstellungsaktionen im AWS CodePipeline Benutzerhandbuch.
- Beispiels-Vorlagenwerte nach Aktionsanbietern, wie etwa für das `Owner`-Feld oder die `configuration`-Felder, finden Sie in der Referenz zur Aktionsstruktur im Benutzerhandbuch für AWS CodePipeline .
- Beispiele für Pipeline-Stack-Vorlagen im YAML- oder JSON-Format finden Sie AWS CloudFormation im AWS CodePipeline Benutzerhandbuch unter Tutorial: Eine Pipeline erstellen mit.

Verwenden von Funktionen zum Überschreiben von Parametern mit CodePipeline Pipelines

In einer CodePipeline Phase können Sie Parameterüberschreibungen für AWS CloudFormation Aktionen angeben. Mit Parameterüberschreibungen können Sie Vorlagenparameterwerte angeben, die Werte in einer Vorlagenkonfigurationsdatei überschreiben. AWS CloudFormation stellt Funktionen bereit, mit denen Sie dynamische Werte angeben können (Werte, die erst bekannt sind, wenn die Pipeline ausgeführt wird).

Themen

- Fn::GetArtifactAtt
- Fn::GetParam
- Weitere Informationen finden Sie auch unter

`Fn::GetArtifactAtt`

Die `Fn::GetArtifactAtt`-Funktion ruft den Wert eines Attributs aus einem Eingabeartefakt ab, z. B. den Namen des S3-Buckets, in dem das Artefakt gespeichert ist. Verwenden Sie diese Funktion, um Attribute eines Artefakts anzugeben, z. B. seinen Datei- oder Amazon S3 Bucket-Namen.

Wenn Sie eine Pipeline ausführen, CodePipeline kopiert und schreibt sie in den Artefaktspeicher der Pipeline (einen S3-Bucket). CodePipeline generiert die Dateinamen im Artefaktspeicher. Diese Dateinamen sind nicht bekannt, bevor Sie die Pipeline ausführen.

In Ihrer Pipeline könnten Sie beispielsweise eine Quellstufe haben, in der der Quellcode Ihrer AWS Lambda Funktion in den Artefaktspeicher CodePipeline kopiert wird. In der nächsten Phase haben Sie eine AWS CloudFormation Vorlage, die die Lambda-Funktion erstellt, aber den Dateinamen AWS CloudFormation benötigt, um die Funktion zu erstellen. Sie müssen die `Fn::GetArtifactAtt`-Funktion verwenden, um den genauen S3-Bucket- und Dateinamen zu übergeben.

Syntax

Verwenden Sie die folgende Syntax zum Abrufen eines Attributwerts eines Artefakts.

```
{ "Fn::GetArtifactAtt" : [ "artifactName", "attributeName" ] }
```

`artifactName`

Der Name des Eingabeartefakts. Sie müssen dieses Artefakt als Eingabe für die zugehörige Aktion deklarieren.

`attributeName`

Der Name des Artefaktattributs, dessen Wert Sie abrufen möchten. Details zu den einzelnen Artefakten finden Sie im folgenden Abschnitt "Attribute".

Beispiel

Die folgenden Parameter-Überschreibungen geben die Parameter `BucketName` und `ObjectKey` durch Abrufen des S3-Bucket-Namens und Dateinamens des `LambdaFunctionSource`-Artefakts an. In diesem Beispiel wird davon ausgegangen, dass der Quellcode der Lambda-Funktion CodePipeline kopiert und als Artefakt gespeichert wurde, beispielsweise als Teil einer Quellstufe.

```
{
  "BucketName" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "BucketName"]},
  "ObjectKey" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "ObjectKey"]}
}
```

Attribute

Sie können die folgenden Attribute für ein Artefakt abrufen.

`BucketName`

Der Name des S3-Buckets, in dem das Artefakt gespeichert ist.

`ObjectKey`

Der Name der `.zip` Datei, die das Artefakt enthält, das von generiert wurde CodePipeline, z. B. `1ABCyZZ.zip`

`URL`

Die Amazon Simple Storage Service (Amazon S3) URL des Artefakts, z. B. `https://s3.us-west-2.amazonaws.com/artifactstorebucket-yivczw8jma0c/test/TemplateSo/1ABCyZZ.zip`.

Fn::GetParam

Die `Fn::GetParam`-Funktion gibt einen Wert aus einem Schlüssel-Wert-Paar in einer JSON-formatierten Datei zurück. Die JSON-Datei muss in ein Artefakt eingeschlossen sein.

Verwenden Sie diese Funktion, um Ausgabewerte aus einem AWS CloudFormation Stapel abzurufen und sie als Eingabe für eine andere Aktion zu verwenden. Wenn Sie beispielsweise einen Ausgabedateinamen für eine AWS CloudFormation Aktion angeben, CodePipeline speichern Sie die Ausgabe in einer JSON-Datei und fügen sie dann der `.zip` Datei des Ausgabeartefakts hinzu. Verwenden Sie die `Fn::GetParam`-Funktion, um den Ausgabewert abzurufen, und verwenden Sie ihn als Eingabe für eine andere Aktion.

Syntax

Verwenden Sie die folgende Syntax, um einen Wert aus einem Schlüssel-Wert-Paar abzurufen.

```
{ "Fn::GetParam" : [ "artifactName", "JSONFileName", "keyName" ] }
```

`artifactName`

Der Name des Artefakts, das als Eingabeartefakt für die zugehörige Aktion eingeschlossen werden muss.

`JSONFileName`

Der Name einer JSON-Datei, die in dem Artefakt enthalten ist.

`keyName`

Der Name des Schlüssels, dessen Wert Sie abrufen möchten.

Beispiele

Die folgenden Beispiele zeigen, wie die `Fn::GetParam`-Funktion in einer Parameterüberschreibung verwendet wird.

Syntax

Die folgende Parameterüberschreibung gibt den `WebSiteURL`-Parameter an, indem der Wert des Schlüssels URL aus der Datei `stack-output.json` abgerufen wird, die im Artefakt `WebStackOutput` enthalten ist.

```
{
  "WebSiteURL" : { "Fn::GetParam" : ["WebStackOutput", "stack-output.json", "URL"]}
}
```

AWS CloudFormation Schnipsel aus Vorlagen

Die folgenden AWS CloudFormation Vorlagenausschnitte aus einer CodePipeline Pipeline zeigen, wie Stack-Ausgaben übergeben werden. Diese Codeausschnitte zeigen zwei Phasen der Pipeline-Definition. In der ersten Phase wird ein Stack erstellt und dessen Ausgaben werden in der Datei `TestOutput.json` im Artefakt `StackAOutput` gespeichert. Diese Werte werden mit den Eigenschaften `OutputFileName` und `OutputArtifacts` angegeben.

Der Name des Quelleingangsartefakts für die Phasen lautet `TemplateSource`. Der Dateiname für die Stack-Vorlage lautet `teststackA.yaml` und der Dateiname für die Konfigurationsdatei `test-configuration.json`. In beiden Phasen werden diese Werte wie dargestellt für die Eigenschaften `TemplateConfiguration` und `TemplatePath` angegeben.

```
TemplateConfiguration: TemplateSource::test-configuration.json
TemplatePath: TemplateSource::teststackA.yaml
```

Example Erstellen der Stack A-Phase

```
- Name: CreateTestStackA
  Actions:
```

```
    - Name: CloudFormationCreate
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      Configuration:
        ActionMode: CREATE_UPDATE
        Capabilities: CAPABILITY_IAM
        OutputFileName: TestOutput.json
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: StackA
        TemplateConfiguration: TemplateSource::test-configuration.json
        TemplatePath: TemplateSource::teststackA.yaml
      InputArtifacts:
        - Name: TemplateSource
      OutputArtifacts:
        - Name: StackAOutput
      RunOrder: '1'
```

In einer nachfolgenden Phase verwendet Stack B die Ausgaben von Stack A. In der `ParameterOverrides`-Eigenschaft wird im Beispiel die `Fn::GetParam`-Funktion zur Angabe des `StackBInputParam`-Parameters verwendet. Die resultierende Wert ist der Wert, der dem Schlüssel `StackAOutputName` zugeordnet ist.

Example Erstellen der Stack B-Phase

```
- Name: CreateTestStackB
  Actions:
    - Name: CloudFormationCreate
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      Configuration:
        ActionMode: CREATE_UPDATE
        Capabilities: CAPABILITY_IAM
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: StackB
        TemplateConfiguration: TemplateSource::test-configuration.json
        TemplatePath: TemplateSource::teststackB.yaml
        ParameterOverrides: |
```

```
          {
            "StackBInputParam" : { "Fn::GetParam" : ["StackAOutput", "TestOutput.json",
"StackAOutputName"]}
          }
      InputArtifacts:
        - Name: TemplateSource
        - Name: StackAOutput
      RunOrder: '1'
```

Weitere Informationen finden Sie auch unter

Die folgenden verwandten Ressourcen unterstützen Sie bei der Arbeit mit diesen Parametern.

- Weitere Informationen zu den CloudFormation Aktionsparametern in CodePipeline finden Sie in der Konfigurationsreferenz für AWS CloudFormation Bereitstellungsaktionen im AWS CodePipeline Benutzerhandbuch.
- Beispiels-Vorlagenwerte nach Aktionsanbietern, wie etwa für das `Owner`-Feld oder die `configuration`-Felder, finden Sie in der Referenz zur Aktionsstruktur im Benutzerhandbuch für AWS CodePipeline .
- Beispiele für Pipeline-Stack-Vorlagen im YAML- oder JSON-Format finden Sie AWS CloudFormation im AWS CodePipeline Benutzerhandbuch unter Tutorial: Eine Pipeline erstellen mit.

Das AWS CloudFormation Vorlagen-Referenzhandbuch

Detaillierte Informationen zu den Ressourcen und Komponenten, die für die Erstellung von AWS CloudFormation Vorlagen verfügbar sind, finden Sie jetzt im Referenzhandbuch für AWS CloudFormation Vorlagen. Dieses Handbuch dient als technische Referenz für Vorlagen und ergänzt die Inhalte für die ersten Schritte im CloudFormation Benutzerhandbuch.

Das AWS CloudFormation Vorlagen-Referenzhandbuch enthält Einzelheiten zu:

- Ressourcentypen und ihre Eigenschaften
- `DeletionPolicy`und andere Attribute auf Ressourcenebene
- Intrinsische Funktionen wie und `Ref` `Sub`
- Transformationen (Makros) CloudFormation
- Hilfsskripte für EC2 die Instanzkonfiguration
- Maschinenlesbare Definitionen von Ressourcentypen

Dieser Leitfaden wird regelmäßig aktualisiert, um Folgendes zu berücksichtigen:

- Neue Ressourcentypen
- Änderungen an der Immobilie
- Neue Rückgabewerte
- Aktualisierungen der Dokumentation

Sicherheit in AWS CloudFormation

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für gelten AWS CloudFormation, finden Sie unter AWS Services im Umfang nach Compliance-Programmen AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden können AWS CloudFormation. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen AWS CloudFormation , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer AWS CloudFormation Ressourcen unterstützen.

Themen

- CloudFormation Stapel vor dem Löschen schützen
- Verhindern von Aktualisierungen der Stack-Ressourcen
- Datenschutz in AWS CloudFormation
- Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management
- AWS CloudFormation API-Aufrufe protokollieren mit AWS CloudTrail
- Infrastruktursicherheit in AWS CloudFormation
- Resilienz in AWS CloudFormation
- Konformitätsvalidierung für AWS CloudFormation
- Konfiguration und Schwachstellenanalyse in AWS CloudFormation

- Bewährte Sicherheitsmethoden für CloudFormation
- Zugriff CloudFormation über einen Schnittstellenendpunkt (AWS PrivateLink)

CloudFormation Stapel vor dem Löschen schützen

Sie können das versehentliche Löschen eines Stacks verhindern, indem Sie den Beendigungsschutz des Stacks aktivieren. Wenn ein Benutzer versucht, einen Stack mit aktiviertem Beendigungsschutz zu löschen, schlägt das Löschen fehl und der Stack – einschließlich seines Status – bleibt unverändert. Sie können den Beendigungsschutz aktivieren, wenn Sie den Stack erstellen. Der Beendigungsschutz von Stacks ist standardmäßig deaktiviert. Sie können den Kündigungsschutz für einen Stack mit einem beliebigen Status außer `DELETE_IN_PROGRESS` oder `DELETE_COMPLETE` einrichten.

Wenn Sie den Terminierungsschutz für einen Stack aktivieren oder deaktivieren, wird dieselbe Auswahl auch auf alle verschachtelten Stacks übertragen, die zu diesem Stack gehören. Sie können den Beendigungsschutz nicht direkt für einen verschachtelten Stack aktivieren oder deaktivieren. Wenn ein Benutzer versucht, einen verschachtelten Stack zu löschen, der zu einem Stack mit aktiviertem Beendigungsschutz gehört, schlägt die Operation fehl und der verschachtelte Stack bleibt unverändert.

Wenn ein Benutzer jedoch eine Stack-Aktualisierung vornimmt, die zum Löschen eines verschachtelten Stacks führt, löscht AWS CloudFormation den betreffenden verschachtelten Stack.

Der Beendigungsschutz ist nicht mit dem Deaktivieren von Rollbacks identisch. Der Beendigungsschutz bezieht sich nur auf Versuche, den Stack zu löschen, während das Deaktivieren von Rollbacks für das automatische Rollback gilt, das beim Fehlschlagen der Stack-Erstellung erfolgt.

So aktivieren Sie den Beendigungsschutz beim Erstellen eines Stacks

Erweitern Sie auf der Seite Specify stack options (Stack-Optionen angeben) des Assistenten Create stack (Stack erstellen) unter Advanced options (Erweiterte Optionen) den Abschnitt Termination Protection (Beendigungsschutz) und wählen Sie Enable (Aktivieren) aus. Weitere Informationen finden Sie unter Konfigurieren Sie die Stack-Optionen.

So aktivieren oder deaktivieren Sie den Beendigungsschutz eines vorhandenen Stacks

1. Melden Sie sich bei /cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole. https://console.aws.amazon.com
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region

3. Wählen Sie den gewünschten Stack.

> **Note**
>
> Wenn neben dem Stack-Namen NESTED angezeigt wird, handelt es sich um einen verschachtelten Stack. Sie können nur den Beendigungsschutz des Root-Stacks ändern, zu dem der verschachtelte Stack gehört.

4. Wählen Sie im Stack-Detailbereich die Option Stack actions (Stack-Aktionen) und dann Edit termination protection (Beendigungsschutz bearbeiten) aus.

   CloudFormation zeigt das Dialogfeld Kündigungsschutz bearbeiten an.
5. Wählen Sie Enable (Aktivieren) oder Disable (Deaktivieren) und anschließend Save (Speichern) aus.

So aktivieren oder deaktivieren Sie den Beendigungsschutz eines verschachtelten Stacks

Wenn neben dem Stack-Namen NESTED angezeigt wird, handelt es sich um einen verschachtelten Stack. Sie können nur den Beendigungsschutz des Root-Stacks ändern, zu dem der verschachtelte Stack gehört. So ändern Sie den Beendigungsschutz des Root-Stacks:

1. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm Ihre aus. AWS-Region
3. Wählen Sie den gewünschten verschachtelten Stack.
4. Wählen Sie im Abschnitt Stack info (Stack-Informationen) unter Overview (Übersicht) den als Root stack (Root-Stack) aufgeführten Stack-Namen aus.

   CloudFormation zeigt die Stack-Details für den Root-Stack an.
5. Wählen Sie Stack actions (Stack-Aktionen) und dann Edit termination protection (Beendigungsschutz bearbeiten) aus.

   CloudFormation zeigt das Dialogfeld Kündigungsschutz bearbeiten an.
6. Wählen Sie Enable (Aktivieren) oder Disable (Deaktivieren) und anschließend Save (Speichern) aus.

Aktivieren oder Deaktivieren des Beendigungsschutzes mithilfe der Befehlszeile

Verwenden Sie den update-termination-protection-Befehl.

Benutzer festlegen, die den Beendigungsschutz von Stacks ändern dürfen

Zum Aktivieren oder Deaktivieren des Beendigungsschutzes von Stacks benötigt der Benutzer die Berechtigung für die Aktion `cloudformation:UpdateTerminationProtection`. Die Richtlinie unten erlaubt Benutzern beispielsweise das Aktivieren und Deaktivieren des Beendigungsschutzes von Stacks.

Weitere Informationen zum Angeben von Berechtigungen in AWS CloudFormation finden Sie unterSteuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management.

Example Beispielrichtlinie, die Berechtigungen zum Ändern des Beendigungsschutzes von Stacks gewährt

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Allow",
        "Action":[
            "cloudformation:UpdateTerminationProtection"
        ],
        "Resource":"*"
    }]
}
```

Verhindern von Aktualisierungen der Stack-Ressourcen

Wenn Sie ein Stack erstellen, sind alle Aktualisierungsaktionen für alle Ressourcen zulässig. Standardmäßig können alle Personen, die berechtigt sind, Stack-Aktualisierungen vorzunehmen, alle Ressourcen im Stack aktualisieren. Während eines Updates müssen einige Ressourcen möglicherweise unterbrochen oder vollständig ersetzt werden, was zu neuem physischem IDs oder völlig neuem Speicher führt. Sie können mithilfe einer Stack-Richtlinie verhindern, dass Stack-Ressourcen während einer Stack-Aktualisierung unabsichtlich aktualisiert oder gelöscht werden. Eine Stack-Richtlinie ist ein JSON-Dokument, in dem die Aktualisierungsaktionen definiert werden, die für designierte Ressourcen ausgeführt werden können.

Nachdem Sie eine Stack-Richtlinie festgelegt haben, sind alle Ressourcen im Stack standardmäßig geschützt. Um Aktualisierungen für bestimmte Ressourcen zuzulassen, geben Sie in der Stack-Richtlinie eine explizite `Allow`-Anweisung für diese Ressourcen an. Sie können nur eine Stack-Richtlinie pro Stack definieren, aber mehrere Ressourcen in einer einzelnen Richtlinie schützen. Eine Stack-Richtlinie gilt für alle CloudFormation-Benutzer, die versuchen, den Stack zu aktualisieren. Sie können unterschiedlichen Benutzern keinen verschiedenen Stack-Richtlinien zuordnen.

Eine Stack-Richtlinie gilt nur während Stack-Aktualisierungen. Sie bietet keine Zugriffskontrollen wie eine AWS Identity and Access Management (IAM)-Richtlinie. Verwenden Sie eine Stack-Richtlinie nur als ausfallsicheren Mechanismus, um versehentliche Aktualisierungen bestimmter Stack-Ressourcen zu verhindern. Verwenden Sie IAM, um den Zugriff auf AWS Ressourcen oder Aktionen zu steuern.

Themen

- Beispiel-Stack-Richtlinie
- Definieren einer Stack-Richtlinie
- Festlegen einer Stack-Richtlinie
- Aktualisieren geschützter Ressourcen
- Ändern einer Stack-Richtlinie
- Weitere Beispiel-Stack-Richtlinien

Beispiel-Stack-Richtlinie

Die folgende Beispiel-Stack-Richtlinie verhindert Aktualisierungen an der `ProductionDatabase`-Ressource:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/ProductionDatabase"
```

```
    }
  ]
}
```

Wenn Sie eine Stack-Richtlinie festlegen, sind alle Ressourcen standardmäßig geschützt. Um Aktualisierungen für alle Ressourcen zuzulassen, fügen wir eine `Allow`-Anweisung hinzu, die alle Aktionen für alle Ressourcen zulässt. Zwar gibt die `Allow`-Anweisung alle Ressourcen an, jedoch überschreibt die explizite `Deny`-Anweisung diese Anweisung für die Ressource mit der logischen ID `ProductionDatabase`. Diese `Deny`-Anweisung verhindert sämtliche Aktualisierungsaktionen, wie z. B. Ersetzung oder Löschung, für die `ProductionDatabase`-Ressource.

Das `Principal`-Element ist erforderlich, unterstützt aber nur den Platzhalter (`*`), was bedeutet, dass die Anweisung für alle Prinzipale gilt.

> **Note**
>
> Aktualisiert während eines Stack-Updates CloudFormation automatisch Ressourcen, die von anderen aktualisierten Ressourcen abhängig sind. CloudFormation Aktualisiert beispielsweise eine Ressource, die auf eine aktualisierte Ressource verweist. CloudFormation nimmt keine physischen Änderungen, wie z. B. die ID der Ressource, an automatisch aktualisierten Ressourcen vor. Wenn diesen Ressourcen jedoch eine Stack-Richtlinie zugeordnet ist, benötigen Sie die Berechtigung, sie zu aktualisieren.

Definieren einer Stack-Richtlinie

Wenn Sie ein Stack erstellen, ist keine Stack-Richtlinie festgelegt, sodass alle Aktualisierungsaktionen für alle Ressourcen zulässig sind. Um Stack-Ressourcen vor Aktualisierungsaktionen zu schützen, definieren Sie eine Stack-Richtlinie und legen Sie diese dann für den Stack fest. Eine Stack-Richtlinie ist ein JSON-Dokument, das die CloudFormation Stack-Aktualisierungsaktionen definiert, die CloudFormation Benutzer ausführen können, und die Ressourcen, für die die Aktionen gelten. Sie legen die Stack-Richtlinie bei der Erstellung eines Stacks fest, indem Sie eine Textdatei mit der Stack-Richtlinie angeben oder diese eingeben. Wenn Sie eine Stack-Richtlinie für den Stack festlegen, werden Aktualisierungen, die nicht ausdrücklich zugelassen sind, standardmäßig verweigert.

Sie definieren eine Stack-Richtlinie mit fünf Elemente: `Effect`, `Action`, `Principal`, `Resource` und `Condition`. Der folgende Pseudocode veranschaulicht die Stack-Richtlinien-Syntax.

```
{
  "Statement" : [
    {
      "Effect" : "Deny_or_Allow",
      "Action" : "update_actions",
      "Principal" : "*",
      "Resource" : "LogicalResourceId/resource_logical_ID",
      "Condition" : {
        "StringEquals_or_StringLike" : {
          "ResourceType" : [resource_type, ...]
        }
      }
    }
  ]
}
```

`Effect`

Bestimmt, ob die Aktionen, die Sie angeben, für die angegebene(n) Ressource(n) verweigert oder zugelassen werden. Sie können nur `Deny` oder `Allow` festlegen, wie z. B.:

```
"Effect" : "Deny"
```

> **Important**
>
> Wenn eine Stack-Richtlinie überlappende Anweisungen enthält (Zulassen und Verweigern von Aktualisierungen für eine Ressource), wird eine `Deny`-Anweisung immer durch eine `Allow`-Anweisung überschrieben. Um sicherzustellen, dass eine Ressource geschützt ist, verwenden Sie eine `Deny`-Anweisung für die betreffende Ressource.

Aktion

Gibt die Aktualisierungsaktionen an, die verweigert oder zugelassen werden:

Update:Modify

Gibt Aktualisierungsaktionen an, während denen Ressourcen möglicherweise nicht unterbrochen oder unterbrochen werden, während Änderungen angewendet werden. Alle Ressourcen behalten ihre physischen Werte bei IDs.

Update:Replace

Gibt Aktualisierungsaktionen an, bei denen Ressourcen neu erstellt werden. CloudFormation erstellt eine neue Ressource mit den angegebenen Updates und löscht dann die alte Ressource. Da die Ressource neu erstellt wird, hat die neue Ressourcen möglicherweise eine andere physische ID.

Update:Delete

Gibt Aktualisierungsaktionen an, während denen Ressourcen entfernt werden. Aktualisierungen, durch die Ressourcen vollständig aus einer Stack-Vorlage entfernt werden, erfordern diese Aktion.

Update:*

Gibt alle Aktualisierungsaktionen an. Das Sternchen ist ein Platzhalter, der für alle Aktualisierungsaktionen steht.

Im folgenden Beispiel wird gezeigt, wie Sie nur die Ersetzungs- und Löschungsaktionen angeben:

```
"Action" : ["Update:Replace", "Update:Delete"]
```

Um alle Aktualisierungsaktionen außer einer zuzulassen, verwenden Sie `NotAction`. Um beispielsweise alle Aktualisierungsaktionen außer `Update:Delete` zuzulassen, verwenden Sie `NotAction`, wie im folgenden Beispiel veranschaulicht:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "NotAction" : "Update:Delete",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Auftraggeber

Das `Principal`-Element gibt die Entität an, für die die Richtlinie gilt. Dieses Element ist erforderlich, unterstützt aber nur den Platzhalter (`*`), was bedeutet, dass die Richtlinie für alle Prinzipale gilt.

Ressource

Gibt die logische Struktur IDs der Ressourcen an, für die die Richtlinie gilt. Verwenden Sie das `Condition` Element, um Ressourcentypen anzugeben.

Um eine einzelne Ressource anzugeben, verwenden Sie deren logische ID. Zum Beispiel:

```
"Resource" : ["LogicalResourceId/myEC2instance"]
```

Sie können einen Platzhalter mit logischem Wert verwenden IDs. Wenn Sie beispielsweise ein gängiges logisches ID-Präfix für alle verwandten Ressourcen verwenden, können Sie alle mit einem Platzhalter angeben:

```
"Resource" : ["LogicalResourceId/CriticalResource*"]
```

Sie können auch ein `Not`-Element mit Ressourcen verwenden. Um beispielsweise Aktualisierungen für alle Ressourcen außer einer zuzulassen, verwenden Sie ein `NotResource`-Element, um diese Ressource zu schützen:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Wenn Sie eine Stack-Richtlinie festlegen, werden Aktualisierungen, die nicht ausdrücklich zugelassen sind, verweigert. Wenn Sie Aktualisierungen für alle Ressourcen außer der `ProductionDatabase`-Ressource zulassen, verweigern Sie Aktualisierungen der `ProductionDatabase`-Ressource.

Bedingungen

Gibt den Ressourcentyp an, für den die Richtlinie gilt. Verwenden Sie das `Resource` Element, um die Logik IDs bestimmter Ressourcen anzugeben.

Sie können einen Ressourcentyp angeben, z. B. all EC2 und RDS-DB-Instances, wie im folgenden Beispiel gezeigt:

```
{
  "Statement" : [
  {
    "Effect" : "Deny",
    "Principal" : "*",
    "Action" : "Update:*",
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "ResourceType" : ["AWS::EC2::Instance", "AWS::RDS::DBInstance"]
      }
    }
  },
  {
    "Effect" : "Allow",
    "Principal" : "*",
    "Action" : "Update:*",
    "Resource" : "*"
  }
  ]
}
```

Die `Allow` Anweisung gewährt Aktualisierungsberechtigungen für alle Ressourcen und die `Deny` Anweisung verweigert Aktualisierungen von EC2 und RDS-DB-Instances. Die `Deny`-Anweisung überschreibt immer Zulassungsaktionen.

Sie können einen Platzhalter mit Ressourcentypen verwenden. Sie können beispielsweise Aktualisierungsberechtigungen für alle EC2 Amazon-Ressourcen — wie Instances, Sicherheitsgruppen und Subnetze — verweigern, indem Sie einen Platzhalter verwenden, wie im folgenden Beispiel gezeigt:

```
"Condition" : {
  "StringLike" : {
    "ResourceType" : ["AWS::EC2::*"]
  }
}
```

Sie müssen die `StringLike`-Bedingung verwenden, wenn Sie Platzhalter verwenden.

Festlegen einer Stack-Richtlinie

Sie können die Konsole verwenden oder eine Stack-Richtlinie anwenden AWS CLI , wenn Sie einen Stack erstellen. Sie können die auch verwenden AWS CLI , um eine Stack-Richtlinie auf einen vorhandenen Stack anzuwenden. Nachdem Sie eine Stack-Richtlinie angewendet haben, können Sie sie nicht mehr aus dem Stack entfernen, aber Sie können sie verwenden, AWS CLI um sie zu ändern.

Stack-Richtlinien gelten für alle CloudFormation Benutzer, die versuchen, den Stack zu aktualisieren. Sie können unterschiedlichen Benutzern keinen verschiedenen Stack-Richtlinien zuordnen.

Informationen zum Schreiben von Stack-Richtlinien finden Sie unter Definieren einer Stack-Richtlinie.

So legen Sie eine Stack-Richtlinie beim Erstellen eines Stacks fest (Konsole)

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm die aus, in der der Stapel erstellt werden AWS-Region soll.
3. Wählen Sie auf der Seite „CloudFormation Stapel“ die Option Stapel erstellen aus.
4. Erweitern Sie im Assistenten für das Erstellen eines Stacks auf der Seite Configure stack options (Stack-Optionen konfigurieren) den Abschnitt Advanced (Erweitert) und wählen Sie dann Stack policy (Stack-Richtlinie) aus.
5. Geben Sie die Stack-Richtlinie an:
   - Um eine Richtlinie direkt in der Konsole zu schreiben, wählen Sie Enter stack policy (Stack-Richtlinie eingeben) aus und geben Sie die Stack-Richtlinie direkt in das Textfeld ein.
   - Um eine Richtlinie zu verwenden, die in einer separaten Datei definiert ist, wählen Sie Upload a file (Datei hochladen) und dann Choose file (Datei auswählen) aus, um die Datei auszuwählen, die die Stack-Richtlinie enthält.

So legen Sie eine Stack-Richtlinie beim Erstellen eines Stacks fest (AWS CLI)

- Verwenden Sie den Befehl create-stack mit der Option `--stack-policy-body`, um eine geänderte Richtlinie einzugeben, oder der Option `--stack-policy-url`, um eine Datei anzugeben, die die Richtlinie enthält.

Um eine Stack-Richtlinie für einen vorhandenen Stack festzulegen (AWS CLI nur)

- Verwenden Sie den Befehl set-stack-policy mit der Option `--stack-policy-body`, um eine geänderte Richtlinie einzugeben, oder der Option `--stack-policy-url`, um eine Datei anzugeben, die die Richtlinie enthält.

> **Note**
>
> Um eine Richtlinie zu einem vorhandenen Stack hinzuzufügen, müssen Sie über die entsprechende Berechtigung für die CloudFormation SetStackPolicyAktion verfügen.

Aktualisieren geschützter Ressourcen

Um geschützte Ressourcen zu aktualisieren, erstellen Sie eine temporäre Richtlinie, die die Stack-Richtlinie überschreibt und Aktualisierungen für diese Ressourcen zulässt. Geben Sie die Überschreibungsrichtlinie ab, wenn Sie den Stack aktualisieren. Die Überschreibungsrichtlinie ändert die Stack-Richtlinie nicht dauerhaft.

Um geschützte Ressourcen zu aktualisieren, benötigen Sie die Berechtigung, die CloudFormation SetStackPolicyAktion zu verwenden. Informationen zum Festlegen von CloudFormation-Berechtigungen finden Sie unter Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management.

> **Note**
>
> Während eines Stack-Updates CloudFormation werden Ressourcen, die von anderen aktualisierten Ressourcen abhängen, automatisch aktualisiert. CloudFormation Aktualisiert beispielsweise eine Ressource, die auf eine aktualisierte Ressource verweist. CloudFormation nimmt keine physischen Änderungen an automatisch aktualisierten Ressourcen vor, z. B. an der ID der Ressourcen. Wenn diesen Ressourcen jedoch eine Stack-Richtlinie zugeordnet ist, benötigen Sie die Berechtigung, sie zu aktualisieren.

So aktualisieren Sie eine geschützte Ressource (Konsole)

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

2. Wählen Sie den Stack aus, den Sie aktualisieren möchten. Wählen Sie Stack actions (Stack-Aktionen) und anschließend Update Stack (Stack aktualisieren) aus.
3. Wenn Sie die Stack-Vorlage nicht geändert haben, wählen Sie Use current template (Aktuelle Vorlage verwenden) aus und klicken Sie dann auf Next (Weiter). Wenn Sie die Vorlage geändert haben, wählen Sie Replace current template (Aktuelle Vorlage ersetzen) aus und geben Sie den Speicherort der aktualisierten Vorlage im Abschnitt Specify template (Vorlage angeben) an:

   - Für eine lokal auf Ihrem Computer gespeicherte Vorlage wählen Sie Upload a template file (Vorlagendatei hochladen) aus. Wählen Sie Choose File, um zur Datei zu navigieren. Wählen Sie die Datei aus und klicken Sie auf Weiter.
   - Für eine in einem Amazon S3-Bucket gespeicherte Vorlage wählen Sie Amazon S3-URL aus. Geben Sie die URL für die Vorlage ein und klicken Sie dann auf Weiter.

     Wenn Sie eine Vorlage in einem Bucket mit aktivierter Versionierung haben, können Sie eine bestimmte Version der Vorlagenvorlage angeben, indem Sie sie an die URL anhängen`?versionId=`*`version-id`*. Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem Bucket mit aktivierter Versionierung im Amazon Simple Storage Service-Benutzerhandbuch.
4. Wenn Ihre Vorlage Parameter enthält, geben Sie auf der Seite Specify stack details (Stack-Details angeben) die Parameterwerte ein oder ändern Sie sie und wählen Sie dann Next (Weiter) aus.

   CloudFormation füllt jeden Parameter mit dem Wert auf, der aktuell im Stack festgelegt ist, mit Ausnahme der mit dem Attribut deklarierten Parameter. `NoEcho` Sie können die aktuellen Werte für diese Parameter verwenden, indem Sie Use existing value (Existierenden Wert verwenden) auswählen.

   Weitere Informationen zur Verwendung `NoEcho` zum Maskieren vertraulicher Informationen sowie zur Verwendung dynamischer Parameter zur Verwaltung von Geheimnissen finden Sie in der Keine Anmeldeinformationen in Vorlagen einbetten bewährten Methode.
5. Geben Sie eine Überschreibungs-Stack-Richtlinie an.

   a. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) im Abschnitt Advanced options (Erweiterte Optionen) die Option Stack policy (Stack-Richtlinie) aus.

   b. Wählen Sie Upload a file (Datei hochladen) aus.

c. Klicken Sie auf Choose File (Datei auswählen) und navigieren Sie zu der Datei, die die überschreibende Stack-Richtlinie enthält, oder geben Sie eine Richtlinie ein.

d. Wählen Sie Weiter aus.

Die Überschreibungsrichtlinie muss eine `Allow`-Anweisung für die geschützten Ressourcen angeben, die Sie aktualisieren möchten. Um beispielsweise alle geschützten Ressourcen zu aktualisieren, geben Sie eine temporäre Überschreibungsrichtlinie an, die alle Aktualisierungen zulässt:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

> **Note**
>
> CloudFormation wendet die Override-Richtlinie nur während dieses Updates an. Die Überschreibungsrichtlinie ändert die Stack-Richtlinie nicht dauerhaft. Informationen zum Ändern einer Stack-Richtlinie finden Sie unter Ändern einer Stack-Richtlinie .

6. Überprüfen Sie die Stack-Informationen und die Änderungen, die Sie übermittelt haben.

   Überprüfen Sie, ob Sie die richtigen Informationen gesendet haben, z. B. die richtigen Parameterwerte oder die richtige Vorlagen-URL. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie I acknowledge that this template may create IAM resources, um anzugeben, dass Sie IAM-Ressourcen in der Vorlage verwenden möchten. Weitere Informationen finden Sie unter Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen.

   Vergewissern Sie sich im Abschnitt Vorschau Ihrer Änderungen, dass alle erwarteten Änderungen vorgenommen CloudFormation werden. Wählen Sie beispielsweise aus, dass die CloudFormation Ressourcen, die Sie hinzufügen, entfernen oder ändern wollten, hinzugefügt, entfernt und geändert werden. CloudFormationgeneriert diese Vorschau, indem

ein Änderungssatz für den Stapel erstellt wird. Weitere Informationen finden Sie unter CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren.

7. Wenn Sie mit Ihren Änderungen zufrieden sind, klicken Sie auf Update (Aktualisieren).

> **Note**
>
> An dieser Stelle haben Sie auch die Möglichkeit, den Änderungssatz anzusehen, um Ihre vorgeschlagenen Updates sorgfältiger zu überprüfen. Klicken Sie dazu nicht auf Aktualisieren, sondern auf Änderungssatz anzeigen. CloudFormation zeigt den Änderungssatz an, der auf der Grundlage Ihrer Aktualisierungen generiert wurde. Wenn Sie bereit sind, den Stack zu aktualisieren, klicken Sie auf Execute (Ausführen).

CloudFormation zeigt die Seite mit den Stack-Details für Ihren Stack an. Ihr Stack hat jetzt den Status`UPDATE_IN_PROGRESS`. Nachdem CloudFormation die Aktualisierung des Stacks erfolgreich abgeschlossen wurde, wird der Stack-Status auf gesetzt`UPDATE_COMPLETE`.

Schlägt die Stack-Aktualisierung fehl CloudFormation, werden die Änderungen automatisch rückgängig gemacht und der Stack-Status wird auf gesetzt`UPDATE_ROLLBACK_COMPLETE`.

So aktualisieren Sie eine geschützte Ressource (AWS CLI)

- Verwenden Sie den Befehl update-stack mit der Option `--stack-policy-during-update-body`, um eine geänderte Richtlinie einzugeben, oder der Option `--stack-policy-during-update-url`, um eine Datei anzugeben, die die Richtlinie enthält.

> **Note**
>
> CloudFormation wendet die Override-Richtlinie nur während dieses Updates an. Die Überschreibungsrichtlinie ändert die Stack-Richtlinie nicht dauerhaft. Informationen zum Ändern einer Stack-Richtlinie finden Sie unter Ändern einer Stack-Richtlinie .

Ändern einer Stack-Richtlinie

Um zusätzliche Ressourcen zu schützen oder Schutz von Ressourcen zu entfernen, ändern Sie die Stack-Richtlinie. Wenn Sie beispielsweise eine Datenbank, die Sie schützen möchten, zu einem

Stack hinzufügen, fügen Sie der Stack-Richtlinie eine `Deny`-Anweisung für diese Datenbank hinzu. Um die Richtlinie zu ändern, benötigen Sie Berechtigungen zum Verwenden der SetStackPolicy-Aktion.

Verwenden Sie die AWS CLI , um Stack-Richtlinien zu ändern.

So ändern Sie eine Stack-Richtlinie (AWS CLI)

- Verwenden Sie den Befehl set-stack-policy mit der Option `--stack-policy-body`, um eine geänderte Richtlinie einzugeben, oder der Option `--stack-policy-url`, um eine Datei anzugeben, die die Richtlinie enthält.

Stack-Richtlinien können nicht gelöscht werden. Zum Entfernen des gesamten Schutzes von allen Ressourcen, ändern Sie die Richtlinie, sodass alle Aktionen für alle Ressourcen ausdrücklich zugelassen werden. Mit der folgenden Richtlinie können alle Aktualisierungen für alle Ressourcen durchgeführt werden:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Weitere Beispiel-Stack-Richtlinien

Die folgenden Beispielrichtlinien zeigen, wie Sie Aktualisierungen aller Stack-Ressourcen und spezifischer Ressourcen verhindern und wie Sie bestimmte Arten von Aktualisierungen verhindern.

Verhindern von Aktualisierungen aller Stack-Ressourcen

Um Aktualisierungen aller Stack-Ressourcen zu verhindern, gibt die folgende Richtlinie eine `Deny`-Anweisung für alle Aktualisierungsaktionen für alle Ressourcen an.

```
{
```

```
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Verhindern von Aktualisierungen an einer einzelnen Ressource

Mit der folgenden Richtlinie werden alle Aktualisierungsaktionen in der Datenbank mit der logischen ID `MyDatabase` verweigert. Es werden alle Aktualisierungsaktionen für alle anderen Stack-Ressourcen mit einer `Allow`-Anweisung zugelassen. Die `Allow`-Anweisung gilt nicht für die `MyDatabase`-Ressource, da die `Deny`-Anweisung immer Zulassungsaktionen überschreibt.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyDatabase"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Sie können dasselbe Ergebnis wie im vorherigen Beispiel erzielen, wenn Sie eine Standardverweigerung verwenden. Wenn Sie eine Stack-Richtlinie festlegen, werden alle Aktualisierungen CloudFormation verweigert, die nicht ausdrücklich zugelassen sind. Die folgende Richtlinie lässt Aktualisierungen für alle Ressourcen zu, außer für die `ProductionDatabase`-Ressource, die standardmäßig verweigert wird.

```
{
```

```
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

> **Important**
>
> Es gibt Risiken bei der Verwendung einer Standardverweigerung. Wenn es an einer anderen Stelle in der Richtlinie eine `Allow`-Anweisung gibt (z. B. eine `Allow`-Anweisung, die einen Platzhalter verwendet), können Sie unwissentlich Aktualisierungsberechtigungen für Ressourcen erteilen, die Sie gar nicht aktualisieren möchten. Da eine ausdrückliche Verweigerung sämtliche Zulassungsaktionen überschreibt, können Sie sicherstellen, dass eine Ressource durch Verwendung einer `Deny`-Anweisung geschützt wird.

Verhindern von Aktualisierungen aller Instances eines Ressourcentyps

Mit der folgenden Richtlinie werden alle Aktualisierungsaktionen für den RDS-DB-Instance-Ressourcentyp verweigert. Es werden alle Aktualisierungsaktionen für alle anderen Stack-Ressourcen mit einer `Allow`-Anweisung zugelassen. Die `Allow`-Anweisung gilt nicht für die RDS-DB-Instance-Ressourcen, da eine `Deny`-Anweisung immer Zulassungsaktionen überschreibt.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::RDS::DBInstance"]
        }
      }
    },
    {
```

```
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Verhindern von Ersetzungsaktualisierungen für eine Instance

Mit der folgenden Richtlinie werden Aktualisierungen verweigert, die zu einer Ersetzung der Instance mit der logischen ID `MyInstance` führen würden. Es werden alle Aktualisierungsaktionen für alle anderen Stack-Ressourcen mit einer `Allow`-Anweisung zugelassen. Die `Allow`-Anweisung gilt nicht für die `MyInstance`-Ressource, da die `Deny`-Anweisung immer Zulassungsaktionen überschreibt.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:Replace",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyInstance"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Verhindern von Aktualisierungen an verschachtelten Stacks

Die folgende Richtlinie verweigert alle Aktualisierungsaktionen für den CloudFormation Stack-Ressourcentyp (verschachtelte Stacks). Es werden alle Aktualisierungsaktionen für alle anderen Stack-Ressourcen mit einer `Allow`-Anweisung zugelassen. Die `Allow`-Anweisung gilt nicht für die CloudFormation -Stack-Ressourcen, da die `Deny`-Anweisung immer Zulassungsaktionen überschreibt.

```
{
```

```
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::CloudFormation::Stack"]
        }
      }
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Datenschutz in AWS CloudFormation

Das Modell der AWS gemeinsamen Verantwortung und geteilter Verantwortung gilt für den Datenschutz in AWS CloudFormation. 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).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.

- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- 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 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

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 AWS CloudFormation API oder auf andere AWS-Services Weise arbeiten oder diese verwenden. AWS CLI AWS SDKs 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.

Verschlüsselung im Ruhezustand

AWS CloudFormation Speichert Ihre Daten gemäß dem Modell der AWS gemeinsamen Verantwortung verschlüsselt im Ruhezustand. Kunden sind dafür verantwortlich, Verschlüsselungs- und Speicherrichtlinien für in ihren Konten gespeicherte Daten festzulegen. Wir empfehlen beispielsweise, die Verschlüsselung im Ruhezustand für Vorlagen und andere Daten zu aktivieren, die in S3-Buckets oder SNS-Themen gespeichert sind. Kunden definieren ebenso Verschlüsselungseinstellungen für alle Datenspeichersysteme, die von CloudFormation bereitgestellt werden.

Verschlüsselung während der Übertragung

AWS CloudFormation verwendet verschlüsselte Kanäle für die Dienstkommunikation im Rahmen des Modells der gemeinsamen Verantwortung.

Richtlinie für den Datenverkehr zwischen Netzwerken

AWS CloudFormation Die Dienstkommunikation zwischen Regionen oder Availability Zones wird standardmäßig sicher verschlüsselt.

Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management

Mit AWS Identity and Access Management (IAM) können Sie IAM-Benutzer erstellen und deren Zugriff auf bestimmte Ressourcen in Ihrem System kontrollieren. AWS-Konto Wenn Sie IAM verwenden, können Sie steuern, was Benutzer verwenden können CloudFormation, z. B. ob sie Stack-Vorlagen anzeigen, Stacks erstellen oder Stapel löschen können.

Neben CloudFormation spezifischen Aktionen können Sie verwalten, welche AWS Dienste und Ressourcen jedem Benutzer zur Verfügung stehen. Auf diese Weise können Sie steuern, auf welche Ressourcen Benutzer zugreifen können, wenn sie sie verwenden CloudFormation. Sie können beispielsweise angeben, welche Benutzer EC2 Amazon-Instances erstellen, Datenbank-Instances beenden oder Updates durchführen können VPCs. Dieselben Berechtigungen gelten immer dann CloudFormation, wenn sie diese Aktionen ausführen.

Verwenden Sie die Informationen in den folgenden Abschnitten, um zu steuern, wer darauf zugreifen kann CloudFormation. Wir werden auch untersuchen, wie Sie die Erstellung von IAM-Ressourcen in Vorlagen autorisieren, Anwendungen, die auf EC2 Instances ausgeführt werden, die erforderlichen Berechtigungen erteilen und temporäre Sicherheitsanmeldedaten verwenden, um die Sicherheit in Ihrer AWS Umgebung zu erhöhen.

Definition identitätsbasierter IAM-Richtlinien für CloudFormation

Um Zugriff darauf zu gewähren CloudFormation, müssen Sie IAM-Richtlinien erstellen und zuweisen, die Ihren IAM-Identitäten (wie Benutzern oder Rollen) die Erlaubnis geben, die benötigten API-Aktionen aufzurufen.

Mit identitätsbasierten IAM-Richtlinien können Sie zulässige oder verweigerte Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zulässig oder verweigert werden. CloudFormationunterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel.

Wenn Sie mit IAM noch nicht vertraut sind, sollten Sie sich zunächst mit den Elementen einer IAM-JSON-Richtlinie vertraut machen. Weitere Informationen finden Sie unter Referenz zu den

IAM-JSON-Richtlinienelementen im IAM-Benutzerhandbuch. Um zu erfahren, wie Sie IAM-Richtlinien erstellen können, lesen Sie das Tutorial Erstellen und Anhängen Ihrer ersten vom Kunden verwalteten Richtlinie in der IAM-Dokumentation durch.

Themen

- Richtlinienaktionen für CloudFormation
- Konsolenspezifische Aktionen für CloudFormation
- Richtlinienressourcen für CloudFormation
- Schlüssel für die Richtlinienbedingung für CloudFormation

Richtlinienaktionen für CloudFormation

Im `Action` Element Ihrer IAM-Richtlinienerklärung können Sie jede API-Aktion angeben, die CloudFormation angeboten wird. Sie müssen dem Aktionsnamen die Zeichenfolge in Kleinbuchstaben voranstellen. `cloudformation:` Beispiel: `cloudformation:CreateStack`, `cloudformation:CreateChangeSet` und `cloudformation:UpdateStack`.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommas:

```
"Action": [ "cloudformation:action1", "cloudformation:action2" ]
```

Sie können auch mehrere Aktionen mittels Platzhaltern angeben. Sie können beispielsweise alle Aktionen, deren Namen mit dem Wort beginnen`Get`, wie folgt angeben:

```
"Action": "cloudformation:Get*"
```

Eine vollständige Liste der mit dem `cloudformation` Dienstpräfix verknüpften Aktionen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS CloudFormation und Aktionen, Ressourcen und Bedingungsschlüssel für AWS -Cloud-Control- API in der Serviceautorisierungsreferenz.

Beispiele

Im Folgenden finden Sie ein Beispiel für eine Berechtigungsrichtlinie, die Berechtigungen zum Anzeigen von CloudFormation Stacks gewährt.

Example 1: Eine Beispielrichtlinie, die View-Stack-Berechtigungen gewährt

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Allow",
        "Action":[
            "cloudformation:DescribeStacks",
            "cloudformation:DescribeStackEvents",
            "cloudformation:DescribeStackResource",
            "cloudformation:DescribeStackResources"
        ],
        "Resource":"*"
    }]
}
```

Benutzer, die Stacks erstellen oder löschen, benötigen zusätzliche Berechtigungen, die auf ihren Stack-Vorlagen basieren. Wenn Ihre Vorlage beispielsweise eine Amazon SQS SQS-Warteschlange beschreibt, müssen Benutzer über Berechtigungen sowohl für Amazon SQS-Aktionen als auch CloudFormation für Amazon SQS SQS-Aktionen verfügen, wie in der folgenden Beispielrichtlinie dargestellt.

Example 2: Eine Beispielrichtlinie, die Stack-Aktionen zum Erstellen und Anzeigen sowie alle Amazon SQS SQS-Aktionen gewährt

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Allow",
        "Action":[
            "sqs:*",
            "cloudformation:CreateStack",
            "cloudformation:DescribeStacks",
            "cloudformation:DescribeStackEvents",
            "cloudformation:DescribeStackResources",
            "cloudformation:GetTemplate",
```

```
            "cloudformation:ValidateTemplate"
        ],
        "Resource":"*"
    }]
}
```

Konsolenspezifische Aktionen für CloudFormation

Benutzer der CloudFormation Konsole benötigen zusätzliche Berechtigungen, die über die Berechtigungen für oder hinausgehen. AWS Command Line Interface CloudFormation APIs Diese zusätzlichen Berechtigungen unterstützen konsolenspezifische Funktionen wie Vorlagen-Uploads in Amazon S3 S3-Buckets und Drop-down-Listen für AWS-spezifische Parametertypen.

Erteilen Sie für alle unten aufgeführten Aktionen Berechtigungen für alle Ressourcen. Beschränken Sie sie nicht auf bestimmte Stacks oder Buckets.

Die folgende Aktion wird nur von der CloudFormation Konsole verwendet und ist in der API-Referenz nicht dokumentiert. Die Aktion erlaubt Benutzern, Vorlagen zu Amazon S3-Buckets hochzuladen.

- `cloudformation:CreateUploadBucket`

Wenn Benutzer Vorlagen hochladen, benötigen Benutzer außerdem die folgenden Amazon S3 S3-Berechtigungen:

- `s3:PutObject`
- `s3:ListBucket`
- `s3:GetObject`
- `s3:CreateBucket`

Um Werte in den Parameter-Dropdownlisten für Vorlagen mit AWS-spezifischen Parametertypen zu sehen, benötigen Benutzer Berechtigungen, um die entsprechenden Describe-API-Aufrufe durchzuführen. Beispielsweise sind die folgenden Berechtigungen erforderlich, wenn diese Parametertypen in der Vorlage verwendet werden:

- `ec2:DescribeKeyPairs`— Für den `AWS::EC2::KeyPair::KeyName` Parametertyp erforderlich.

- `ec2:DescribeSecurityGroups`— Erforderlich für den `AWS::EC2::SecurityGroup::Id` Parametertyp.
- `ec2:DescribeSubnets`— Erforderlich für den `AWS::EC2::Subnet::Id` Parametertyp.
- `ec2:DescribeVpcs`— Erforderlich für den `AWS::EC2::VPC::Id` Parametertyp.

Weitere Hinweise zu AWS-spezifischen Parametertypen finden Sie unterGeben Sie vorhandene Ressourcen zur Laufzeit mit von CloudFormation -bereitgestellten Parametertypen an.

Richtlinienressourcen für CloudFormation

In einer IAM-Richtlinienanweisung gibt das `Resource`-Element das Objekt oder die Objekte an, für die die Anweisung gilt. Denn CloudFormation jede IAM-Richtlinienerklärung gilt für die Ressourcen, die Sie unter Verwendung ihrer Amazon-Ressourcennamen (ARNs) angeben. Das spezifische ARN-Format hängt von der Ressource ab.

Eine vollständige Liste der CloudFormation Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Ressourcentypen definiert von AWS CloudFormation in der Service Authorization Reference. Informationen zu den Aktionen, die Sie mit dem ARN jeder Ressource angeben können, finden Sie unter Aktionen definiert von AWS CloudFormation.

Sie können Aktionen für einen bestimmten Stack angeben, wie im folgenden Richtlinienbeispiel gezeigt. Wenn Sie einen ARN angeben, ersetzen Sie ihn durch Ihre ressourcenspezifischen Informationen. *`placeholder text`*

Example 1: Eine Beispielrichtlinie, die die Aktionen zum Löschen und Aktualisieren des Stacks für den angegebenen Stack verweigert

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "cloudformation:DeleteStack",
                "cloudformation:UpdateStack"
            ],
```

```
            "Resource": "arn:aws:cloudformation:us-
east-1:111122223333:stack/MyProductionStack/*"
        }
    ]
}
```

Die obige Richtlinie verwendet einen Platzhalter am Ende des Stack-Namens, sodass das Löschen des Stacks und das Aktualisieren des Stacks sowohl für die vollständige Stack-ID (wie`arn:aws:cloudformation:region:account-id:stack/MyProductionStack/abc9dbf0-43c2-11e3-a6e8-50fa526be49c`) als auch für den Stacknamen (wie`MyProductionStack`) verweigert werden.

Damit `AWS::Serverless` Transformationen einen Änderungssatz erstellen können, fügen Sie die Berechtigung `arn:aws:cloudformation:region:aws:transform/Serverless-2016-10-31` auf Ressourcenebene hinzu, wie in der folgenden Richtlinie dargestellt.

Example 2: Eine Beispielrichtlinie, die die Aktion „Änderungssatz erstellen“ für die angegebene Transformation ermöglicht

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateChangeSet"
            ],
            "Resource": "arn:aws:cloudformation:us-east-1:aws:transform/
Serverless-2016-10-31"
        }
    ]
}
```

Schlüssel für die Richtlinienbedingung für CloudFormation

In einer IAM-Richtlinienerklärung können Sie optional Bedingungen angeben, die steuern, wann eine Richtlinie in Kraft tritt. Sie können beispielsweise eine Richtlinie definieren, die -Benutzern

erlaubt, einen Stack nur dann zu erstellen, wenn sie eine bestimmte URL-Vorlage angeben. Sie können CloudFormation spezifische Bedingungen und AWS allgemeine Bedingungen definieren, z. B. `DateLessThan` festlegen, wann eine Richtlinie nicht mehr wirksam wird. Weitere Informationen und eine Liste der AWS allgemeinen Bedingungen finden Sie unter Bedingung in der Referenz zu den IAM-Richtlinienelementen im IAM-Benutzerhandbuch.

> **Note**
>
> Verwenden Sie nicht die Bedingung -wide. `aws:SourceIp` AWS CloudFormationstellt Ressourcen mithilfe der eigenen IP-Adresse bereit, nicht der IP-Adresse der ursprünglichen Anfrage. Wenn Sie beispielsweise einen Stack erstellen, CloudFormation stellt er Anfragen von seiner IP-Adresse aus, um eine EC2 Amazon-Instance zu starten oder einen Amazon S3-Bucket zu erstellen, und nicht von der IP-Adresse aus dem `CreateStack` Aufruf oder dem create-stack Befehl.

In der folgenden Liste werden die CloudFormation -spezifischen Bedingungen beschrieben. Diese Bedingungen werden nur angewendet, wenn Benutzer Stacks erstellen oder aktualisieren:

`cloudformation:ChangeSetName`

Ein Name des CloudFormation Änderungssatzes, den Sie einer Richtlinie zuordnen möchten. Verwenden Sie diese Bedingung, um zu steuern, welche Änderungssätze -Benutzer ausführen oder löschen können.

`cloudformation:ImportResourceTypes`

Die Vorlagenressourcentypen, die Sie einer Richtlinie zuordnen möchten, z. B. `AWS::EC2::Instance`. Verwenden Sie diese Bedingung, um zu steuern, mit welchen Ressourcentypen -Benutzer arbeiten können, wenn sie Ressourcen in einen Stack importieren. Diese Bedingung wird anhand der Ressourcentypen geprüft, die Benutzer im `ResourcesToImport` Parameter deklarieren. Dieser Parameter wird derzeit nur für AWS CLI und API-Anfragen unterstützt. Wenn Sie diesen Parameter verwenden, müssen Sie alle Ressourcentypen angeben, die Benutzer während des Importvorgangs steuern sollen. Weitere Informationen zu dem `ResourcesToImport` Parameter finden Sie unter der CreateChangeSetAktion in der AWS CloudFormation API-Referenz.

Eine Liste der möglichen `ResourcesToImport` Optionen finden Sie unterUnterstützung für Ressourcentypen.

Verwenden Sie die dreiteilige Ressourcennamenskonvention, um anzugeben, mit welchen Ressourcentypen Benutzer arbeiten können, von allen Ressourcen innerhalb einer Organisation bis hinunter zu einem einzelnen Ressourcentyp.

`organization::*`

Geben Sie alle Ressourcentypen für eine bestimmte Organisation an.

`organization::service_name::*`

Geben Sie alle Ressourcentypen für den angegebenen Service innerhalb einer bestimmten Organisation an.

`organization::service_name::resource_type`

Geben Sie einen bestimmten Ressourcentyp an.

Zum Beispiel:

`AWS::*`

Geben Sie alle unterstützten AWS Ressourcentypen an.

`AWS::service_name::*`

Geben Sie alle unterstützten Ressourcen für einen bestimmten AWS -Service an.

`AWS::service_name::resource_type`

Geben Sie einen bestimmten AWS Ressourcentyp an, z. B. `AWS::EC2::Instance` (alle EC2 Instanzen).

`cloudformation:ResourceTypes`

Die Vorlagenressourcentypen wie z. B. `AWS::EC2::Instance`, die Sie einer Richtlinie zuordnen möchten. Verwenden Sie diese Bedingung, um zu steuern, welche Ressourcentypen -Benutzern zur Verfügung stehen, wenn sie einen Stack erstellen oder aktualisieren. Diese Bedingung wird anhand der Ressourcentypen geprüft, die Benutzer im `ResourceTypes`-Parameter deklarieren, der aktuell nur für AWS CLI und API-Anforderungen unterstützt wird. Bei Verwendung dieses Parameters müssen Benutzer alle Ressourcentypen angeben, die sich in ihrer Vorlage befinden. Weitere Informationen zu dem `ResourceTypes` Parameter finden Sie unter der CreateStackAktion in der AWS CloudFormation API-Referenz.

Eine Liste der Ressourcentypen finden Sie im Referenzhandbuch für AWS CloudFormation Vorlagen.

Verwenden Sie die dreiteilige Ressourcennamenskonvention, um anzugeben, mit welchen Ressourcentypen Benutzer arbeiten können, von allen Ressourcen innerhalb einer Organisation bis hinunter zu einem einzelnen Ressourcentyp.

`organization::*`

Geben Sie alle Ressourcentypen für eine bestimmte Organisation an.

`organization::service_name::*`

Geben Sie alle Ressourcentypen für den angegebenen Service innerhalb einer bestimmten Organisation an.

`organization::service_name::resource_type`

Geben Sie einen bestimmten Ressourcentyp an.

Zum Beispiel:

`AWS::*`

Geben Sie alle unterstützten AWS Ressourcentypen an.

`AWS::service_name::*`

Geben Sie alle unterstützten Ressourcen für einen bestimmten AWS -Service an.

`AWS::service_name::resource_type`

Geben Sie einen bestimmten AWS Ressourcentyp an, z. B. `AWS::EC2::Instance` (alle EC2 Instanzen).

`Alexa::ASK::*`

Geben Sie alle Ressourcentypen im Alexa Skill Kit an.

`Alexa::ASK::Skill`

Geben Sie den individuellen Alexa: :ASK: :Skill-Ressourcentyp an.

`Custom::*`

Geben Sie alle benutzerdefinierten Ressourcen an.

Weitere Informationen finden Sie unter Erstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen.

`Custom::resource_type`

Geben Sie einen bestimmten benutzerdefinierten Ressourcentyp an.

Weitere Informationen finden Sie unter Erstellen Sie eine benutzerdefinierte Bereitstellungslogik mit benutzerdefinierten Ressourcen.

`cloudformation:RoleARN`

Der Amazon-Ressourcenname (ARN) einer IAM-Servicerolle, die Sie einer Richtlinie zuordnen möchten. Verwenden Sie diese Bedingung, um zu steuern, welche Servicerolle -Benutzer verwenden können, wenn sie mit Stacks oder Änderungssätzen arbeiten.

`cloudformation:StackPolicyUrl`

Eine Amazon S3-Stack-Richtlinien-URL, die Sie einer Richtlinie zuordnen möchten. Verwenden Sie diese Bedingung, um zu steuern, welche Stack-Richtlinien -Benutzer einem Stack während einer Aktion zum Erstellen oder Aktualisieren von Stacks zuordnen können. Weitere Informationen zu Stack-Richtlinien finden Sie unter Verhindern von Aktualisierungen der Stack-Ressourcen.

> **Note**
>
> Um sicherzustellen, dass Benutzer nur Stacks mit den von Ihnen hochgeladenen Stack-Richtlinien erstellen oder aktualisieren können, stellen Sie den S3-Bucket so ein, dass er nur für diese Benutzer lesbar ist.

`cloudformation:TemplateUrl`

Eine Amazon S3-Vorlagen-URL, die Sie einer Richtlinie zuordnen möchten. Verwenden Sie diese Bedingung, um zu steuern, welche Vorlagen -Benutzer verwenden können, wenn sie Stacks erstellen oder aktualisieren.

> **Note**
>
> Um sicherzustellen, dass Benutzer nur Stacks mit den von Ihnen hochgeladenen Vorlagen erstellen oder aktualisieren können, stellen Sie den S3-Bucket so ein, dass er nur für diese Benutzer lesbar ist.

> **Note**
>
> Die folgenden CloudFormation spezifischen Bedingungen gelten für die gleichnamigen API-Parameter:

- `cloudformation:ChangeSetName`
- `cloudformation:RoleARN`
- `cloudformation:StackPolicyUrl`
- `cloudformation:TemplateUrl`

  Gilt beispielsweise `cloudformation:TemplateUrl` nur für den `TemplateUrl` Parameter für `CreateStackUpdateStack`, und `CreateChangeSet` APIs.

Beispiele für IAM-Richtlinien, die Bedingungsschlüssel zur Zugriffskontrolle verwenden, finden Sie unterBeispiel für identitätsbasierte IAM-Richtlinien für CloudFormation.

Bestätigen von IAM-Ressourcen in CloudFormation-Vorlagen

Bevor Sie einen Stack erstellen können, wird Ihre CloudFormation Vorlage validiert. Bei der Validierung prüft CloudFormation Ihre Vorlage auf IAM-Ressourcen, die es möglicherweise erstellt. IAM-Ressourcen, wie z. B. ein Benutzer mit Vollzugriff, können auf jede Ressource in Ihrem AWS-Konto zugreifen und diese ändern. Daher wird empfohlen, die den einzelnen IAM-Ressourcen zugeordneten Berechtigungen zu prüfen, bevor Sie fortfahren, damit Sie nicht versehentlich Ressourcen mit eskalierten Berechtigungen erstellen. Um sicherzustellen, dass Sie dies getan haben, müssen Sie bestätigen, dass die Vorlage diese Ressourcen enthält, und CloudFormation die angegebenen Funktionen bereitstellen, bevor der Stack erstellt wird.

Sie können die Funktionen von CloudFormation Vorlagen mithilfe der CloudFormation Konsole, AWS Command Line Interface (AWS CLI) oder der API überprüfen:

- Wählen Sie in der CloudFormation Konsole auf der Seite „Stack-Optionen konfigurieren“ der Assistenten „Stack erstellen“ oder „Stack aktualisieren“ die Option Ich bestätige, dass diese Vorlage IAM-Ressourcen erstellen kann.
- Geben Sie im AWS CLI Feld, wenn Sie die update-stackBefehle create-stackund verwenden, den `CAPABILITY_NAMED_IAM` Wert `CAPABILITY_IAM` oder für die `--capabilities` Option an. Wenn Ihre Vorlage IAM-Ressourcen enthält, können Sie jede der beiden Fähigkeiten angeben. Wenn Ihre Vorlage benutzerdefinierte Namen für IAM-Ressourcen enthält, müssen Sie `CAPABILITY_NAMED_IAM` angeben.
- Wenn Sie in der API die UpdateStackAktionen CreateStackund verwenden, geben Sie `Capabilities.member.1=CAPABILITY_IAM` oder an`Capabilities.member.1=CAPABILITY_NAMED_IAM`. Wenn Ihre Vorlage IAM-Ressourcen

enthält, können Sie jede der beiden Fähigkeiten angeben. Wenn Ihre Vorlage benutzerdefinierte Namen für IAM-Ressourcen enthält, müssen Sie `CAPABILITY_NAMED_IAM` angeben.

> **Important**
>
> Wenn Ihre Vorlage benutzerdefinierte benannte IAM-Ressourcen enthält, erstellen Sie unter Wiederverwendung der gleichen Vorlage nicht mehrere Stacks. IAM-Ressourcen müssen innerhalb Ihres Kontos global eindeutig sein. Wenn Sie dieselbe Vorlage verwenden, um mehrere Stacks in verschiedenen Regionen zu erstellen, teilen sich Ihre Stacks möglicherweise dieselben IAM-Ressourcen, anstatt dass jeder eine eindeutige hat. Die gemeinsame Verwendung von Ressourcen durch Stacks kann unbeabsichtigte Folgen haben, die nicht rückgängig gemacht werden können. Wenn Sie beispielsweise gemeinsam verwendete IAM-Ressourcen löschen oder aktualisieren, ändern Sie versehentlich die Ressourcen anderer Stacks.

Verwaltung von Anmeldeinformationen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden

Wenn Sie eine Anwendung haben, die auf einer EC2 Amazon-Instance läuft und Anfragen an AWS Ressourcen wie Amazon S3-Buckets oder eine DynamoDB-Tabelle stellen muss, benötigt AWS die Anwendung Sicherheitsanmeldedaten. Allerdings stellt die Verteilung und Einbettung langfristiger Sicherheitsanmeldeinformationen in jede Instance, die Sie starten, eine Herausforderung und ein potenzielles Sicherheitsrisiko dar. Anstatt langfristige Anmeldeinformationen wie IAM-Benutzeranmeldedaten zu verwenden, empfehlen wir, dass Sie eine IAM-Rolle erstellen, die einer EC2 Amazon-Instance zugeordnet ist, wenn die Instance gestartet wird. Eine Anwendung kann dann temporäre Sicherheitsanmeldedaten von der EC2 Amazon-Instance abrufen. Sie müssen auf der Instance keine langfristigen Anmeldeinformationen einbetten. Um die Verwaltung von Anmeldeinformationen zu vereinfachen, können Sie außerdem nur eine einzige Rolle für mehrere EC2 Amazon-Instances angeben. Sie müssen nicht für jede Instance eindeutige Anmeldeinformationen erstellen.

Einen Vorlagenausschnitt, der zeigt, wie eine Instance mit einer Rolle gestartet wird, finden Sie unter Vorlagenbeispiele für IAM-Rollen.

> **Note**
>
> Anwendungen auf Instances, die temporäre Sicherheitsanmeldeinformationen verwenden, können jede beliebige CloudFormation -Aktion aufrufen. Da es jedoch mit vielen anderen AWS Diensten CloudFormation interagiert, müssen Sie sicherstellen, dass alle Dienste, die Sie verwenden möchten, temporäre Sicherheitsanmeldedaten unterstützen. Eine Liste der Dienste, die temporäre Sicherheitsanmeldedaten akzeptieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Temporären Zugriff gewähren (Verbundzugriff)

In einigen Fällen möchten Sie möglicherweise Benutzern ohne AWS Anmeldeinformationen temporären Zugriff auf Ihre AWS-Konto gewähren. Verwenden Sie AWS Security Token Service (AWS STS), anstatt langfristige Anmeldeinformationen zu erstellen und zu löschen, wann immer Sie temporären Zugriff gewähren möchten. Sie können beispielsweise IAM-Rollen verwenden. Von einer IAM-Rolle aus können Sie programmgesteuert viele temporäre Sicherheitsanmeldeinformationen (einschließlich eines Zugriffsschlüssels, eines geheimen Zugriffsschlüssels und eines Sicherheits-Tokens) erstellen und dann verteilen. Diese Anmeldeinformationen haben eine begrenzte Lebensdauer, sodass sie AWS-Konto nach Ablauf nicht mehr für den Zugriff auf Ihre Anmeldeinformationen verwendet werden können. Sie können auch mehrere IAM-Rollen erstellen, um einzelnen Benutzern verschiedene Ebenen von Berechtigungen zu gewähren. IAM-Rollen sind nützlich für Szenarien wie verbundene Identitäten und Single Sign-On.

Ein Identitätsverbund ist eine unabhängige Identität, die Sie für mehrere Systeme verwenden können. Bei Enterprise-Benutzern mit einem festgelegten On-Premises-Identitätssystem (wie LDAP oder Active Directory) kann die gesamte Authentifizierung mit Ihrem lokalen Identitätssystem erfolgen. Nachdem ein Benutzer authentifiziert wurde, stellen Sie aus dem entsprechenden IAM-Benutzer oder der entsprechenden IAM-Rolle temporäre Sicherheitsanmeldeinformationen bereit. Sie können beispielsweise eine Administratorrolle und eine Entwicklerrolle erstellen, in denen Administratoren vollen Zugriff auf das AWS Konto haben und Entwickler nur berechtigt sind, mit CloudFormation Stacks zu arbeiten. Nachdem ein Administrator authentifiziert wurde, ist der Administrator berechtigt, temporäre Sicherheitsanmeldeinformationen von der Administratorrolle abzurufen. Entwickler können temporäre Sicherheitsanmeldedaten jedoch nur von der Entwicklerrolle beziehen.

Sie können Verbundbenutzern auch Zugriff auf die AWS Management Console gewähren. Nachdem sich Benutzer bei Ihrem On-Premises-Identitätssystem authentifiziert haben, können Sie programmgesteuert eine temporäre URL erstellen, die direkten Zugriff auf die AWS Management

Console bietet. Wenn Benutzer die temporäre URL verwenden, müssen sie sich nicht anmelden, AWS da sie bereits authentifiziert wurden (Single Sign-On). Da die URL außerdem aus den temporären Sicherheitsanmeldeinformationen des Benutzers erstellt wurde, bestimmen die mit diesen Anmeldeinformationen verfügbaren Berechtigungen, welche Berechtigungen Benutzer in der AWS Management Console besitzen.

Sie können mehrere verschiedene verwenden, um temporäre Sicherheitsanmeldeinformationen AWS STS APIs zu generieren. Weitere Informationen darüber, welche API verwendet werden sollte, finden Sie unter AWS STS Anmeldeinformationen vergleichen im IAM-Benutzerhandbuch.

> **Important**
>
> Sie können IAM nicht verwenden, wenn Sie temporäre Sicherheitsanmeldeinformationen verwenden, die über die `GetFederationToken`-API generiert wurden. Wenn Sie IAM verwenden müssen, verwenden Sie stattdessen temporäre Sicherheitsanmeldeinformationen aus einer Rolle.

CloudFormation interagiert mit vielen anderen AWS Diensten. Wenn Sie temporäre Sicherheitsanmeldedaten mit verwenden CloudFormation, stellen Sie sicher, dass alle Dienste, die Sie verwenden möchten, temporäre Sicherheitsanmeldedaten unterstützen. Eine Liste der Dienste, die temporäre Sicherheitsanmeldedaten akzeptieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Weitere Informationen finden Sie in den folgenden verwandten Ressourcen im IAM-Benutzerhandbuch:

- Allgemeine Szenarien für temporäre Anmeldeinformationen
- Aktivieren Sie den benutzerdefinierten Identity Broker-Zugriff auf die AWS Konsole

Beispiel für identitätsbasierte IAM-Richtlinien für CloudFormation

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, CloudFormation-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder API ausführen. AWS 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. Weitere Informationen finden Sie unter Definition identitätsbasierter IAM-Richtlinien für CloudFormation.

Die folgenden Beispiele zeigen Richtlinienanweisungen, mit denen Sie Berechtigungen für eine oder mehrere CloudFormation Aktionen zulassen oder verweigern können.

Themen

- Erfordern Sie eine bestimmte Vorlagen-URL
- Alle CloudFormation Importvorgänge ablehnen
- Erlauben Sie Importvorgänge für bestimmte Ressourcentypen
- IAM-Ressourcen in Stack-Vorlagen verweigern
- Erlaubt die Erstellung von Stacks mit bestimmten Ressourcentypen
- Steuern Sie den Zugriff auf der Grundlage von API-Aktionen, die Ressourcen verändern
- Beschränken Sie Stack-Set-Operationen auf der Grundlage von Region und Ressourcentypen
- Alle IaC-Generatoroperationen zulassen

Erfordern Sie eine bestimmte Vorlagen-URL

Die folgende Richtlinie gewährt Berechtigungen, nur die *https://s3.amazonaws.com/amzn-s3-demo-bucket/test.template* Vorlagen-URL zum Erstellen oder Aktualisieren eines Stacks zu verwenden.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:UpdateStack"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudformation:TemplateUrl": [
```

```
                        "https://s3.amazonaws.com/amzn-s3-demo-bucket/
test.template"
                    ]
                }
            }
        }
    ]
}
```

Alle CloudFormation Importvorgänge ablehnen

Die folgende Richtlinie gewährt Berechtigungen zum Abschließen aller CloudFormation Vorgänge mit Ausnahme von Importvorgängen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAllStackOperations",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*"
    },
    {
      "Sid": "DenyImport",
      "Effect": "Deny",
      "Action": "cloudformation:*",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "cloudformation:ImportResourceTypes": [
            "*"
          ]
        }
      }
    }
  ]
}
```

Erlauben Sie Importvorgänge für bestimmte Ressourcentypen

Die folgende Richtlinie gewährt Berechtigungen für alle Stack-Operationen, zusätzlich zu Importvorgängen nur für bestimmte Ressourcen (in diesem Beispiel`AWS::S3::Bucket`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowImport",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
          "cloudformation:ImportResourceTypes": [
            "AWS::S3::Bucket"
          ]
        }
      }
    }
  ]
}
```

IAM-Ressourcen in Stack-Vorlagen verweigern

Die folgende Richtlinie gewährt Berechtigungen zum Erstellen von Stacks, lehnt jedoch Anfragen ab, wenn die Vorlage des Stacks Ressourcen aus dem IAM-Dienst enthält. Die Richtlinie erfordert außerdem, dass Benutzer den `ResourceTypes`-Parameter angeben, der nur für AWS CLI -API-Anforderungen verfügbar ist. Diese Richtlinie verwendet explizite Zugriffsverweigerungsanweisungen, sodass diese Richtlinie immer wirksam ist, wenn eine andere Richtlinie zusätzliche Berechtigungen gewährt (eine explizite Zugriffsverweigerungsanweisung setzt eine explizite Zugriffserlaubnisanweisung immer außer Kraft).

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect" : "Allow",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",
      "Condition" : {
        "ForAnyValue:StringLikeIfExists" : {
          "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
        }
      }
    },
    {
      "Effect": "Deny",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "cloudformation:ResourceTypes": "true"
        }
      }
    }
  ]
}
```

Erlaubt die Erstellung von Stacks mit bestimmten Ressourcentypen

Die folgende Richtlinie ähnelt dem vorherigen Beispiel. Die Richtlinie gewährt Berechtigungen zum Erstellen eines Stacks, sofern die Vorlage des Stacks keine Ressource aus dem IAM-Dienst enthält. Außerdem müssen Benutzer den `ResourceTypes` Parameter angeben, der nur für AWS CLI API-Anfragen verfügbar ist. Diese Richtlinie ist einfacher, verwendet aber keine expliziten Zugriffsverweigerungsanweisungen. Andere Richtlinien, die zusätzliche Berechtigungen gewähren, könnten diese Richtlinie außer Kraft setzen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect" : "Allow",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",
      "Condition" : {
        "ForAllValues:StringNotLikeIfExists" : {
          "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
        },
        "Null":{
          "cloudformation:ResourceTypes": "false"
        }
      }
    }
  ]
}
```

Steuern Sie den Zugriff auf der Grundlage von API-Aktionen, die Ressourcen verändern

Die folgende Richtlinie gewährt Berechtigungen zum Filtern des Zugriffs nach dem Namen einer API-Aktion, die eine Ressource verändert. Damit wird gesteuert, mit welchen APIs IAM-Benutzern Tags zu einem Stack oder Stack-Set hinzugefügt oder entfernt werden können. Der Vorgang, der zum Hinzufügen oder Entfernen von Tags verwendet wird, sollte als Wert für den Bedingungsschlüssel hinzugefügt werden. Die folgenden Richtlinien gewähren `TagResource` und `UntagResource` berechtigen zur Durchführung von Mutationsoperationen`CreateStack`.

JSON

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

```
            "cloudformation:TagResource",
            "cloudformation:UntagResource"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "cloudformation:CreateAction": [
                    "CreateStack"
                ]
            }
        }
    }]
}
```

Beschränken Sie Stack-Set-Operationen auf der Grundlage von Region und Ressourcentypen

Die folgende Richtlinie gewährt vom Service verwaltete Stackset-Berechtigungen. Ein Benutzer mit dieser Richtlinie kann nur Vorgänge an Stack-Sets mit Vorlagen durchführen, die Amazon-S3-Ressourcentypen (`AWS::S3::*`) oder den `AWS::SES::ConfigurationSet`-Ressourcentyp enthalten. Wenn der Benutzer mit der ID beim Organisationsverwaltungskonto angemeldet ist`123456789012`, kann er auch nur Operationen an Stack-Sets ausführen, die auf die Organisationseinheit mit der ID abzielen*`ou-1fsfsrsdsfrewr`*, und nur Operationen an dem Stack-Set mit der ID ausführen`stack-set-id`, das auf die AWS-Konto mit der ID *`987654321012`* abzielt.

Stack-Set-Operationen schlagen fehl, wenn die Stackset-Vorlage andere als die in der Richtlinie angegebenen Ressourcentypen enthält, oder wenn es sich bei den Bereitstellungszielen um OU oder Konten IDs handelt, die nicht in der Richtlinie für die entsprechenden Verwaltungskonten und Stacksets angegeben sind.

Diese Richtlinien-Einschränkungen gelten nur, wenn Stack-Set-Vorgänge auf die AWS-Regionen`us-east-1`, `us-west-2` oder `eu-west-2` abzielen.

JSON

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

```
            "Effect": "Allow",
            "Action": [
                "cloudformation:*"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stackset/*",
                "arn:aws:cloudformation:*:*:type/resource/AWS-S3-*",
                "arn:aws:cloudformation:us-west-2::type/resource/AWS-SES-
ConfigurationSet",
                "arn:aws:cloudformation:*:111122223333:stackset-target/
*/ou-1fsfsrsdsfrewr",
                "arn:aws:cloudformation:*:111122223333:stackset-target/stack-set-
id/444455556666"
            ],
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudformation:TargetRegion": [
                        "us-east-1",
                        "us-west-2",
                        "eu-west-1"
                    ]
                }
            }
        }
    ]
}
```

Alle IaC-Generatoroperationen zulassen

Die folgende Richtlinie ermöglicht den Zugriff auf CloudFormation Aktionen im Zusammenhang mit dem Scannen von IaC-Generator-Ressourcen und der Vorlagenverwaltung. Die erste Anweisung gewährt Berechtigungen zum Beschreiben, Auflisten und Starten von Ressourcenscans. Sie ermöglicht auch den Zugriff auf zusätzliche erforderliche Berechtigungen (`cloudformation:GetResource`, und`cloudformation:ListTypes`)`cloudformation:ListResources`, die es dem IaC-Generator ermöglichen, Informationen über Ressourcen und verfügbare Ressourcentypen abzurufen. Die zweite Anweisung gewährt volle Berechtigungen zum Erstellen, Löschen, Beschreiben, Auflisten und Aktualisieren generierter Vorlagen.

Außerdem müssen Sie jedem, der Ressourcen mit dem IaC-Generator scannt, Leseberechtigungen für die AWS Zieldienste gewähren. Weitere Informationen finden Sie unter Für das Scannen von Ressourcen sind IAM-Berechtigungen erforderlich.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"ResourceScanningOperations",
            "Effect":"Allow",
            "Action":[
                "cloudformation:DescribeResourceScan",
                "cloudformation:GetResource",
                "cloudformation:ListResources",
                "cloudformation:ListResourceScanRelatedResources",
                "cloudformation:ListResourceScanResources",
                "cloudformation:ListResourceScans",
                "cloudformation:ListTypes",
                "cloudformation:StartResourceScan"
            ],
            "Resource":"*"
        },
        {
            "Sid":"TemplateGeneration",
            "Effect":"Allow",
            "Action":[
                "cloudformation:CreateGeneratedTemplate",
                "cloudformation:DeleteGeneratedTemplate",
                "cloudformation:DescribeGeneratedTemplate",
                "cloudformation:GetResource",
                "cloudformation:GetGeneratedTemplate",
                "cloudformation:ListGeneratedTemplates",
                "cloudformation:UpdateGeneratedTemplate"
            ],
            "Resource":"*"
        }
    ]
}
```

AWS CloudFormation Servicerolle

Eine Servicerolle ist eine AWS Identity and Access Management (IAM-) Rolle, mit der CloudFormation Sie in Ihrem Namen Ressourcen in einem Stack aufrufen können. Sie können eine IAM-Rolle festlegen, die CloudFormation berechtigt, Stack-Ressourcen zu erstellen, zu aktualisieren oder zu löschen. CloudFormation Verwendet standardmäßig eine temporäre Sitzung, die aus Ihren Benutzeranmeldeinformationen für Stack-Operationen generiert wird. Wenn Sie eine Servicerolle angeben, CloudFormation verwendet die Anmeldeinformationen dieser Rolle.

Verwenden Sie eine Servicerolle, um explizit anzugeben, welche Aktionen ausgeführt werden CloudFormation können. Dabei handelt es sich möglicherweise nicht immer um dieselben Aktionen, die Sie oder andere Benutzer ausführen können. Beispielsweise verfügen Sie möglicherweise über Administratorrechte, können den CloudFormation Zugriff jedoch nur auf EC2 Amazon-Aktionen beschränken.

Sie erstellen die Service-Rolle und die Berechtigungsrichtlinie mit dem IAM-Service. Weitere Informationen zum Erstellen einer Servicerolle finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen für einen AWS Service im IAM-Benutzerhandbuch. Legen Sie CloudFormation (`cloudformation.amazonaws.com`) als den Dienst fest, der die Rolle annehmen kann.

Um eine Service-Rolle einem Stack zuzuordnen, legen Sie die Rolle beim Erstellen des Stack fest. Details hierzu finden Sie unter Konfigurieren Sie die Stack-Optionen. Sie können die Servicerolle auch ändern, wenn Sie den Stack in der Konsole oder den Stack über DeleteStackdie API aktualisieren. Bevor Sie eine Service-Rolle festlegen, stellen Sie sicher, dass Sie über die Berechtigung zum Weiterleiten (`iam:PassRole`) verfügen. Die `iam:PassRole`-Berechtigung gibt an, welche Rollen Sie verwenden können. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Gewähren von Benutzerberechtigungen zur AWS Übergabe einer Rolle an einen Dienst.

> **Important**
>
> Wenn Sie eine Servicerolle angeben, verwendet diese Rolle CloudFormation immer für alle Operationen, die auf diesem Stack ausgeführt werden. Es ist nicht möglich, eine Service-Rolle zu entfernen, die einem Stack zugewiesen ist, nachdem der Stack erstellt wurde. Andere Benutzer, die berechtigt sind, Operationen auf diesem Stack auszuführen, können diese Rolle verwenden, unabhängig davon, ob diese Benutzer über die `iam:PassRole` entsprechende Berechtigung verfügen oder nicht. Wenn die Rolle Berechtigungen umfasst, die der Benutzer nicht haben sollte, können Sie die Berechtigungen eines Benutzers

versehentlich weiterleiten. Stellen Sie sicher, dass die Rolle die geringsten Rechte zugesteht. Weitere Informationen finden Sie unter Anwenden von Berechtigungen mit geringsten Berechtigungen im IAM-Benutzerhandbuch.

Serviceübergreifende Confused-Deputy-Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In AWS kann ein dienstübergreifendes Identitätswechsels zu einem Problem mit dem verwirrten Stellvertreter führen. Ein serviceübergreifender Identitätswechsel kann auftreten, wenn ein Service (der Anruf-Service) einen anderen Service anruft (den aufgerufenen Service). Der Anruf-Service 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 Kontextschlüssel aws:SourceArnund die aws:SourceAccountglobalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken AWS CloudFormation , die einem anderen Dienst eine bestimmte Ressource, z. B. eine CloudFormation Erweiterung, gewähren. 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.

Stellen Sie sicher, dass der Wert von ein ARN der Ressource `aws:SourceArn` ist, die CloudFormation gespeichert wird.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. 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. Beispiel, `arn:aws:cloudformation:*:123456789012:*`.

Wenn der `aws:SourceArn` Wert die Konto-ID nicht enthält, müssen Sie beide globalen Bedingungskontextschlüssel verwenden, um die Berechtigungen einzuschränken.

Das folgende Beispiel zeigt, wie Sie die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen Bedingungsschlüssel in verwenden können, CloudFormation um das Problem des verwirrten Stellvertreters zu vermeiden.

Beispiel für eine Vertrauensrichtlinie, die Schlüssel **`aws:SourceArn`** und **`aws:SourceAccount`** Bedingungsschlüssel verwendet

Ruft bei Registrierungsdiensten AWS Security Token Service (AWS STS) auf, CloudFormation um eine Dienstrolle in Ihrem Konto zu übernehmen. Diese Rolle wird für `ExecutionRoleArn` im RegisterType-Vorgang konfiguriert und für `LogRoleArn` im LoggingConfig-Vorgang festgelegt. Weitere Informationen finden Sie unter Konfigurieren Sie eine Ausführungsrolle mit IAM-Berechtigungen und einer Vertrauensrichtlinie für den Zugriff auf öffentliche Erweiterungen.

Dieses Beispiel für eine Vertrauensrichtlinie für Rollen verwendet Bedingungsanweisungen, um die `AssumeRole` Funktionalität der Servicerolle auf Aktionen für die angegebene CloudFormation Erweiterung im angegebenen Konto zu beschränken. Die Bedingungen `aws:SourceArn` und `aws:SourceAccount` werden unabhängig ausgewertet. Jede Anforderung, die Servicerolle zu verwenden, muss beide Bedingungen erfüllen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "resources.cloudformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-east-1:123456789012:type/
resource/Organization-Service-Resource"
        }
```

```
        }
      }
    ]
  }
```

Zusätzliche Informationen

Richtlinien, die beispielsweise die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen Bedingungsschlüssel für eine von verwendete Servicerolle verwenden StackSets, finden Sie unterRichten Sie globale Schlüssel ein, um Confused-Deputy-Probleme zu mindern..

Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Aktualisieren einer Rollenvertrauensrichtlinie.

FAS-Anfragen (Forward Access Sessions) und Berechtigungsauswertung

Beim Erstellen, Aktualisieren und Löschen von CloudFormation Stacks können Benutzer optional einen IAM-Rollen-ARN angeben. Wenn keine Rolle bereitgestellt wird, CloudFormation verwendet es seinen Standarddienstmechanismus, um mit anderen AWS Diensten zu interagieren. In diesem Szenario muss der Aufrufer über die erforderlichen Berechtigungen für die verwalteten Ressourcen verfügen. Wenn ein Benutzer seine eigene IAM-Rolle angibt, CloudFormation übernimmt er diese Rolle, um in seinem Namen Dienstinteraktionen durchzuführen.

CloudFormation Generiert unabhängig davon, ob der Benutzer eine IAM-Rolle bereitstellt, für jeden Ressourcenvorgang ein neues, abgegrenztes FAS-Token. Folglich werden FAS-bezogene Bedingungsschlüssel, einschließlich`aws:ViaAWSService`, in beiden Szenarien aufgefüllt.

Die Verwendung von FAS wirkt sich darauf aus, wie IAM-Richtlinien während des Betriebs bewertet werden. CloudFormation Beim Erstellen eines Stacks mit einer Vorlage, die Ressourcen enthält, die von FAS-bezogenen Bedingungsschlüsseln betroffen sind, kann es zu Zugriffsverweigerungen kommen.

Beispiel für eine IAM-Richtlinie

Beachten Sie die folgende IAM-Richtlinie. `Statement2`verhindert konsequent die Erstellung einer `AWS::KMS::Key` Ressource in CloudFormation. Die Einschränkung wird konsistent durchgesetzt, unabhängig davon, ob während des Stack-Vorgangs eine IAM-Rolle bereitgestellt wird oder nicht. Das liegt daran, dass der `aws:ViaAWSService` Bedingungsschlüssel `true` aufgrund der Verwendung von FAS immer auf gesetzt ist.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [
                "kms:CreateKey"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "Statement2",
            "Effect": "Deny",
            "Action": [
                "kms:CreateKey"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Bool": {
                    "aws:ViaAWSService": "true"
                }
            }
        }
    ]
}
```

Beispiel für eine Stack-Vorlage

Wenn ein Benutzer beispielsweise einen Stack mit der folgenden Beispielvorlage erstellt, `aws:ViaAWSService` ist diese Einstellung auf festgelegt`true`, und die Rollenberechtigungen werden durch die FAS-Richtlinie außer Kraft gesetzt. Die Erstellung `Statement2` des Stacks wird durch die IAM-Richtlinie beeinflusst, die die Aktion verweigert. `CreateKey` Dies führt zu einem Fehler „Zugriff verweigert“.

```
Resources:
  myPrimaryKey:
    Type: AWS::KMS::Key
    Properties:
      Description: An example multi-Region primary key
      KeyPolicy:
        Version: '2012-10-17'
        Id: key-default-1
        Statement:
          - Sid: Enable IAM User Permissions
            Effect: Allow
            Principal:
              AWS: !Join
                - ''
                - - 'arn:aws:iam::'
                  - !Ref AWS::AccountId
                  - ':root'
            Action: kms:*
            Resource: '*'
```

Weitere Informationen zu FAS finden Sie unter Forward Access Sessions im IAM-Benutzerhandbuch.

> **Note**
>
> Die meisten Ressourcen halten sich an dieses Verhalten. Wenn Sie jedoch beim Erstellen, Aktualisieren oder Löschen einer Ressource einen unerwarteten Erfolg oder Misserfolg feststellen und Ihre IAM-Richtlinie FAS-bezogene Bedingungsschlüssel enthält, ist es wahrscheinlich, dass die betreffende Ressource zu einer kleinen Teilmenge von Ressourcen gehört, die diesem Standardmuster nicht folgen.

AWS CloudFormation API-Aufrufe protokollieren mit AWS CloudTrail

AWS CloudFormation ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden CloudFormation. CloudTrail erfasst alle API-Aufrufe CloudFormation als Ereignisse, einschließlich Aufrufe von der CloudFormation Konsole und von Codeaufrufen an die CloudFormation APIs. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung

von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für CloudFormation. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde CloudFormation, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

Themen

- CloudFormation Informationen in CloudTrail
- CloudFormation Logdateieinträge verstehen

CloudFormation Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet CloudFormation, wird diese Aktivität zusammen mit anderen CloudTrail AWS Serviceereignissen im Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS -Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für CloudFormation, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie ein Trail in der Konsole anlegen, gilt dieser für alle Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie unter:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle CloudFormation Aktionen werden von der AWS CloudFormation API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe der `ListStacks` Abschnitte `CreateStackDeleteStack`, und Einträge in den CloudTrail Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Gibt an, ob die Anforderung mit Root- oder IAM-Benutzer-Anmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter CloudTrail -Element userIdentity.

CloudFormation Logdateieinträge verstehen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der den `CreateStack` Vorgang demonstriert. Die Aktion wurde von dem IAM-Benutzer Alice ausgeführt.

> **Note**
>
> Nur die Schlüsselnamen der Eingabeparameter werden protokolliert. Parameterwerte werden nicht protokolliert.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
```

```
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:02:43Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "CreateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "templateURL": "templateURL",
    "tags": [
      {
        "key": "test",
        "value": "tag"
      }
    ],
    "stackName": "my-test-stack",
    "disableRollback": true,
    "parameters": [
      {
        "parameterKey": "password"
      },
      {
        "parameterKey": "securitygroup"
      }
    ]
  },
  "responseElements": {
    "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-stack/
a38e6a60-b397-11e3-b0fc-08002755629e"
  },
  "requestID": "9f960720-b397-11e3-bb75-a5b75389b02d",
  "eventID": "9bf6cfb8-83e1-4589-9a70-b971e727099b"
}
```

Das folgende Beispiel zeigt, dass Alice den `UpdateStack`-Vorgang auf dem `my-test-stack` Stack aufgerufen hat:

```
{
  "eventVersion": "1.01",
  "userIdentity": {
```

```
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:04:29Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "UpdateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "templateURL": "templateURL",
    "parameters": [
      {
        "parameterKey": "password"
      },
      {
        "parameterKey": "securitygroup"
      }
    ],
    "stackName": "my-test-stack"
  },
  "responseElements": {
    "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-stack/
a38e6a60-b397-11e3-b0fc-08002755629e"
  },
  "requestID": "def0bf5a-b397-11e3-bb75-a5b75389b02d",
  "eventID": "637707ce-e4a3-4af1-8edc-16e37e851b17"
}
```

Das folgende Beispiel zeigt, dass Alice den `ListStacks`-Vorgang aufgerufen hat.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
```

```
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:03:16Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "ListStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "b7d351d7-b397-11e3-bb75-a5b75389b02d",
  "eventID": "918206d0-7281-4629-b778-b91eb0d83ce5"
}
```

Das folgende Beispiel zeigt, dass Alice den `DescribeStacks`-Vorgang auf dem `my-test-stack`-Stack aufgerufen hat.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:06:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DescribeStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "stackName": "my-test-stack"
  },
  "responseElements": null,
  "requestID": "224f2586-b398-11e3-bb75-a5b75389b02d",
  "eventID": "9e5b2fc9-1ba8-409b-9c13-587c2ea940e2"
}
```

Das folgende Beispiel zeigt, dass Alice den `DeleteStack`-Vorgang auf dem `my-test-stack`-Stack aufgerufen hat.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:07:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DeleteStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "stackName": "my-test-stack"
  },
  "responseElements": null,
  "requestID": "42dae739-b398-11e3-bb75-a5b75389b02d",
  "eventID": "4965eb38-5705-4942-bb7f-20ebe79aa9aa"
}
```


Infrastruktursicherheit in AWS CloudFormation

Als verwalteter Dienst AWS CloudFormation ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff AWS CloudFormation über das Netzwerk. 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 Systeme 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.

Sie können diese API-Operationen von jedem Netzwerkstandort aus aufrufen, AWS CloudFormation unterstützt jedoch ressourcenbasierte Zugriffsrichtlinien, die Einschränkungen auf der Grundlage der Quell-IP-Adresse beinhalten können. Sie können auch AWS CloudFormation Richtlinien verwenden, um den Zugriff von bestimmten Amazon Virtual Private Cloud (Amazon VPC) -Endpunkten oder bestimmten zu kontrollieren. VPCs Dadurch wird der Netzwerkzugriff auf eine bestimmte AWS CloudFormation Ressource effektiv nur von der spezifischen VPC innerhalb des AWS Netzwerks isoliert.

Resilienz in AWS CloudFormation

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander 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 zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Konformitätsvalidierung für AWS CloudFormation

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen

und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Compliance und Governance im Bereich Sicherheit – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.
- Referenz für berechtigte HIPAA-Services – Listet berechtigte HIPAA-Services auf. Nicht alle AWS-Services sind HIPAA-fähig.
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dies AWS-Service bietet einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Die Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerelementreferenz.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Konfiguration und Schwachstellenanalyse in AWS CloudFormation

Konfiguration und IT-Kontrollen liegen in der gemeinsamen Verantwortung AWS von Ihnen, unserem Kunden. Weitere Informationen finden Sie im Modell der AWS gemeinsamen Verantwortung.

Bewährte Sicherheitsmethoden für CloudFormation

AWS CloudFormation 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

- Verwendung von IAM für die Zugriffskontrolle
- Keine Anmeldeinformationen in Vorlagen einbetten
- Wird verwendet AWS CloudTrail , um CloudFormation Anrufe zu protokollieren

Verwendung von IAM für die Zugriffskontrolle

IAM ist ein AWS Dienst, mit dem Sie Benutzer und deren Berechtigungen verwalten können. AWS Sie können IAM with verwenden CloudFormation , um festzulegen, welche CloudFormation Aktionen Benutzer ausführen können, z. B. das Anzeigen von Stack-Vorlagen, das Erstellen von Stacks oder das Löschen von Stacks. Darüber hinaus benötigt jeder, der CloudFormation Stacks verwaltet, Berechtigungen für Ressourcen innerhalb dieser Stacks. Wenn Benutzer beispielsweise EC2 Amazon-Instances starten, aktualisieren oder beenden möchten, müssen sie über die Erlaubnis verfügen, die entsprechenden EC2 Amazon-Aktionen aufzurufen. CloudFormation

In den meisten Fällen benötigen Benutzer vollen Zugriff, um alle Ressourcen in einer Vorlage zu verwalten. CloudFormation führt Aufrufe durch, um diese Ressourcen in ihrem Namen zu erstellen, zu ändern und zu löschen. Verwenden Sie eine Servicerolle, um die Berechtigungen zwischen einem Benutzer und dem CloudFormation Service zu trennen. CloudFormation verwendet anstelle der Benutzerrichtlinie die Richtlinie der Servicerolle, um Anrufe zu tätigen. Weitere Informationen finden Sie unter AWS CloudFormation Servicerolle.

Keine Anmeldeinformationen in Vorlagen einbetten

Anstatt vertrauliche Informationen in Ihre CloudFormation Vorlagen einzubetten, empfehlen wir Ihnen, dynamische Verweise in Ihrer Stack-Vorlage zu verwenden.

Dynamische Verweise bieten eine kompakte und leistungsstarke Möglichkeit, auf externe Werte zu verweisen, die in anderen Diensten gespeichert und verwaltet werden, z. B. im AWS Systems Manager Parameter Store oder AWS Secrets Manager. Wenn Sie eine dynamische Referenz verwenden, CloudFormation ruft sie den Wert der angegebenen Referenz ab, falls dies bei Stack- und Änderungsset-Vorgängen erforderlich ist, und übergibt den Wert an die entsprechende Ressource. Speichert jedoch CloudFormation niemals den tatsächlichen Referenzwert. Weitere Informationen finden Sie unter Ruft mithilfe dynamischer Verweise Werte ab, die in anderen Diensten gespeichert sind.

AWS Secrets Manager hilft Ihnen, die Anmeldeinformationen für Ihre Datenbanken und Services sicher zu verschlüsseln, zu speichern und wieder abzurufen. Der AWS Systems Manager Parameter Store ermöglicht eine sichere, hierarchische Speicherung für die Konfigurationsdatenverwaltung.

Für weitere Informationen zum Definieren von Vorlagenparametern siehe CloudFormation ParametersVorlagensyntax.

Wird verwendet AWS CloudTrail , um CloudFormation Anrufe zu protokollieren

AWS CloudTrail verfolgt jeden, der CloudFormation API-Aufrufe in Ihrem macht AWS-Konto. API-Aufrufe werden protokolliert, wenn jemand die CloudFormation API, die CloudFormation Konsole, eine Back-End-Konsole oder CloudFormation AWS CLI Befehle verwendet. Aktivieren Sie die Protokollierung und legen Sie einen Amazon-S3-Bucket zum Speichern der Protokolle fest. Auf diese Weise können bei Bedarf Sie nachvollziehen, wer welche CloudFormation-Aufruf in Ihrem Konto ausgeführt hat. Weitere Informationen finden Sie unter AWS CloudFormation API-Aufrufe protokollieren mit AWS CloudTrail.

Zugriff CloudFormation über einen Schnittstellenendpunkt (AWS PrivateLink)

Sie können verwenden AWS PrivateLink , um eine private Verbindung zwischen Ihrer VPC und CloudFormation herzustellen. Sie können darauf zugreifen, CloudFormation als ob es in Ihrer VPC

wäre, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder AWS Direct Connect eine Verbindung zu verwenden. Instances in Ihrer VPC benötigen für den Zugriff CloudFormation keine öffentlichen IP-Adressen.

Sie stellen diese private Verbindung her, indem Sie einen Schnittstellen-Endpunkt erstellen, der von AWS PrivateLink unterstützt wird. Wir erstellen eine Endpunkt-Netzwerkschnittstelle in jedem Subnetz, das Sie für den Schnittstellen-Endpunkt aktivieren. Hierbei handelt es sich um vom Anforderer verwaltete Netzwerkschnittstellen, die als Eingangspunkt für den Datenverkehr dienen, der für CloudFormation bestimmt ist.

CloudFormation unterstützt Aufrufe aller API-Aktionen über den Schnittstellenendpunkt.

Überlegungen zu CloudFormation VPC-Endpunkten

Bevor Sie einen Schnittstellenendpunkt einrichten, stellen Sie zunächst sicher, dass Sie die Voraussetzungen im Thema Zugriff auf einen AWS Dienst mithilfe eines VPC-Schnittstellen-Endpunkts im AWS PrivateLink Handbuch erfüllt haben.

Die folgenden zusätzlichen Voraussetzungen und Überlegungen gelten für die Einrichtung eines Schnittstellenendpunkts für CloudFormation:

- Wenn Sie Ressourcen in Ihrer VPC haben, die auf eine benutzerdefinierte Ressourcenanfrage oder eine Wartebedingung antworten müssen, stellen Sie sicher, dass sie Zugriff auf die erforderlichen CloudFormation spezifischen Amazon S3 S3-Buckets haben. CloudFormation verfügt über S3-Buckets in jeder Region, um Antworten auf eine benutzerdefinierte Ressourcenanfrage oder eine Wartebedingung zu überwachen. Wenn eine Vorlage benutzerdefinierte Ressourcen oder Wartebedingungen in einer VPC enthält, muss die VPC-Endpunktrichtlinie Benutzern ermöglichen, Antworten auf die folgenden Buckets zu senden:
  - Lassen Sie bei benutzerdefinierten Ressourcen Datenverkehr zum `cloudformation-custom-resource-response-`*`region`*-Bucket zu. Bei der Verwendung benutzerdefinierter Ressourcen enthalten AWS-Region Namen keine Bindestriche. Beispiel, `uswest2`.
  - Lassen Sie bei Wartebedingungen Datenverkehr zum `cloudformation-waitcondition-`*`region`*-Bucket zu. Bei der Verwendung von Wartebedingungen enthalten AWS-Region Namen Bindestriche. Beispiel, `us-west-2`.

  Wenn die Endpunktrichtlinie den Datenverkehr zu diesen Buckets blockiert, CloudFormation werden keine Antworten empfangen und der Stack-Vorgang schlägt fehl. Wenn Sie beispielsweise eine Ressource in einer VPC in der `us-west-2`-Region haben, die auf eine Wartebedingung

reagieren muss, muss die Ressource in der Lage sein, eine Antwort auf den `cloudformation-waitcondition-us-west-2`-Bucket zu senden.

Eine Liste der Anbieter, die AWS-Regionen CloudFormation derzeit verfügbar sind, finden Sie auf der Seite AWS CloudFormation Endpunkte und Kontingente im. Allgemeine Amazon Web Services-Referenz

- VPC-Endpunkte unterstützen derzeit keine regionsübergreifenden Anfragen. Stellen Sie sicher, dass Sie Ihren Endpunkt in derselben Region erstellen, in der Sie Ihre API-Aufrufe tätigen möchten. CloudFormation
- VPC-Endpunkte unterstützen nur Amazon-bereitgestellte DNS über Route 53. Wenn Sie Ihre eigene DNS verwenden möchten, können Sie die bedingte DNS-Weiterleitung nutzen. Weitere Informationen finden Sie unter DHCP-Optionssätze in Amazon VPC im Amazon VPC-Benutzerhandbuch.
- Die Sicherheitsgruppe für den VPC-Endpunkt müssen eingehende Verbindungen auf Port 443 aus dem privaten Subnetz der VPC zulassen.

Erstellen eines Schnittstellen-VPC-Endpunkts für CloudFormation

Sie können einen VPC-Endpunkt für die CloudFormation Verwendung entweder der Amazon VPC-Konsole oder der AWS Command Line Interface ()AWS CLI erstellen. Weitere Informationen finden Sie unter Erstellen eines VPC-Endpunkts im AWS PrivateLink -Leitfaden.

Erstellen Sie einen Schnittstellenendpunkt für die CloudFormation Verwendung des folgenden Servicenamens:

- com.amazonaws. *`region`*. Wolkenbildung

Wenn Sie privates DNS für den Schnittstellenendpunkt aktivieren, können Sie API-Anfragen CloudFormation unter Verwendung des standardmäßigen regionalen DNS-Namens stellen. Beispiel, `cloudformation.us-east-1.amazonaws.com`.

Erstellen einer VPC-Endpunktrichtlinie für CloudFormation

Eine Endpunktrichtlinie ist eine IAM-Ressource, die Sie an einen Schnittstellen-Endpunkt anfügen können. Die standardmäßige Endpunktrichtlinie ermöglicht den vollen Zugriff CloudFormation über den Schnittstellenendpunkt. Um den Zugriff zu kontrollieren, der CloudFormation von Ihrer VPC aus gewährt wird, fügen Sie dem Schnittstellenendpunkt eine benutzerdefinierte Endpunktrichtlinie hinzu.

Eine Endpunktrichtlinie gibt die folgenden Informationen an:

- Die Prinzipale, die Aktionen ausführen können (AWS-Konten, IAM-Benutzer und IAM-Rollen).
- Aktionen, die ausgeführt werden können
- Die Ressourcen, auf denen die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie im Handbuch unter Steuern des Zugriffs auf VPC-Endpunkte mithilfe von Endpunktrichtlinien.AWS PrivateLink

Beispiel: VPC-Endpunktrichtlinie für CloudFormation-Aktionen

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für. CloudFormation Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die aufgelisteten CloudFormation Aktionen. Das folgende Beispiel verweigert allen Benutzern die Berechtigung, Stacks über den VPC-Endpunkt zu erstellen, und gewährt vollen Zugriff auf alle anderen Aktionen im Service. CloudFormation

```
{
  "Statement": [
    {
      "Action": "cloudformation:*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    },
    {
      "Action": "cloudformation:CreateStack",
      "Effect": "Deny",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

Überwachung CloudFormation und Git-Synchronisierung von Ereignissen mit EventBridge

Amazon EventBridge ist ein serverloser Service, der Ereignisse verwendet, um Anwendungskomponenten miteinander zu verbinden, sodass Sie leichter skalierbare, ereignisgesteuerte Anwendungen erstellen können. Bei der ereignisgesteuerten Architektur werden lose gekoppelte Softwaresysteme entwickelt, die zusammenarbeiten, indem sie Ereignisse senden und darauf reagieren. Ereignisse stellen eine Veränderung in einer Ressource oder Umgebung dar.

CloudFormation Generiert wie bei vielen AWS Diensten Ereignisse und sendet sie an den EventBridge Standard-Event-Bus, der in jedem AWS-Konto automatisch bereitgestellt wird. Ein Event Bus ist ein Router, der Ereignisse empfängt und sie an null oder mehr Ziele weiterleitet. Regeln, die Sie für den Event-Bus angeben, werten Ereignisse aus, sobald sie eintreffen. Jede Regel prüft, ob ein Ereignis mit dem Ereignismuster der Regel übereinstimmt. Wenn das Ereignis übereinstimmt, sendet der Event-Bus das Ereignis an die angegebenen Ziele.

Weitere Informationen finden Sie unter Erste Schritte mit Amazon EventBridge im EventBridge Amazon-Benutzerhandbuch.

Themen

- CloudFormation und Übersicht über Git-Sync-Ereignisse
- EventBridge Amazon-Berechtigungen
- Erstellen eines benutzerdefinierten Ereignismusters für eine EventBridge Regel
- CloudFormation Referenz zu den Einzelheiten der Ereignisse

CloudFormation und Übersicht über Git-Sync-Ereignisse

CloudFormation sendet Ereignisse an EventBridge jedes Mal, wenn ein Vorgang zum Erstellen, Aktualisieren, Löschen oder zur Erkennung von Abweichungen auf einem Stack ausgeführt wird. CloudFormation sendet auch Ereignisse an, EventBridge um Statusänderungen an Stack-Sets und Stack-Set-Instanzen vorzunehmen. Sie können EventBridge Regeln verwenden, um Ereignisse an Ihre definierten Ziele weiterzuleiten. Es ist garantiert, dass diese Ereignisse zugestellt werden, und es kann sein, dass sie nicht in der richtigen Reihenfolge zugestellt werden.

Da CloudFormation Ereignisse Änderungen an Stacks oder Stack-Sets und ihren Ressourcen darstellen, können Sie sie verwenden, um Workflows zu initiieren, die mit den jeweiligen Ereignissen verknüpft sind. Zum Beispiel:

- Erstellen Sie Stack- oder Stack-Set-spezifische Tags für alle Ressourcen, die über CloudFormation bereitgestellt werden.
- Stellen Sie eine Zuordnung zwischen einem CloudFormation Stack oder einem Stack-Set und einem Amazon WorkSpaces Application Manager (Amazon WAM) her.
- Geben Sie eine Zuordnung zu einem AppRegistry für den erstellten Stack oder Stack-Set an.

Die folgenden Ereignisse werden vom Standardereignisbus in generiert CloudFormation und an diesen gesendet EventBridge. Weitere Informationen finden Sie unter ???.

Ereignistyp	Beschreibung	Ereignisquelle
Änderung des Ressourcenstatus	Aktualisierungen, die für einen Stack durchgefü hrt werden und die zugrunde liegenden Ressourceneigenschaften ändern. Eine vollständige Liste der unterstützten AWS Ressourcentypen finden Sie in der Referenz zu AWS Ressourcen- und Eigenschaftstypen.	AWS CloudFormation
Änderung des Stack-Status	Stellt eine Statusänderung eines bestimmten Stacks dar. Einzelheiten zum Code finden Sie unterStack-Statuscodes.	AWS CloudFormation
Änderung des Status der Drift-Erkennung	Stellt eine vom Benutzer initiierte Aktualisi erung der Abweichungserkennung für einen bestimmten Stack dar. Eine vollständige Liste der vollständig veränderl ichen und unveränderlichen Typen, die die Drifterkennung unterstützen, finden Sie unter Unterstützung für Ressourcentypen	AWS CloudFormation

Ereignistyp	Beschreibung	Ereignisquelle
StackSetÄnderung des Status	Stellt eine Statusänderung eines bestimmten Stack-Sets dar.	AWS CloudFormation
StackSet Änderung des Status der Stack-Instance	Stellt eine Statusänderung einer bestimmten StackSet Stack-Instance dar. Einzelheiten zum Code finden Sie unterStack-Instance-Statuscodes.	AWS CloudFormation
StackSet Betriebsstatus	Stellt eine Statusänderung einer bestimmten StackSet Operation dar. Einzelheiten zum Code finden Sie unterStackSets Statuscodes.	AWS CloudFormation

Darüber hinaus sendet AWS CloudFormation Git Sync Ereignisse für Statusänderungen für Repository-Synchronisierungen und Ressourcensynchronisierungen an. EventBridge

Die folgenden Git-Sync-Ereignisse werden vom Standardereignisbus in generiert CodeConnections und an diesen gesendet EventBridge. Weitere Informationen finden Sie unter ???.

Ereignistyp	Beschreibung	Ereignisquelle
Änderung des Repository-Synchronisierungsstatus	Stellt eine Statusänderung einer Git-Repository-Synchronisierung dar.	AWS CodeConnections
Änderung des Status der Ressourcensynchronisierung	Stellt eine Statusänderung einer Git-Ressourcensynchronisierung dar.	AWS CodeConnections

EventBridge Amazon-Berechtigungen

CloudFormation benötigt keine zusätzlichen Berechtigungen für die Übermittlung von Ereignissen an EventBridge. Die Ereignisse enthalten Informationen, die bereits über die CloudFormation API-Operationen verfügbar sind.

Für die von Ihnen angegebenen Ziele sind möglicherweise bestimmte Berechtigungen oder Konfigurationen erforderlich. Weitere Informationen zur Verwendung bestimmter Dienste für Ziele finden Sie unter EventBridge Amazon-Ziele im EventBridge Amazon-Benutzerhandbuch.

Erstellen eines benutzerdefinierten Ereignismusters für eine EventBridge Regel

In EventBridge for CloudFormation - und Git-Sync-Ereignissen finden Sie mehrere vordefinierte Muster. Dies vereinfacht die Erstellung eines Ereignismusters. Anstatt deine eigenen Ereignismuster zu schreiben, kannst du Feldwerte in einem Formular auswählen und das Muster dann für dich EventBridge generieren. Sie können eine neue Regel mit einem dieser vordefinierten Ereignismuster erstellen oder Ihr eigenes benutzerdefiniertes Ereignismuster erstellen.

Wenn ein Dienst z. B. ein Ereignis an den Standard-Ereignisbus CloudFormation übermittelt, EventBridge verwendet er das in Ihrer Regel definierte Ereignismuster, um zu bestimmen, ob das Ereignis an die Ziele der Regel gesendet werden soll. Ein Ereignismuster entspricht den Daten in den gewünschten CloudFormation Ereignissen.

Jedes Ereignismuster ist ein JSON-Objekt, das Folgendes enthält:

- Ein `source` Attribut, das den Dienst identifiziert, von dem das Ereignis stammt. Zum Beispiel `aws.cloudformation` oder `aws.codeconnections`.
- (Optional): Ein `detail-type`-Attribut, das ein Array der zuzuordnenden Ereignistypen enthält.
- (Optional): Ein `detail`-Attribut, das alle anderen Ereignisdaten für den Abgleich enthält.

  Dazu gehören zum Beispiel die Stack-ID, die beteiligten Ressourcen, der Status verschiedener Ressourcen und andere Daten, die für einen bestimmten Ereignistyp relevant sind.

Das folgende Ereignismuster entspricht beispielsweise allen Ereignissen zur Änderung des Ressourcenstatus:

```
{
```

```
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"]
}
```

Das folgende Ereignismuster verwendet zwar Ereignisdetaildaten, um nur Ereignisse zur Änderung des Ressourcenstatus abzugleichen, bei denen eine neue `AWS::SNS::Topic` Ressource `AWS::S3::Bucket` oder eine Ressource CloudFormation erstellt wird:

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"],
  "detail": {
    "status-details": {
      "status": ["CREATE_COMPLETE"]
    },
    "resource-type": ["AWS::S3::Bucket", "AWS::SNS::Topic"]
  }
}
```

Weitere Informationen zum Schreiben von Ereignismustern finden Sie unter Ereignismuster im EventBridge Amazon-Benutzerhandbuch.

CloudFormation Referenz zu den Einzelheiten der Ereignisse

Alle Ereignisse von AWS Diensten haben einen gemeinsamen Satz von Feldern, die Metadaten über das Ereignis enthalten, z. B. den AWS Dienst, der die Quelle des Ereignisses ist, den Zeitpunkt, zu dem das Ereignis generiert wurde, das Konto und die Region, in der das Ereignis stattgefunden hat, und andere. Definitionen dieser allgemeinen Felder finden Sie unter Metadaten für AWS Dienstereignisse in der AWS Ereignisreferenz.

Darüber hinaus weist jedes Ereignis ein `detail`-Feld auf, das spezifische Daten für das betreffende Ereignis enthält. In der folgenden Referenz werden die Detailfelder für die verschiedenen CloudFormation Ereignisse definiert.

Bei der EventBridge Auswahl und Verwaltung von CloudFormation Ereignissen ist es hilfreich, Folgendes zu beachten:

- Das `source` Feld gibt die Ereignisquelle an.

  Zum Beispiel `aws.cloudformation` oder `aws.codeconnections`.

- Das Feld `detail-type` gibt den Ereignistyp an.

  Zum Beispiel `CloudFormation Resource Status Change` oder `CloudFormation Drift Detection Status Change`.
- Das Feld `detail` enthält die Daten, die für das betreffende Ereignis spezifisch sind.

  Dazu gehören zum Beispiel die Stack-ID, die beteiligten Ressourcen, der Status verschiedener Ressourcen und andere Daten, die für einen bestimmten Ereignistyp relevant sind.

Informationen zur Erstellung von Ereignismustern, die es Regeln ermöglichen, CloudFormation Ereignissen zuzuordnen, finden Sie unter Ereignismuster im EventBridge Amazon-Benutzerhandbuch.

Weitere Informationen zu Ereignissen und deren EventBridge Verarbeitung finden Sie unter EventBridgeEreignisse im EventBridge Amazon-Benutzerhandbuch.

Themen

- Details zum Ereignis „Änderung des Ressourcenstatus“
- Details zum Ereignis „Stack-Status ändern“
- Drift-Erkennungsstatus — Ereignisdetail ändern
- StackSet Details zum Ereignis „Statusänderung“
- StackSet Einzelheiten zum Ändern des Stack-Instance-Status
- StackSet Einzelheiten zum Ereignis „Betriebsstatus ändern“
- Details zum Ändern des Repository-Synchronisierungsstatus
- Details zum Ereignis „Status der Ressourcensynchronisierung ändern“

Details zum Ereignis „Änderung des Ressourcenstatus“

Im Folgenden finden Sie die Detailfelder für Ereignisse zur Änderung des Ressourcenstatus.

Die `detail-type` Felder `source` und sind enthalten, da sie spezifische Werte für AWS CloudFormation Ereignisse enthalten.

```
{
  . . .,
  "detail-type": "CloudFormation Resource Status Change",
```

```
  "source": "aws.cloudformation",
  . . .,
  "detail": {
    "stack-id" : "string",
    "logical-resource-id" : "string",
    "physical-resource-id": "string",
    "status-details": {
        "status": "string",
        "status-reason": "string"
    },
     "resource-type": "string",
     "client-request-token": "string"
  }
}
```

`detail-type`

Identifiziert den Ereignistyp.

Bei Ressourcenstatusereignissen lautet dieser Wert `CloudFormation Resource Status Change`.

`source`

Identifiziert den Service, aus dem das Ereignis stammt. Für CloudFormation Ereignisse ist dieser Wert`aws.cloudformation`.

`detail`

Ein JSON-Objekt, das Informationen zum Ereignis enthält. Der Service, der das Ereignis generiert, bestimmt den Inhalt dieses Feldes.

Bei Ressourcenstatusereignissen gehören zu diesen Daten:

`stack-id`

Die eindeutige Stack-ID, die dem Stack zugeordnet ist.

`logical-resource-id`

Der in der Vorlage definierte logische Name der Ressource.

`physical-resource-id`

Der Name oder die eindeutige Kennung, die einer physischen Instanz-ID einer Ressource entspricht, die von unterstützt wird CloudFormation.

`status-details`

`status`

Status der Ressource.

`status-reason`

Grund für den Status der Ressource.

`resource-type`

Typ der Ressource. Beispiel, `AWS::S3::Bucket`.

`client-request-token`

Ein Zugriffstoken für API-Aufrufe. Allen Ereignissen, die von einem bestimmten Stack-Vorgang ausgelöst werden, wird dasselbe Client-Anfrage-Token zugewiesen, das Sie verwenden können, um Vorgänge zu verfolgen. Stack-Operationen, die von der Konsole aus initiiert werden, verwenden das Token-Format Console- StackOperation -ID, mit dem Sie den Stack-Vorgang leicht identifizieren können. Wenn Sie beispielsweise einen Stack unter Verwendung der Konsole erstellen, würde jedem resultierenden Stack-Ereignis dasselbe Token im folgenden Format zugewiesen: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Example Beispiel: Ereignis „Änderung des Ressourcenstatus“

Nachfolgend ist ein beispielhaftes Ressourcenstatusereignis aufgeführt. Dieses Ereignis CloudFormation gibt an, dass die angeforderte Ressource, ein Amazon S3 S3-Bucket, erfolgreich im angegebenen Stack erstellt wurde.

```
{
    "version":"0",
    "id":"6a7e8feb-b491-4cf7-a9f1-bf3703467718",
    "detail-type":"CloudFormation Resource Status Change",
    "source":"aws.cloudformation",
    "account":"111122223333",
    "time":"2017-12-22T18:43:48Z",
    "region":"us-west-1",
    "resources":[
        "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack"
    ],
    "detail":{
        "stack-id":"arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
```

```
        "logical-resource-id":"my-s3-bucket",
        "physical-resource-id":"arn:aws:s3:::my-s3-bucket-us-east-1",
        "status-details":{
            "status":"CREATE_COMPLETE",
            "status-reason":""
        },
        "resource-type":"AWS::S3::Bucket",
        "client-request-token":""
    }
}
```

Details zum Ereignis „Stack-Status ändern“

Im Folgenden finden Sie die Detailfelder für Ereignisse zur Änderung des Stack-Status.

Die `detail-type` Felder `source` und sind enthalten, da sie spezifische Werte für AWS CloudFormation Ereignisse enthalten.

```
{
  . . .,
  "detail-type":"CloudFormation Stack Status Change",
  "source":"aws.cloudformation",
  . . .,
  "detail":{
    "stack-id":"string",
    "status-details":{
      "status":"string",
      "status-reason":"string"
    },
    "client-request-token":"string"
  }
}
```

`detail-type`

Identifiziert den Ereignistyp.

Bei Stack-Statusereignissen lautet dieser Wert `CloudFormation Stack Status Change`.

`source`

Identifiziert den Service, aus dem das Ereignis stammt. Für CloudFormation Ereignisse ist dieser Wert`aws.cloudformation`.

`detail`

Ein JSON-Objekt, das Informationen zum Ereignis enthält. Der Service, der das Ereignis generiert, bestimmt den Inhalt dieses Feldes.

Bei Stack-Statusereignissen gehören zu diesen Daten:

`stack-id`

Die eindeutige, dem Stack zugeordnete Stack-ID.

`status-details`

`status`

Status des Stacks.

Eine vollständige Liste der Stack-Statuscodes finden Sie unterStack-Statuscodes.

`status-reason`

Grund für den Status der Ressource.

`client-request-token`

Ein Zugriffstoken für API-Aufrufe. Allen Ereignissen, die von einem bestimmten Stack-Vorgang ausgelöst werden, wird dasselbe Client-Anfrage-Token zugewiesen, das Sie verwenden können, um Vorgänge zu verfolgen. Stack-Operationen, die von der Konsole aus initiiert werden, verwenden das Token-Format Console- StackOperation -ID, mit dem Sie den Stack-Vorgang leicht identifizieren können. Wenn Sie beispielsweise einen Stack unter Verwendung der Konsole erstellen, würde jedem resultierenden Stack-Ereignis dasselbe Token im folgenden Format zugewiesen: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Example Beispiel: Stack-Status-Ereignis

Im Folgenden finden Sie ein Beispiel für ein Stack-Status-Ereignis, bei dem CloudFormation der angeforderte Stack erfolgreich erstellt wurde`teststack`.

```
{
    "version":"0",
    "id":"6a7e8feb-b491-4cf7-a9f1-bf3703467718",
    "detail-type":"CloudFormation Stack Status Change",
```

```
    "source":"aws.cloudformation",
    "account":"111122223333",
    "time":"2017-12-22T18:43:48Z",
    "region":"us-west-1",
    "resources":[
        "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack"
    ],
    "detail":{
        "stack-id":"arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
        "status-details":{
            "status":"CREATE_COMPLETE",
            "status-reason":""
        },
        "client-request-token":""
    }
}
```

Drift-Erkennungsstatus — Ereignisdetail ändern

Im Folgenden finden Sie die Detailfelder für Ereignisse zur Stack-Abweichungserkennung.

Die `detail-type` Felder `source` und sind enthalten, da sie spezifische Werte für AWS CloudFormation Ereignisse enthalten.

```
{
  . . .,
  "detail-type":"CloudFormation Drift Detection Status Change",
  "source":"aws.cloudformation",
  . . .,
  "detail":{
    "stack-id":"string",
    "stack-drift-detection-id":"string",
    "status-details":{
      "stack-drift-status":"string",
      "detection-status":"string"
    },
      "drift-detection-details":{
        "drifted-stack-resource-count":integer
      },
    "client-request-token":"string"
  }
}
```

`detail-type`

Identifiziert den Ereignistyp.

Bei Stack-Abweichungserkennungsereignissen lautet dieser Wert `CloudFormation Drift Detection Status Change`.

`source`

Identifiziert den Service, aus dem das Ereignis stammt. Für CloudFormation Ereignisse ist dieser Wert`aws.cloudformation`.

`detail`

Ein JSON-Objekt, das Informationen zum Ereignis enthält. Der Service, der das Ereignis generiert, bestimmt den Inhalt dieses Feldes.

Bei Stack-Abweichungserkennungsereignissen gehören zu diesen Daten:

`stack-id`

Die eindeutige Stack-ID, die dem Stack zugeordnet ist.

`stack-drift-detection-id`

Die Stack-Abweichungserkennungs-ID.

`status-details`

`stack-drift-status`

Abweichungsstatus des Stack.

`detection-status`

Status des Abweichungserkennungs-Vorgangs.

`drift-detection-details`

`drifted-stack-resource-count`

Anzahl der abweichenden Ressourcen. Wenn der Wert `-1` ist, ist die Abweichungserkennung im Gange. Alle anderen nicht negativen Ganzzahlen stellen die tatsächliche Anzahl der abweichenden Ressourcen dar.

`client-request-token`

Ein Zugriffstoken für API-Aufrufe. Allen Ereignissen, die von einem bestimmten Stack-Vorgang ausgelöst werden, wird dasselbe Client-Anfrage-Token zugewiesen, das Sie verwenden

können, um Vorgänge zu verfolgen. Stack-Operationen, die von der Konsole aus initiiert werden, verwenden das Token-Format Console- StackOperation -ID, mit dem Sie den Stack-Vorgang leicht identifizieren können. Wenn Sie beispielsweise einen Stack unter Verwendung der Konsole erstellen, würde jedem resultierenden Stack-Ereignis dasselbe Token im folgenden Format zugewiesen: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Example Beispiel: Ereignis zur Erkennung von Stack-Drift

Im Folgenden finden Sie ein Beispiel für ein Stack Drift Detection-Ereignis. Dieses Ereignis gibt an, CloudFormation ob die Drift-Erkennung auf dem angegebenen Stack abgeschlossen wurde und dass der Stack derzeit `DRIFTED` aufgrund einer Drift-Ressource den Drift-Status hat.

```
{
    "version":"0",
    "id":"6a7e8feb-b491-4cf7-a9f1-bf3703467718",
    "detail-type":"CloudFormation Drift Detection Status Change",
    "source":"aws.cloudformation",
    "account":"111122223333",
    "time":"2017-12-22T18:43:48Z",
    "region":"us-west-1",
    "resources": ["string"],
    "detail":{
        "stack-id":"arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
        "stack-drift-detection-id":"624af370-311a-11e8-b6b7-500cexample",
        "status-details":{
            "stack-drift-status":"DRIFTED",
            "detection-status":"DETECTION_COMPLETE"
        },
        "drift-detection-details":{
            "drifted-stack-resource-count":1
        },
    "client-request-token":""
    }
}
```

StackSet Details zum Ereignis „Statusänderung“

Im Folgenden finden Sie die Detailfelder für StackSet Statusänderungsereignisse.

Die `detail-type` Felder `source` und sind enthalten, da sie spezifische Werte für AWS CloudFormation Ereignisse enthalten.

```
{
  . . .,
  "detail-type": "CloudFormation StackSet Status Change",
  "source": "aws.cloudformation",
  . . .,
  "detail": {
    "stack-set-arn" : "string",
    "status-details": {
        "status":"string"
    }
  }
}
```

`detail-type`

Identifiziert den Ereignistyp.

Für StackSet Statusereignisse ist dieser Wert`CloudFormation StackSet Status Change`.

`source`

Identifiziert den Service, aus dem das Ereignis stammt. Für CloudFormation Ereignisse ist dieser Wert`aws.cloudformation`.

`detail`

Ein JSON-Objekt, das Informationen zum Ereignis enthält. Der Service, der das Ereignis generiert, bestimmt den Inhalt dieses Feldes.

Für StackSet Statusereignisse umfassen diese Daten:

`stack-set-arn`

Der Amazon-Ressourcenname (ARN), der dem Stack-Set zugeordnet ist.

`status-details`

`status`

Der StackSet Status.

Gültige Werte: `ACTIVE` | `DELETED`

Example Beispiel: Ereignis StackSet „Statusänderung“

Im Folgenden finden Sie ein Beispiel für ein StackSet Statusänderungsereignis. Dieses Ereignis gibt an, CloudFormation dass das angegebene Stack-Set gelöscht wurde.

```
{
  "version": "0",
  "id": "42h6hb90-hg0w-11op-b01v-0xhnh0934z09",
  "detail-type": "CloudFormation StackSet Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-23T17:06:18Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:111122223333:stackset/test12345:3f3a3fbe-
c937-4eb3-a87d-e36a0af3f663"
  ],
  "detail": {
    "stack-set-arn" : "arn:aws:cloudformation:us-east-1:111122223333:stackset/
test12345:3f3a3fbe-c937-4eb3-a87d-e36a0af3f663",
    "status-details": {
        "status":"DELETED"
    }
  }
}
```

StackSet Einzelheiten zum Ändern des Stack-Instance-Status

Im Folgenden finden Sie die Detailfelder für Statusereignisse der StackSet Stack-Instanz.

Die `detail-type` Felder `source` und sind enthalten, da sie spezifische Werte für AWS CloudFormation Ereignisse enthalten.

```
{
  . . .,
  "detail-type": "CloudFormation StackSet StackInstance Status Change",
  "source": "aws.cloudformation",
  . . .,
  "detail": {
    "stack-set-arn" : "string",
    "stack-id" : "string",
    "action" : "string",
    "status-details": {
```

```
        "status": "string",
        "status-reason": "string",
        "detailed-status": "string"
      }
    }
  }
}
```

`detail-type`

Identifiziert den Ereignistyp.

Für Statusereignisse der StackSet Stack-Instanz lautet dieser Wert`CloudFormation StackSet StackInstance Status Change`.

`source`

Identifiziert den Service, aus dem das Ereignis stammt. Für CloudFormation Ereignisse ist dieser Wert`aws.cloudformation`.

`detail`

Ein JSON-Objekt, das Informationen zum Ereignis enthält. Der Service, der das Ereignis generiert, bestimmt den Inhalt dieses Feldes.

Bei Statusereignissen für StackSet Stack-Instances beinhalten diese Daten:

`stack-set-arn`

Der Amazon-Ressourcenname (ARN), der dem zugeordnet ist StackSet.

`stack-id`

Die eindeutige Stack-ID, die der Stack-Instance zugeordnet ist.

`action`

Die Art des Stack-Set-Vorgangs.

Zulässige Werte: `CREATE` | `UPDATE` | `DELETE` | `DETECT_DRIFT`

`status-details`

`status`

Der StackSet Instanzstatus.

Weitere Details finden Sie unter Stack-Instance-Statuscodes.

Zulässige Werte: `CURRENT` | `OUTDATED` | `INOPERABLE`

`status-reason`

Grund für den Status der StackSet Instanz.

`detailed-status`

Der detaillierte Status der StackSet Instanz.

Gültige Werte: `CANCELLED` `FAILED` | `FAILED_IMPORT` | `INOPERABLE` | `PENDING` | `RUNNING` | `SKIPPED_SUSPENDED_ACCOUNT` | `SUCCEEDED`

Example Beispiel: Ereignis „ StackSet Stack Instance Status Change“

Im Folgenden finden Sie ein Beispiel für ein StackSet Stack Instance Status Change-Ereignis.

```
{
  "version": "0",
  "id": "42h6hb90-hg0w-11op-b01v-0xhnh0934z09",
  "detail-type": "CloudFormation StackSet StackInstance Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-22T19:19:23Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-
d041-44ad-94f8-b8268aca1e59"
  ],
  "detail": {
     "stack-set-arn": "arn:aws:cloudformation:us-east-1:111122223333:stackset/
test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59",
    "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
    "status-details": {
        "status": "OUTDATED",
        "status-reason": "User Initiated",
        "detailed-status": "PENDING"
    }
  }
}
```

StackSet Einzelheiten zum Ereignis „Betriebsstatus ändern“

Im Folgenden finden Sie die Detailfelder für Ereignisse zur Änderung des StackSet Betriebsstatus.

Die `detail-type` Felder `source` und sind enthalten, da sie spezifische Werte für AWS CloudFormation Ereignisse enthalten.

```
{
  . . .,
  "detail-type": "CloudFormation StackSet Operation Status Change",
  "source": "aws.cloudformation",
  . . .,
  "detail": {
    "stack-set-arn" : "string",
    "stack-set-operation-id" : "string",
    "status-details": {
        "status": "string"
      }
    }
  }
}
```

`detail-type`

Identifiziert den Ereignistyp.

Für Ereignisse im StackSet Betriebsstatus ist dieser Wert`CloudFormation StackSet Operation Status Change`.

`source`

Identifiziert den Service, aus dem das Ereignis stammt. Für CloudFormation Ereignisse ist dieser Wert`aws.cloudformation`.

`detail`

Ein JSON-Objekt, das Informationen zum Ereignis enthält. Der Service, der das Ereignis generiert, bestimmt den Inhalt dieses Feldes.

Für Ereignisse im Zusammenhang mit dem StackSet Betriebsstatus umfassen diese Daten:

`stack-set-arn`

Der Amazon-Ressourcenname (ARN), der dem zugeordnet ist StackSet.

`stack-set-operation-id`

Die eindeutige ID, die dem StackSet Vorgang zugeordnet ist.

`status-details`

`status`

Der Status StackSet des Vorgangs.

Weitere Details finden Sie unter StackSets Statuscodes.

Zulässige Werte: `RUNNING` | `SUCCEEDED` | `FAILED` | `STOPPING` | `STOPPED` | `QUEUED`

Example Beispiel: Ereignis „Änderung des StackSet Betriebsstatus“

Im Folgenden finden Sie ein Beispiel für ein Ereignis zur Änderung des StackSet Betriebsstatus. Die Ereignisdetails, die dazu CloudFormation führen, dass der angeforderte Vorgang für das angegebene Stack-Set erfolgreich abgeschlossen wurde.

```
{
  "version": "0",
  "id": "4de89905-fd92-6a6b-9509-23c04bcb6a21",
  "detail-type": "CloudFormation StackSet Operation Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-22T05:46:24Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-
d041-44ad-94f8-b8268aca1e59"
  ],
  "detail": {
    "stack-set-arn": "arn:aws:cloudformation:us-east-1:111122223333:stackset/
test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59",
    "stack-set-operation-id": "ce69adce-2221-4483-8c4b-c51f284f25e8",
    "status-details": {
        "status": "SUCCEEDED"
    }
  }
}
```

Details zum Ändern des Repository-Synchronisierungsstatus

Im Folgenden finden Sie die Detailfelder für Ereignisse zur Änderung des Status der Repository-Synchronisierung.

Die `detail-type` Felder `source` und sind enthalten, da sie spezifische Werte für AWS CloudFormation Ereignisse enthalten.

```
{
  . . .,
  "detail-type": "Git Sync Repository Sync Status Change",
  "source": "aws.codeconnections",
  . . .,
  "detail": {
    "connectionArn" : "string",
    "providerType" : "string",
    "repositoryName": "string",
    "providerType": "string",
    "repositoryName": "string",
    "repositoryArn": "string",
    "repositoryLinkId": "string",
    "ownerId": "string",
    "commit": "string",
    "branch": "string",
    "syncType": "string",
    "status": "string",
    "previousSync": "string"
    }
  }
}
```

`detail-type`

Identifiziert den Ereignistyp.

Für Statusereignisse der Repository-Synchronisierung ist dieser Wert`Git Sync Repository Sync Status Change`.

`source`

Identifiziert den Service, aus dem das Ereignis stammt. Für Git-Sync-Ereignisse ist dieser Wert`aws.codeconnections`.

`detail`

Ein JSON-Objekt, das Informationen zum Ereignis enthält. Der Service, der das Ereignis generiert, bestimmt den Inhalt dieses Feldes.

Für Ereignisse mit dem Repository-Synchronisierungsstatus beinhalten diese Daten:

`connectionArn`

Der Amazon-Ressourcenname (ARN), der mit verknüpft ist CodeConnections.

`providerType`

Der Git-Anbieter, mit dem eine Verbindung hergestellt wurde CloudFormation.

Zulässige Werte: `GitHub` | `GitHub Enterprise` | `GitLab` | `BitBucket`

`repositoryName`

Der Name des Git-Repositorys.

`repositoryArn`

Der mit dem Git-Repository verknüpfte ARN.

`repositoryLinkId`

Die eindeutige ID, die dem Repository-Link zugeordnet ist.

`ownerId`

Die eindeutige ID, die dem Repository-Besitzer zugeordnet ist.

`commit`

Die eindeutige ID, die dem Repository-Commit zugeordnet ist.

`branch`

Die eindeutige ID, die dem Repository-Zweig zugeordnet ist.

`syncType`

Die Art der Synchronisierung, die durchgeführt wird.

`status`

Der aktuelle Repository-Synchronisierungsstatus.

Zulässige Werte: `FAILED` | `INITIATED` | `IN_PROGRESS` | `SUCCEEDED`

`previousSync`

Der Synchronisierungsstatus vor dem aktuellen Status.

Zulässige Werte: `FAILED` | `INITIATED` | `IN_PROGRESS` | `SUCCEEDED`

Example Beispiel: Ereignis zur Änderung des Repository-Synchronisierungsstatus

Im Folgenden finden Sie ein Beispiel für ein Ereignis zur Änderung des Status der Repository-Synchronisierung. Die Ereignisdetails, die CodeConnections das Repository erfolgreich synchronisiert haben.

```
{
  "version": "0",
  "id": "1b5d8feb-agbv-4cf7-a9f1-bf3703467718",
  "detail-type": "GitSync Repository Sync Status Change",
  "source": "aws.codeconnections",
  "account": "111122223333",
  "time": "2023-12-22T18:43:48Z",
  "region": "us-east-1",
  "resources": ["arn:aws:aws:codestar-connections:us-east-1:111122223333:repository-
link/550e8400-e29b-41d4-a716-446655440000",],
  "detail": {
    "connectionArn": "arn:aws:codestar-connections:us-east-1:111122223333:connection/
sample-connection-id",
    "providerType": "GitHub",
    "repositoryName": "sample-repository-name",
    "repositoryArn": "arn:aws:aws:codestar-connections:us-
east-1:111122223333:repository-link/550e8400-e29b-41d4-a716-446655440000"
    "repositoryLinkId": "550e8400-e29b-41d4-a716-446655440000"
    "ownerId": "sample-owner-id",
    "commit": "sample-commit-id",
    "branch": "main",
    "syncType": "CFN_STACK_SYNC",
    "status": "SUCCEEDED",
    "previousStatus": "IN_PROGRESS",
  }
}
```

Details zum Ereignis „Status der Ressourcensynchronisierung ändern“

Im Folgenden finden Sie die Detailfelder für Ereignisse zur Änderung des Resource Sync-Status.

Die `detail-type` Felder `source` und sind enthalten, da sie spezifische Werte für AWS CloudFormation Ereignisse enthalten.

```
{
  . . .,
```

```
  "detail-type": "Git Sync Resource Sync Status Change",
  "source": "aws.codeconnections",
  . . .,
  "detail": {
    "providerType" : "string",
    "commit" : "string",
    "repositoryName": "string",
    "branch": "string",
    "syncType": "string",
    "syncTarget": "string",
    "status": "string",
    "previousSync": "string"
  }
}
```

`detail-type`

Identifiziert den Ereignistyp.

Für Statusereignisse der Repository-Synchronisierung ist dieser Wert`Git Sync Repository Sync Status Change`.

`source`

Identifiziert den Service, aus dem das Ereignis stammt. Für Git-Sync-Ereignisse ist dieser Wert`aws.codeconnections`.

`detail`

Ein JSON-Objekt, das Informationen zum Ereignis enthält. Der Service, der das Ereignis generiert, bestimmt den Inhalt dieses Feldes.

Für Ereignisse mit dem Status der Ressourcensynchronisierung umfassen diese Daten:

`providerType`

Der Git-Anbieter, mit dem eine Verbindung hergestellt wurde CloudFormation.

Zulässige Werte: `GitHub` | `GitHub Enterprise` | `GitLab` | `BitBucket`

`commit`

Die eindeutige ID, die dem Repository-Commit zugeordnet ist.

`repositoryName`

Der Name des Git-Repositorys.

`branch`

Die eindeutige ID, die dem Repository-Zweig zugeordnet ist.

`syncType`

Die Art der Synchronisierung, die durchgeführt wird.

`syncTarget`

Der Zielstapel für die Ressourcensynchronisierung.

`status`

Der aktuelle Repository-Synchronisierungsstatus.

Zulässige Werte: `FAILED` | `INITIATED` | `IN_PROGRESS` | `SUCCEEDED`

`previousSync`

Der Synchronisierungsstatus vor dem aktuellen Status.

Zulässige Werte: `FAILED` | `INITIATED` | `IN_PROGRESS` | `SUCCEEDED`

Example Beispiel: Ereignis zur Änderung des Status der Ressourcensynchronisierung

Im Folgenden finden Sie ein Beispiel für ein Ereignis zur Änderung des Status der Ressourcensynchronisierung. Die Details des Ereignisses, das die Ressource erfolgreich synchronisiert CodeConnections hat.

```
{
  "version": "0",
  "id": "1b5d8feb-agbv-4cf7-a9f1-bf3703467718",
  "detail-type": "Git Sync Resource Sync Status Change",
  "source": "aws.codeconnections",
  "account": "111122223333",
  "time": "2023-12-22T18:43:48Z",
  "region": "us-east-1",
  "resources": ["arn:aws:aws:cloudformation:us-east-1:111122223333:stack/
targetStack1"],
  "detail": {
    "providerType": "GitHub",
    "commit": "sample-commit-id",
    "repositoryName": "sample-repository-name",
    "branch": "main",
```

```
    "syncType": "CFN_STACK_SYNC",
    "syncTarget": "arn:aws:aws:cloudformation:us-east-1:111122223333:stack/
targetStack1",
    "status": "SUCCEEDED",
    "previousStatus": "IN_PROGRESS"
  }
}
```

CloudFormation Kontingente verstehen

Ihr AWS-Konto verfügt über CloudFormation Kontingente, die Sie möglicherweise kennen müssen, wenn Sie Vorlagen erstellen und Stapel erstellen. Durch das Verständnis dieser Kontingente können Sie Einschränkungsfehler vermeiden, bei denen Sie Ihre Vorlagen oder Stacks überarbeiten müssten.

Die folgende Tabelle zeigt die Kontingente CloudFormation .

Kontingente	Beschreibung	Wert	Optimierungsstrategie
cfn-signalDaten zum Wartezustand	Die maximale Datenmenge, die übertragen `cfn-signal` werden kann.	4,096 Bytes	Um eine größere Menge zu übergeben, senden Sie die Daten an einen Amazon S3 S3-Bucket und verwenden Sie dann, `cfn-signal` um die Amazon S3 S3-URL an diesen Bucket zu übergeben.
Benutzerdefinierte Ressourcenantwort	Die maximale Datenmenge, die ein benutzerdefinierter Ressourcenanbieter weitergeben kann.	4,096 Bytes	
Dynamische Verweise pro Vorlage	Die maximale Anzahl dynamischer Verweise, die in einer einzelnen CloudFormation Stack-Vorlage zulässig ist.	60 dynamische Verweise in einer Stack-Vorlage	
Hooks pro Konto	Die maximal zulässige Anzahl von Hooks pro Konto und Region.	100 Hooks	

Kontingente	Beschreibung	Wert	Optimierungsstrategie
Hooks pro Ressource	Die maximale Anzahl von Hooks, die pro Ressource konfiguriert werden können.	100 Hooks	
Größe der Hook-Konfiguration	Die maximale Menge an Konfigurationsdaten, die ein Hook speichern kann.	204,8 KB	
Mappings	Die maximale Anzahl von Zuordnungen, die Sie in Ihrer CloudFormation Vorlage deklarieren können.	200 Zuordnungen	Um mehr Zuordnungen anzugeben, teilen Sie Ihre Vorlage in mehrere Vorlagen auf, z. B. mit verschachtelten Stacks.
Zuordnungsattribute	Die maximale Anzahl von Zuordnungsattributen für jedes Mapping, die Sie in Ihrer CloudFormation Vorlage deklarieren können.	200 Attribute	Um mehr Zuordnungsattribute anzugeben, teilen Sie die Attribute in mehrere Zuordnungen auf.
Zuordnungsname und Zuordnungsattributname	Die maximale Größe jedes Mapping-Namens.	255 Zeichen	
Module	Die maximale Anzahl von Modulen, die Sie in der CloudFormation Registrierung registrieren können, pro Konto und Region.	100 Module	

Kontingente	Beschreibung	Wert	Optimierungsstrategie
Modulversionen	Die maximale Anzahl von Versionen, die Sie in der CloudFormation Registrierung für ein bestimmtes Modul registrieren können.	100 Versionen	Um neue Versionen zu registrieren, verwenden Sie dies zunächst, DeregisterType um die Registrierung von Versionen aufzuheben, die Sie nicht mehr verwenden.
Verschachtelte Stacks	Die maximale Anzahl von CloudFormation Ressourcen, die ein verschachtelter Stack pro Vorgang erstellen, aktualisieren oder löschen kann. Sie können beispielsweise eine geschachtelte Stack-Hierarchie mit insgesamt mehr als 2500 Ressourcen haben, aber Sie können nicht mehr als 2500 dieser Ressourcen in einer einzigen Bereitstellung erstellen, aktualisieren oder löschen.	2 500 Ressourcen	Teilen Sie die Stack-Hierarchie in verschiedene Stacks auf.

Kontingente	Beschreibung	Wert	Optimierungsstrategie
Ausgaben	Die maximale Anzahl von Ausgaben, die Sie in Ihrer CloudFormation Vorlage deklarieren können.	200 Ausgaben	
Ausgabenname	Die maximale Größe eines Ausgabena mens.	255 Zeichen	
Parameter	Die maximale Anzahl von Parameter n, die Sie in Ihrer CloudFormation Vorlage deklarieren können.	200 Parameter	Wenn Sie weitere Parameter angeben möchten, können Sie Zuordnungen oder Listen verwenden, um einem einzelnen Parameter mehrere Werte zuzuweisen.
Parametername	Die maximale Größe eines Parameter namens.	255 Zeichen	
Parameterwert	Die maximale Größe eines Parameter werts.	4,096 Bytes	Um einen größeren parametrisierten Wert zu verwenden, erstellen Sie mehrere Parameter und verwenden Sie dann die `Fn::Join` Funktion, um die mehreren Werte zu einem einzigen Wert zu verketten.

Kontingente	Beschreibung	Wert	Optimierungsstrategie
Private Ressourcen	Die maximale Anzahl an privaten Ressourcen, die Sie pro Konto und Region in der CloudFormation Registrierung registrieren können.	50 private Ressourcen	
Versionen von privaten Ressourcen	Die maximale Anzahl von Versionen, die Sie in der CloudFormation Registrierung für eine bestimmte private Ressource registrieren können.	50 private Ressourcen	Um neue Versionen zu registrieren, verwenden Sie diese Option zunächst, DeregisterTypeum die Registrierung von Versionen aufzuheben, die Sie nicht mehr verwenden.
Ressourcen	Die maximale Anzahl von Ressourcen, die Sie in Ihrer CloudFormation Vorlage deklarieren können.	500 Ressourcen	Um mehr Ressourcen anzugeben, teilen Sie Ihre Vorlage in mehrere Vorlagen auf, z. B. mit verschachtelten Stacks.
Ressourcen in gleichzeitigen Stack-Operationen	Die maximale Anzahl von Ressourcen, die Sie zu einem bestimmten Zeitpunkt an Stack-Vorgängen (Erstellungs-, Aktualisierungs- oder Löschvorgänge) in Ihrer Region beteiligen können.	Verwenden Sie die DescribeAccountLimitsAPI, um das aktuelle Limit für ein Konto in einer bestimmten Region zu ermitteln.	

Kontingente	Beschreibung	Wert	Optimierungsstrategie
Ressourcenname	Die maximale Größe eines Ressource nnamens.	255 Zeichen	
Stacks	Die maximale Anzahl von CloudForm ation Stacks, die Sie erstellen können.	2000 Stacks	Um mehr Stapel zu erstellen, löschen Sie Stapel, die Sie nicht benötigen, oder fordern Sie eine Erhöhung der maximalen Anzahl von Stacks in Ihrem an. AWS-Konto Weitere Informationen finden Sie unter AWS Servicekontingente im Allgemeine AWS-Referenz.
Stack name	Die maximale Größe eines Stack-Namens.	128 Zeichen	

Kontingente	Beschreibung	Wert	Optimierungsstrategie
StackSets	Die maximale Anzahl von CloudFormation Stack-Sets, die Sie in Ihrem Administr atorkonto erstellen können.	1 000 Stack-Sets	Um mehr Stack-Sets zu erstellen, löschen Sie Stack-Sets, die Sie nicht benötigen , oder fordern Sie eine Erhöhung der maximalen Anzahl von Stack-Sets in Ihrem an AWS-Konto. Weitere Informationen finden Sie unter AWS Servicekontingente im Allgemeine AWS-Referenz.
Stack-Instances	Die maximale Anzahl von Stack-Instances, die Sie pro Stack-Set erstellen können.	100 000 Stack-Ins tances pro Stack-Set	Um mehr Stack-Ins tances zu erstellen , löschen Sie Stack-Instances, die Sie nicht benötigen, oder fordern Sie eine Erhöhung der maximalen Anzahl von Stack-Instances in Ihrer an AWS-Konto. Weitere Informationen finden Sie unter AWS Servicekontingente im Allgemeine AWS-Referenz.

Kontingente	Beschreibung	Wert	Optimierungsstrategie
StackSets Instanzoperationen	Die maximale Anzahl von Stack-Instances für alle Stack-Sets, auf denen Sie in jeder Region gleichzeitig Operationen ausführen können, pro Administratorkonto.	10 000 Vorgänge	Diese Einschränkung gilt für alle Stack-Sets, die an einer Region beteiligt sind. Dazu gehören Stack-Instances, die von der Erstellung und Aktualisierung von Stack-Sets betroffen sind, sowie das direkte Erstellen, Aktualisieren oder Löschen von Stack-Instances.
StackSets Operationen in der Warteschlange	Die maximale Anzahl von Vorgängen in der Warteschlange für ein Stack-Set zu einem bestimmten Zeitpunkt.	10 000 Vorgänge	
Stapel, die mit einem S3-Objekt pro Stackset-Vorgang importiert wurden	Die maximale Anzahl von Stacks, die Sie mit einem S3-Objekt pro Stack-Set-Vorgang importieren können.	200 Stacks	
Stapel, die mit Inline-Stack-IDs pro Stack-Set-Vorgang importiert wurden	Die maximale Anzahl von Stacks, die Sie mit Inline-Stack IDs pro Stack-Set-Vorgang importieren können.	10 Stapel	

Kontingente	Beschreibung	Wert	Optimierungsstrategie
Vorlagentextgröße in einer Anforderung	Die maximale Größe eines Vorlagentexts, den Sie in einer `CreateStack` `UpdateStack` , oder `ValidateTemplate` -Anforderung übergeben können.	51,200 Bytes	Um einen größeren Vorlagentext zu verwenden, teilen Sie Ihre Vorlage in mehrere Vorlagen auf, z. B. mit verschachtelten Stacks. Oder laden Sie die Vorlage in einen Amazon S3-Bucket hoch.
Vorlagentextgröße in einem Amazon S3-Objekt	Die maximale Größe eines Vorlagentexts, den Sie einem Amazon S3 S3-Objekt für eine`CreateStack` ,`UpdateStack` , `ValidateTemplate` Anfrage mit einer Amazon S3 S3-Vorlagen-URL übergeben können.	1 MB	Um einen größeren Vorlagentext zu verwenden, teilen Sie Ihre Vorlage in mehrere Vorlagen auf, z. B. mit verschachtelten Stacks. Oder verwenden Sie die Minimierung, um die CloudFormation Vorlagengröße zu reduzieren.
Vorlagenbeschreibung	Die maximale Größe einer Vorlagenbeschreibung.	1,024 Bytes	
Versionen pro Hook	Die maximale Anzahl von Versionen, die Sie pro Hook erstellen können.	100 Versionen	

Verfügbarkeit von Funktionen

CloudFormation Möglicherweise sind nicht alle Funktionen von in jeder Region verfügbar. Weitere Informationen zu AWS-Regionen finden Sie in der Tabelle mit den Regionen der globalen Infrastruktur.

- In den folgenden Regionen sind Makros zurzeit nicht verfügbar:
  - Asien-Pazifik (Jakarta)
- Die Durchführung von blue/green ECS-Bereitstellungen CodeDeploy mithilfe von CloudFormation ist derzeit in den folgenden Regionen nicht verfügbar:
  - Afrika (Kapstadt)
  - Asien-Pazifik (Osaka)
  - Europa (Milan)

StackSets und Makros

StackSets unterstützt derzeit nicht das Erstellen oder Aktualisieren von Stacksets mit vom Dienst verwalteten Berechtigungen anhand von Vorlagen, die Makros enthalten. Dazu gehören Transformationen, bei denen es sich um Makros handelt, die von gehostet werden. CloudFormation Weitere Informationen über Makros finden Sie unter Vorlagenmakros.

Problembehebung CloudFormation

Bei der Verwendung können Probleme auftreten CloudFormation, wenn Sie CloudFormation Stacks erstellen, aktualisieren oder löschen. Die folgenden Abschnitte können zum Beheben von häufig auftretenden Problemen helfen.

Allgemeine Fragen zu finden CloudFormation Sie unter. AWS CloudFormation FAQs Sie können in der CloudFormation Community auch nach Antworten suchen und Fragen posten unter AWS re:Post.

Themen

- Anleitung zur Fehlerbehebung
- Behebung von Fehlern
- Den -Support kontaktieren

Anleitung zur Fehlerbehebung

Falls CloudFormation dein Stack nicht erstellt, aktualisiert oder gelöscht werden kann, kannst du dir Fehlermeldungen oder Logs ansehen, um mehr über das Problem zu erfahren. In den folgenden Aufgaben werden allgemeine Methoden zur Behebung eines CloudFormation Problems beschrieben. Weitere Informationen zu bestimmten Fehlern und Lösungen finden Sie im Abschnitt Behebung von Fehlern.

- Verwenden Sie die CloudFormation -Konsole, um den Status Ihres Stacks anzuzeigen. In der Konsole können Sie eine Liste von Stack-Ereignissen anzeigen, während Ihr Stack erstellt, aktualisiert oder gelöscht wird. Finden Sie das Fehlerereignis in dieser Liste und sehen dann den Statusgrund für dieses Ereignis an. Der Statusgrund kann eine Fehlermeldung von CloudFormation oder von einem bestimmten Dienst enthalten, die Ihnen bei der Behebung Ihres Problems helfen kann. Weitere Informationen zum Anzeigen von Stack-Ereignissen finden Sie unter CloudFormation Stack-Ereignisse anzeigen.
- Bei EC2 Problemen mit Amazon sehen Sie sich die Protokolle `cloud-init` und CFN an. Diese Protokolle werden auf der EC2 Amazon-Instance im `/var/log/` Verzeichnis veröffentlicht. Diese Protokolle erfassen Prozesse und Befehlsausgaben während CloudFormation der Einrichtung Ihrer Instance. Unter Windows finden EC2 Sie den Dienst konfigurieren unter`%ProgramFiles%\Amazon\EC2ConfigService`, EC2 Launch in`%ProgramData%\Amazon\EC2-Windows\Launch\Logs`, EC2 Launch v2 in `%ProgramData%\Amazon\EC2Launch\log` und cfn loggt sich ein. `C:\cfn\log`

Sie können Ihre CloudFormation Vorlage auch so konfigurieren, dass die Protokolle auf Amazon veröffentlicht werden CloudWatch, wodurch Protokolle in Amazon angezeigt werden, AWS Management Console sodass Sie keine Verbindung zu Ihrer EC2 Amazon-Instance herstellen müssen. Weitere Informationen finden Sie im Application Management-Blog unter CloudFormation Protokolle in der Konsole anzeigen.

Behebung von Fehlern

Wenn Sie in Ihrem CloudFormation Stack auf die folgenden Fehler stoßen, können Sie die folgenden Lösungen verwenden, um die Ursache der Probleme zu finden und sie zu beheben.

Themen

- Stack-Fehler löschen
- Abhängigkeitsfehler
- AWS Config und AWS Systems Manager Konflikte
- Fehler bei der Analyse der Parameter bei der Übergabe einer Liste
- IAM Unzureichende Berechtigungen
- Ungültiger Wert oder nicht unterstützte Ressourceneigenschaft
- Kontingent überschritten
- Verschachtelte Stacks stecken im UPDATE_COMPLETE_CLEANUP_IN_PROGRESS, UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS oder UPDATE_ROLLBACK_IN_PROGRESS fest.
- Keine Aktualisierungen zur Ausführung vorhanden
- Ressource konnte während eines Erstellens, Aktualisierens oder Löschens des Stack-Vorgangs nicht stabilisiert werden.
- Sicherheitsgruppe ist in VPC nicht vorhanden.
- Rollback-Aktualisierung fehlgeschlagen
- Die Wartebedingung hat nicht die erforderliche Anzahl von Signalen von einer EC2 Amazon-Instance empfangen
- Ressource wurde vom Stack entfernt, aber nicht gelöscht.

Stack-Fehler löschen

Um dieses Problem zu beheben, führen Sie die folgenden Schritte aus:

- Einige Ressourcen müssen leer sein, bevor sie gelöscht werden können. Sie müssen beispielsweise alle Objekte in einem Amazon S3 S3-Bucket oder alle Instances in einer EC2 Amazon-Sicherheitsgruppe löschen, bevor Sie den Bucket oder die Sicherheitsgruppe löschen können.
- Stellen Sie sicher, dass Sie die nötigen IAM-Berechtigungen zum Löschen von Ressourcen im Stack besitzen. Zusätzlich zu den CloudFormation Berechtigungen müssen Sie berechtigt sein, die zugrunde liegenden Dienste wie Amazon S3 oder Amazon zu nutzen EC2.
- Wenn sich Stacks im `DELETE_FAILED` Status befinden, weil eine Ressource nicht gelöscht werden CloudFormation konnte, führen Sie den Löschvorgang mit dem `RetainResources` Parameter erneut aus und geben Sie die Ressource an, die nicht gelöscht CloudFormation werden kann. CloudFormation löscht den Stapel, ohne die beibehaltene Ressource zu löschen. Die Beibehaltung von Ressourcen ist nützlich, wenn eine Ressource nicht gelöscht werden kann, z. B. ein S3-Bucket mit Objekten, die Sie behalten möchten, aber dennoch den Stack löschen möchten.

  Nachdem Sie den Stapel gelöscht haben, können Sie die beibehaltenen Ressourcen manuell löschen, indem Sie den zugehörigen AWS Dienst verwenden.

  Alternativ können Sie erwägen, die `FORCE_DELETE_STACK` Option mit dem `DeletionMode` Parameter zu verwenden. Weitere Informationen zum erzwungenen Löschen eines Stacks finden Sie unterLöschen Sie einen Stack aus der CloudFormation Konsole.
- Sie können keine Stacks löschen, deren Beendigungsschutz aktiviert ist. Wenn Sie versuchen, einen Stack mit aktiviertem Beendigungsschutz zu löschen, schlägt das Löschen fehl und der Stack – einschließlich seines Status – bleibt unverändert. Deaktivieren Sie den Beendigungsschutz für den Stack und wiederholen Sie dann das Löschen.

  Dies umfasst verschachtelte Stacks, für deren Root-Stacks der Beendigungsschutz aktiviert ist. Deaktivieren Sie den Beendigungsschutz des Root-Stacks und wiederholen Sie dann das Löschen. Es wird dringend empfohlen, verschachtelte Stacks nicht direkt, sondern nur im Rahmen des Löschens des Root-Stacks und aller zugehörigen Ressourcen zu löschen.

  Weitere Informationen finden Sie unter CloudFormation Stapel vor dem Löschen schützen.
- Für alle anderen Probleme können Sie AWS -Support, falls vorhanden, einen Support Fall erstellen. Siehe Den -Support kontaktieren.

Abhängigkeitsfehler

Um einen Abhängigkeitsfehler zu beheben, fügen Sie ein `DependsOn` Attribut zu den Ressourcen hinzu, die von anderen Ressourcen in Ihrer Vorlage abhängen. In einigen Fällen müssen Sie Abhängigkeiten explizit deklarieren, damit Sie Ressourcen in der richtigen Reihenfolge erstellen oder löschen CloudFormation können. Wenn Sie beispielsweise eine Elastic IP-Adresse und eine VPC mit einem Internet-Gateway im selben Stack erstellen, muss die Elastic IP-Adresse vom Anhang des Internet-Gateways abhängen. Weitere Informationen finden Sie unter DependsOn-Attribut.

AWS Config und AWS Systems Manager Konflikte

AWS Config und AWS Systems Manager kann Infrastrukturverwaltungsaufgaben automatisieren, die zu Konflikten bei der Bereitstellung eines CloudFormation Stacks führen können. Gehen Sie wie folgt vor, um mögliche Konflikte zu vermeiden:

- Überprüfen Sie die Konfiguration von AWS Config und Systems Manager im zugehörigen AWS-Konto und AWS-Region.
- Suchen Sie nach aktiven Regeln oder Automatisierungsdokumenten, die während einer CloudFormation Bereitstellung ausgelöst werden könnten. Diese können möglicherweise zu Konflikten oder Ressourcenabhängigkeiten führen, die mit Ihrer Bereitstellung in Konflikt stehen.
- Suchen Sie in Ihrer CloudFormation Vorlage nach Ressourcen, die von AWS Config und Systems Manager verwaltet werden. Prüfen Sie, ob es zu Überschneidungen oder Interdependenzen kommt, und ziehen Sie in Erwägung, die Vorlagen- oder Automatisierungskonfiguration anzupassen, um Konflikte zu vermeiden.
- Deaktivieren oder unterbrechen Sie vorübergehend alle zugehörigen AWS Config Regeln oder Systems Manager Manager-Automatisierungen während der CloudFormation Bereitstellung. Denken Sie daran, die ursprünglichen Konfigurationen nach der erfolgreichen Bereitstellung wiederherzustellen, um das gewünschte Maß an Automatisierung und Konformität aufrechtzuerhalten.
- Suchen Sie in den CloudFormation Protokollen und Fehlermeldungen nach Hinweisen auf AWS Config Probleme im Zusammenhang mit Systems Manager, um die Ursache des Konflikts zu ermitteln.

Weitere Informationen zu AWS Config Regeln finden Sie unter Ressourcen auswerten mit. AWS-Config-Regeln

Weitere Informationen zu Systems Manager Manager-Automatisierungen finden Sie unter AWS Systems Manager Automatisierung.

Fehler bei der Analyse der Parameter bei der Übergabe einer Liste

Wenn Sie das AWS CLI oder verwenden CloudFormation , um eine Liste zu übergeben, fügen Sie vor jedem Komma das Escape-Zeichen (\) hinzu. Das folgende Beispiel zeigt, wie Sie einen Eingabeparameter angeben, wenn Sie die AWS CLI verwenden.

```
ParameterKey=CIDR,ParameterValue='10.10.0.0/16\,10.10.0.0/24\,10.10.1.0/24'
```

IAM Unzureichende Berechtigungen

Wenn Sie mit einem CloudFormation Stack arbeiten, benötigen Sie nicht nur Nutzungsberechtigungen CloudFormation, sondern Sie müssen auch berechtigt sein, die zugrunde liegenden Dienste zu verwenden, die in Ihrer Vorlage beschrieben sind. Wenn Sie beispielsweise einen Amazon S3-Bucket erstellen oder eine EC2 Amazon-Instance starten, benötigen Sie Berechtigungen für Amazon S3 oder Amazon EC2. Überprüfen Sie die IAM-Richtlinie und stellen Sie sicher, dass Sie über die erforderlichen Berechtigungen verfügen, bevor Sie mit CloudFormation - Stacks arbeiten. Weitere Informationen finden Sie unter Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management.

Ungültiger Wert oder nicht unterstützte Ressourceneigenschaft

Wenn Sie einen CloudFormation Stack erstellen oder aktualisieren, kann Ihr Stack aufgrund ungültiger Eingabeparameter, nicht unterstützter Ressourceneigenschaftsnamen oder nicht unterstützter Ressourceneigenschaftswerte fehlschlagen. Stellen Sie für Eingabe-Parameter sicher, dass die Ressource vorhanden ist. Wenn Sie beispielsweise ein EC2 Amazon-Schlüsselpaar oder eine VPC-ID angeben, muss die Ressource in Ihrem Konto und in der Region vorhanden sein, in der Sie Ihren Stack erstellen oder aktualisieren. Sie können AWS-spezifische Parametertypen verwenden, um sicherzustellen, dass Sie gültige Werte verwenden.

Für die Namen und Werte von Ressourceneigenschaften aktualisieren Sie Ihre Vorlage, um gültige Namen und Werte zu verwenden. Weitere Informationen finden Sie in der Referenz zu AWS Ressourcen- und Eigenschaftstypen.

Kontingent überschritten

Vergewissern Sie sich, dass Sie nicht den Ressourcenkontingent erreicht haben. Die standardmäßige maximale Anzahl von Amazon EC2 On-Demand-Instances, die Sie starten können, beträgt beispielsweise 5. Wenn Sie versuchen, mehr Amazon EC2 On-Demand-Instances als Ihr Kontokontingent zu erstellen, schlägt die Instance-Erstellung fehl und Sie erhalten die Fehlermeldung`Status=start_failed`. Informationen zum Anzeigen der AWS -Standardkontingente nach Service, finden Sie unter AWS -Service Quotas in Allgemeine AWS-Referenz.

Informationen zu CloudFormation Kontingenten und Optimierungsstrategien finden Sie unter. CloudFormation Kontingente verstehen

Außerdem wird bei einem Update, wenn eine Ressource ersetzt wird, eine neue Ressource CloudFormation erstellt, bevor die alte gelöscht wird. Dieser Ersatz könnte Ihr Konto über den Ressourcenkontingent setzen. Dies würde dazu führen, dass die Aktualisierung fehlschlägt. Sie können überschüssige Ressourcen löschen oder eine Kontingent-Erhöhung beantragen.

Verschachtelte Stacks stecken im **`UPDATE_COMPLETE_CLEANUP_IN_PROGRESS`**, **`UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS`** oder **`UPDATE_ROLLBACK_IN_PROGRESS`** fest.

Ein verschachtelter Stack konnte nicht zurückgesetzt werden. Aufgrund potenzieller Ressourcenabhängigkeiten zwischen verschachtelten Stacks beginnt CloudFormation das Bereinigen der verschachtelten Stack-Ressourcen erst, wenn alle verschachtelten Stacks aktualisiert oder ein Rollback durchgeführt wurde. Wenn ein Rollback für einen verschachtelten Stapel fehlschlägt, werden alle Operationen CloudFormation abgebrochen, unabhängig davon, in welchem Status sich die anderen verschachtelten Stacks befinden. Ein verschachtelter Stack, der die Aktualisierung oder das Rollback abgeschlossen hat, aber kein Signal erhalten hat, dass mit dem Aufräumen begonnen werden CloudFormation soll, weil ein anderer verschachtelter Stack das Rollback nicht durchführen konnte, befindet sich im Status Oder. `UPDATE_COMPLETE_CLEANUP_IN_PROGRESS` `UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS` Ein verschachtelter Stack, der nicht aktualisieren konnte, aber kein Signal erhalten hat, um den Rollback zu starten, befindet sich in einem `UPDATE_ROLLBACK_IN_PROGRESS`-Status.

Ein verschachtelter Stack kann aufgrund von Änderungen, die außerhalb von vorgenommen wurden, möglicherweise nicht rückgängig gemacht werden CloudFormation, wenn die Stapelvorlage den

Status des Stacks nicht genau wiedergibt. Ein verschachtelter Stack kann auch fehlschlagen, wenn eine Auto-Scaling-Gruppe in einem verschachtelten Stack keine ausreichende Timeout-Zeit für das Ressourcen-Signal hatte, als die Gruppe erstellt oder aktualisiert wurde.

Um den Stapel zu reparieren, kontaktieren Sie AWS -Support.

Keine Aktualisierungen zur Ausführung vorhanden

Um einen CloudFormation Stack zu aktualisieren, müssen Sie Änderungen an den Vorlagen- oder Parameterwerten an CloudFormation senden. Einige Vorlagenänderungen werden jedoch CloudFormation nicht als Aktualisierung erkannt, z. B. Änderungen an einer Löschrichtlinie, einer Aktualisierungsrichtlinie, einer Bedingungsdeklaration oder einer Ausgabedeklaration. Wenn Sie Änderungen vornehmen müssen, ohne etwas anderes zu ändern, können Sie ein Metadaten Attribut für alle Ressourcen hinzufügen oder ändern. Das Metadatenattribut kann ein beliebiger Wert sein, da CloudFormation es seinen Inhalt nicht interpretiert. Weitere Informationen finden Sie unter Metadata-Attribut.

Ressource konnte während eines Erstellens, Aktualisierens oder Löschens des Stack-Vorgangs nicht stabilisiert werden.

Eine Ressource hat nicht reagiert, weil der Vorgang den CloudFormation Timeout-Zeitraum überschritten hat oder ein AWS Dienst unterbrochen wurde. Überprüfen Sie bei Dienstunterbrechungen, ob der entsprechende AWS Dienst ausgeführt wird, und wiederholen Sie dann den Stack-Vorgang.

Wenn die AWS Dienste erfolgreich ausgeführt wurden, überprüfen Sie, ob Ihr Stack eine der folgenden Ressourcen enthält:

- `AWS::AutoScaling::AutoScalingGroup` zum Erstellen, Aktualisieren und Löschen von Vorgängen
- `AWS::CertificateManager::Certificate` zum Erstellen von Vorgängen
- `AWS::CloudFormation::Stack` zum Erstellen, Aktualisieren und Löschen von Vorgängen
- `AWS::ElasticSearch::Domain` zum Aktualisieren von Vorgängen
- `AWS::RDS::DBCluster` zum Erstellen und Aktualisieren von Vorgängen
- `AWS::RDS::DBInstance` zum Erstellen, Aktualisieren und Löschen von Vorgängen
- `AWS::Redshift::Cluster` zum Aktualisieren von Vorgängen

Vorgänge für diese Ressourcen können länger dauern als die Standard-Timeout-Zeit. Die Timeout-Zeit hängt von der Ressource und den Anmeldeinformationen ab, die Sie tatsächlich nutzen. Geben Sie eine Service-Rolle, an, wenn Sie den Stack-Vorgang ausführen, um die Timeout-Zeit zu verlängern. Wenn Sie bereits eine Service-Rolle verwenden oder Ihr Stack eine Ressource enthält, die nicht aufgeführt ist, wenden Sie sich bitte an den AWS -Support.

Wenn sich Ihr Stack im `UPDATE_ROLLBACK_FAILED` Status befindet, sehen Sie nach unter Update-Rollback fehlgeschlagen.

Sicherheitsgruppe ist in VPC nicht vorhanden.

Überprüfen Sie, ob die Sicherheitsgruppe in der VPC vorhanden ist, die Sie angegeben haben. Wenn die Sicherheitsgruppe vorhanden ist, stellen Sie sicher, dass Sie die ID der Sicherheitsgruppe angeben und nicht den Namen. Wenn beispielsweise die `AWS::EC2::SecurityGroupIngress` Ressource einen `SourceSecurityGroupName` und `SourceSecurityGroupId` Eigenschaften hat. Für VPC-Sicherheitsgruppen müssen Sie die `SourceSecurityGroupId` Eigenschaft verwenden und die Sicherheitsgruppen-ID angeben.

Rollback-Aktualisierung fehlgeschlagen

Eine abhängige Ressource kann nicht in seinen ursprünglichen Zustand zurückversetzt werden, wodurch das Rollback fehlschlägt (`UPDATE_ROLLBACK_FAILED`-Status). Beispielsweise kann es einen Stack geben, der ein Rollback auf eine alte Datenbank-Instance ausführt, die außerhalb von CloudFormation gelöscht wurde. Da sie CloudFormation nicht weiß, dass die Datenbank gelöscht wurde, geht sie davon aus, dass die Datenbankinstanz noch existiert, und versucht, zu ihr zurückzukehren, wodurch das Update-Rollback fehlschlägt.

Je nach der Ursache des Fehlers können Sie ihn manuell beheben und das Rollback fortfahren. Wenn Sie mit dem Rollback fortfahren, können Sie Ihren Stack wieder in den Arbeitszustand bringen (den `UPDATE_ROLLBACK_COMPLETE` Status) und dann erneut versuchen, den Stack zu aktualisieren. Die folgende Liste beschreibt Lösungen für allgemeine Fehler, die fehlerhafte Rollback-Aktualisierungen verursachen:

- Fehler beim Abrufen der erforderlichen Anzahl von Signalen

  Verwenden Sie den signal-resource-Befehl, um die erforderliche Anzahl erfolgreicher Signale manuell zur Ressource zu senden, die auf sie wartet, und setzen Sie dann das Rollback der Aktualisierung fort. Zum Beispiel könnten Instances in einer Auto-Scaling-Gruppe während eines Aktualisierungs-Rollbacks scheitern, Erfolg innerhalb der Timeout-Dauer zu signalisieren.

Erfolgssignale zur Auto-Scaling-Gruppe manuell senden. Wenn Sie mit dem Update-Rollback fortfahren, werden Ihre CloudFormation Signale angezeigt und der Rollback wird fortgesetzt.

- Änderungen an einer Ressource wurden außerhalb von vorgenommen CloudFormation

  Synchronisieren Sie Ressourcen manuell, damit sie mit der ursprüngliche Stack-Vorlage übereinstimmen und setzen dann mit dem Rollback der Aktualisierung fort. Wenn Sie beispielsweise manuell eine Ressource gelöscht haben, CloudFormation zu der ein Rollback ausgeführt werden soll, müssen Sie diese Ressource manuell mit demselben Namen und denselben Eigenschaften wie im ursprünglichen Stapel erstellen.

- Unzureichende Berechtigungen

  Stellen Sie sicher, dass Sie über ausreichende IAM-Berechtigungen zum Ändern der Ressourcen besitzen, und setzen dann das Aktualisierungs-Rollback fort. Zum Beispiel könnte Ihnen Ihre IAM-Richtlinie erlauben, einen S3-Bucket zu erstellen, aber nicht zu ändern. Fügen Sie die Änderungsaktionen Ihrer Richtlinie hinzu.

- Ungültiger Sicherheits-Token

  CloudFormation erfordert einen neuen Satz von Anmeldeinformationen. Es ist keine Änderung erforderlich. Mit dem Rollback der Aktualisierung fortsetzen, was die Anmeldeinformationen aktualisiert.

- Beschränkungsfehler

  Löschen Sie Ressourcen, die Sie nicht benötigen oder fordern sie eine Kontingent-Erhöhung an, und setzen dann mit dem Rollback der Aktualisierung fort. Wenn Ihr Kontingent für die Anzahl der EC2 On-Demand-Instances beispielsweise 5 beträgt und das Update-Rollback dieses Kontingent überschreitet, schlägt es fehl.

- Die Ressource hat sich nicht stabilisiert

  Eine Ressource hat nicht reagiert, weil der Vorgang möglicherweise den CloudFormation Timeout-Zeitraum überschritten hat oder ein AWS Dienst unterbrochen wurde. Es ist keine Änderung erforderlich. Wenn der Ressourcenvorgang abgeschlossen ist oder der AWS Dienst wieder in Betrieb ist, setzen Sie das Rollback des Updates fort.

Um das Rollback eines Updates fortzusetzen, können Sie die CloudFormation Konsole oder die AWS Befehlszeilenschnittstelle (AWS CLI) verwenden. Weitere Informationen finden Sie unter Rollback einer Aktualisierung fortsetzen.

Wenn keine dieser Lösungen funktioniert, können Sie die Ressourcen überspringen, für die ein Rollback nicht erfolgreich durchgeführt werden konnte. CloudFormation Weitere Informationen finden Sie unter dem `ResourcesToSkip` Parameter für den ContinueUpdateRollbackAPI-Vorgang in der AWS CloudFormation API-Referenz. CloudFormation setzt den Status der angegebenen Ressourcen auf `UPDATE_COMPLETE` und setzt das Rollback des Stacks fort. Nachdem der Rollback-Vorgang abgeschlossen ist, entspricht der Status der übersprungenen Ressourcen nicht dem Status der Ressourcen in der Stack-Vorlage. Bevor Sie eine weitere Stack-Aktualisierung durchführen, müssen Sie die Ressourcen ändern oder den Stack aktualisieren, sodass ihr Status konsistent ist. Andernfalls schlagen nachfolgende Aktualisierungen fehl, und der Stack lässt sich nicht mehr wiederherstellen.

Die Wartebedingung hat nicht die erforderliche Anzahl von Signalen von einer EC2 Amazon-Instance empfangen

Um dieses Problem zu beheben, führen Sie die folgenden Schritte aus:

- Stellen Sie sicher, dass auf dem von Ihnen verwendeten AMI die CloudFormation Hilfsskripte installiert sind. Wenn das AMI die Hilfeskripts nicht hat, können Sie diese auch auf Ihre Instance runterladen. Weitere Informationen finden Sie in der Referenz zu CloudFormation Hilfsskripten im AWS CloudFormation Template Reference Guide.
- Überprüfen Sie, ob der `cfn-signal` Befehl erfolgreich in der Instance ausgeführt wurde. Sie können z. B. Protokolle wie `/var/log/cloud-init.log` oder `/var/log/cfn-init.log` anzeigen, die Ihnen beim Debuggen des Instance-Starts helfen. Sie können die Protokolle abrufen, indem Sie sich bei Ihrer Instance anmelden. Sie müssen jedoch den Rollback bei einem Fehler deaktivieren oder die Instance CloudFormation löschen, wenn Ihr Stack nicht erstellt werden kann. Sie können die Protokolle auch auf Amazon veröffentlichen CloudWatch. Unter Windows können Sie sich die CFN-Logs `C:\cfn\log` und EC2 Config Service Logins `%ProgramFiles%\Amazon\EC2ConfigService` ansehen.
- Überprüfen Sie, ob die Instance eine Verbindung zum Internet hat. Wenn sich die Instance in einer VPC befindet, sollte die Instance in der Lage sein, eine Verbindung mit dem Internet über ein NAT-Gerät herzustellen, wenn es in einem privaten Subnetz ist, oder über ein Internet-Gateway, wenn es sich in einem öffentlichen Subnetz befindet. Versuchen Sie zum Testen der Instance-Internetverbindung auf eine öffentliche Webseite zuzugreifen, z. B. `http://aws.amazon.com`. Zum Beispiel können Sie den folgenden Befehl auf der Instance ausführen. Es sollte einen HTTP 200 Statuscode zurückgeben.

```
curl -I https://aws.amazon.com
```

Weitere Informationen zum Konfigurieren eines NAT-Geräts finden Sie unter NAT im Amazon VPC Benutzerhandbuch.

Ressource wurde vom Stack entfernt, aber nicht gelöscht.

CloudFormation Hat während eines Stack-Updates eine Ressource aus einem Stack entfernt, die Ressource jedoch nicht gelöscht. Die Ressource existiert noch, ist aber nicht mehr über CloudFormation zugänglich. Dies kann während Stack-Updates auftreten, bei denen:

- CloudFormation muss eine bestehende Ressource ersetzen, also erstellt es zuerst eine neue Ressource und versucht dann, die alte Ressource zu löschen.
- Sie die Ressource aus der Stack-Vorlage entfernt haben, daher versucht CloudFormation, die Ressource aus dem Stack zu löschen.

Es kann jedoch Fälle geben, in denen die Ressource nicht gelöscht werden CloudFormation kann. Zum Beispiel, wenn der Benutzer keine Berechtigung hat, eine Ressource eines bestimmten Typs zu löschen.

CloudFormation versucht dreimal, die alte Ressource zu löschen. Wenn die alte Ressource nicht gelöscht werden CloudFormation kann, entfernt es die alte Ressource aus dem Stapel und setzt die Aktualisierung des Stacks fort. Wenn das Stack-Update abgeschlossen ist, wird CloudFormation ein `UPDATE_COMPLETE` Stack-Ereignis ausgelöst, `StatusReason` das jedoch ein Ereignis enthält, das besagt, dass eine oder mehrere Ressourcen nicht gelöscht werden konnten. CloudFormation löst außerdem ein `DELETE_FAILED` Ereignis für die spezifische Ressource aus, in dem ein entsprechendes `StatusReason` Ereignis ausführlicher beschrieben wird, warum die Ressource nicht gelöscht CloudFormation werden konnte.

Um diese Situation zu beheben, löschen Sie die Ressource direkt über die Konsole oder API für den zugrunde liegenden Service.

Den -Support kontaktieren

Falls ja AWS -Support, können Sie zu https://console.aws.amazon.com/support/Hause eine Anfrage für den technischen Support erstellen#/. Sammeln Sie die folgenden Informationen, bevor Sie sich an den Support wenden:

- Die ID des Stacks. Sie finden die Stack-ID auf der Registerkarte Overview der -CloudFormation Konsole. Weitere Informationen finden Sie unter Stack-Informationen von der CloudFormation Konsole aus anzeigen.

> **Important**
>
> Nehmen Sie außerhalb von keine Änderungen an den Ressourcen im Stack vor. CloudFormation Wenn Sie außerhalb von Änderungen an den Ressourcen in Ihrem Stack vornehmen, wird Ihr Stack CloudFormation möglicherweise in einen Zustand versetzt, der nicht wiederhergestellt werden kann.

- Jede Stack-Fehlermeldungen. Weitere Informationen zum Anzeigen von Stack-Fehlermeldungen finden Sie im Anleitung zur Fehlerbehebung Abschnitt.
- Sammeln Sie bei EC2 Amazon-Problemen die Protokolle `cloud-init` und CFN. Diese Protokolle werden auf der EC2 Amazon-Instance im `/var/log/` Verzeichnis veröffentlicht. Diese Protokolle erfassen Prozesse und Befehlsausgaben während Ihre Instance eingerichtet wird. Für Windows erfassen EC2 Sie den Dienst Configure und cfn loggt sich ein `%ProgramFiles%\Amazon\EC2ConfigService` und`C:\cfn\log`.

Sie können in der CloudFormation Community auch nach Antworten suchen und Fragen stellen unter AWS re:Post.

Fehlerbehebung bei erfolglosen CloudFormation Stack-Bereitstellungen mit Amazon Q Developer

Amazon Q Developer ist ein auf generativer künstlicher Intelligenz (KI) basierender Konversationsassistent, der Ihnen helfen kann, AWS Anwendungen zu verstehen, zu erstellen, zu erweitern und zu bedienen. Informationen zu Amazon Q Developer finden Sie im Amazon Q Developer User Guide.

In diesem Dokument wird beschrieben, wie Sie Amazon Q Developer verwenden, um Ihnen bei der Behebung fehlgeschlagener CloudFormation Bereitstellungen zu helfen. Diese Funktion soll Ihnen helfen, Probleme, die bei der Stack-Bereitstellung auftreten, schnell zu identifizieren und zu lösen.

Bereitgestellt von Amazon Bedrock: Amazon Q basiert auf Amazon Bedrock und umfasst eine automatische Missbrauchserkennung, die in Amazon Bedrock implementiert ist, um Sicherheit und den verantwortungsvollen Umgang mit KI zu gewährleisten.

Themen

- Features
- Funktionsweise
- Preisgestaltung
- Regionale Verfügbarkeit
- Einschränkungen

Features

Die Problembehebung von Stack-Bereitstellungen mit der Amazon Q Developer-Lösung bietet die folgenden Funktionen:

- Fehleranalyse — Bietet eine easy-to-understand Analyse von Fehlern bei der CloudFormation Stack-Bereitstellung.
- Lösungsvorschläge — Bietet step-by-step Anweisungen zur Lösung identifizierter Probleme.

Funktionsweise

Wenn die Erstellung oder Aktualisierung eines CloudFormation Stacks fehlschlägt, wird eine Fehlermeldung im angezeigt AWS Management Console.

Sie können die Schaltfläche Mit Q diagnostizieren neben der Fehlermeldung auswählen.

Amazon Q Developer analysiert den Fehler und gibt eine Erklärung des Problems.

Wenn Sie weitere Unterstützung benötigen, wählen Sie „Hilfe bei der Lösung“, um detaillierte Lösungsschritte zu erhalten.

Preisgestaltung

Preisinformationen finden Sie unter Amazon Q Developer-Preise.

Regionale Verfügbarkeit

Amazon Q Developer ist in den meisten Versionen verfügbar AWS-Regionen. Eine Liste aller Regionen, in denen Amazon Q Developer derzeit verfügbar ist, finden Sie unter Unterstützte Regionen für Amazon Q Developer im Amazon Q Developer User Guide.

Einschränkungen

- Auf diese Funktion kann nur über die zugegriffen AWS Management Console werden.
- Diese Funktion deckt die 20 häufigsten Fehler ab. Es deckt nicht alle möglichen Fehler ab.
- Bei komplexen Problembehebungsfällen ist möglicherweise dennoch ein Support Eingreifen erforderlich.

Dokumentenverlauf für das AWS CloudFormation Benutzerhandbuch

In der folgenden Tabelle werden wichtige Änderungen am Inhalt des AWS CloudFormation Benutzerhandbuchs nach Mai 2018 beschrieben. Um Benachrichtigungen über Aktualisierungen der Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

> **Important**
>
> Die Tabellenzeilen, in denen Aktualisierungen des Inhalts der Vorlagenreferenzen ab Mai 2018 beschrieben werden, wurden in die neue AWS CloudFormation Vorlagenreferenz verschoben. Diese Aktualisierungen und alle future Änderungen finden Sie im Dokumentverlauf für die AWS CloudFormation Vorlagenreferenz in der AWS CloudFormation Vorlagenreferenz.

Änderung	Beschreibung	Datum
Leitfaden im PDF-Format verfügbar	Sie können das AWS CloudFormation Benutzerh andbuch jetzt als PDF herunterladen.	30. Mai 2025
Der Inhalt der Vorlagenr eferenz wurde in ein neues Handbuch verschoben	CloudFormation hat den AWS CloudFormation Vorlagen-Referenzleitfaden veröffent licht. Einzelheiten finden Sie im Referenzhandbuch für AWS CloudFormation Vorlagen.	30. Mai 2025
Der IaC-Generator unterstützt teilweises Scannen	Sie können jetzt bestimmte Ressourcentypen auswählen , nach denen gesucht werden soll, was das Generieren von Infrastructure-as-Code	27. März 2025

	(IaC) -Vorlagen aus Ihren vorhandenen Ressourcen erleichtert. Weitere Informationen finden Sie unter Starten eines Ressourcenscans mit dem CloudFormation IaC-Generator.	
Refactoring von Stapeln	Das Stack-Refactoring vereinfacht die Reorganisation der Ressourcen in Ihren CloudFormation Stacks und behält gleichzeitig die vorhandenen Ressourceneigenschaften und -daten bei. Weitere Informationen finden Sie unter Stack-Refactoring.	6. Februar 2025
Fehlerbehebung bei Stack-Bereitstellungen mit Amazon Q Developer	Sie können jetzt Amazon Q Developer verwenden, um häufig auftretende Fehler bei der Bereitstellung von CloudFormation Stacks zu beheben. Weitere Informationen finden Sie unter Problembehandlung bei erfolglosen CloudFormation Stack-Bereitstellungen mit Amazon Q Developer.	22. November 2024

Diagramm zum Zeitplan der Stack-Bereitstellung	Sie können jetzt eine visuelle Darstellung Ihrer Stack-Bereitstellung sehen. Das Zeitdiagramm für die Stack-Bereitstellung zeigt den Status der Stack-Bereitstellung, den Bereitstellungsstatus einzelner Ressourcen und die Zeiten, zu denen sich der Bereitste llungsstatus geändert hat. Weitere Informationen finden Sie unter Einen Zeitplan für eine CloudFormation Stack-Bereitstellung anzeigen.	11. November 2024
Visualisieren Sie Ihre gescannten Ressourcen und generierten Vorlagen	Sie können jetzt Ihre Workflows zur Generierung von Infrastructure as Code (IaC) optimieren, indem Sie die Details der Scanzusam menfassung visualisieren und eine Vorschau der generiert en Vorlagen anzeigen, bevor Sie Ihren Infrastruktur-Stack bereitstellen. Weitere Informati onen finden Sie unter Scan-Zusammenfassung in der CloudFormation Konsole anzeigen und CloudForm ation Stapel aus gescannten Ressourcen erstellen.	22. August 2024

EventBridge Amazon-Integration mit AWS CloudFormation Git-Sync	AWS CloudFormation Git Sync veröffentlicht jetzt Änderungen des Synchronisierungsstatus als Ereignisse an Amazon EventBridge. Weitere Informationen finden Sie unter Ereignisdetails zur Änderung des Repository-Synchronisierungsstatus und Ereignisdetails zur Änderung des Ressourcensynchronisierungsstatus.	29. Juli 2024
Erzwingen Sie das Löschen festgefahrener Stapel	Für blockierte Stack-Löschvorgänge sind zwei neue Optionen verfügbar, um das Löschen von Stacks zu erzwingen. Sie können jetzt wählen, ob Sie das Löschen des Stacks erzwingen, aber die Ressource beibehalten, oder das Löschen des gesamten Stacks erzwingen möchten. Weitere Informationen finden Sie unter Löschen eines Stacks aus der CloudFormation Konsole.	22. Mai 2024

AWS CloudTrail Die Hauptursachen des Vorgangs im Ereignisstapel	CloudFormation verbessert die Problembehebung bei Stack-Vorgängen mit einer neuen AWS CloudTrail Deep-Link-Integration. Diese Funktion verknüpft Stack-Betriebsereignisse in der CloudFormation Konsole direkt mit relevanten CloudTrail Ereignissen. Weitere Informationen finden Sie unter Ermitteln der Ursache eines Stack-Fehlers.	15. Mai 2024
Sätze zur Änderung der Eigenschaftsebene	Mithilfe von Änderungssätzen auf Eigenschaftsebene können Sie eine Vorschau der Änderungen anzeigen, die CloudFormation Bereitstellungen an den Eigenschaftswerten von Ressourcen vornehmen werden. Weitere Informationen finden Sie unter Einen Änderungssatz für einen CloudFormation Stack anzeigen.	12. April 2024

CloudFormation stellt die `CONFIGURATION_COMPLETE` Veranstaltung vor	Sie können das `CONFIGURATION_COMPLETE` Ereignis jetzt verwenden, um schnellere Workflows zu ermöglichen, bei denen Ressourcen erstellt werden. Weitere Informationen finden Sie unter Grundlegendes zu Ereignissen bei der CloudFormation Stackerstellung.	11. März 2024
Generieren Sie AWS CloudFormation Vorlagen und AWS CDK Anwendungen aus vorhandenen AWS Ressourcen	Sie können jetzt eine Vorlage mithilfe von Ressourcen generieren, die in Ihrem Konto bereitgestellt wurden und noch nicht von CloudFormation verwaltet werden. Weitere Informationen finden Sie unter Generieren von Vorlagen aus vorhandenen Ressourcen mit dem IaC-Generator.	2. Februar 2024
StackSets Parallelitätsmodus	Der Parallelitätsmodus ist ein Parameter, mit dem Sie auswählen können`StackSetOperationPreferences`, wie sich die Parallelitätsebene bei Stack-Set-Vorgängen verhält. Weitere Informationen finden Sie unter Wählen Sie den Parallelitätsmodus für. CloudFormation StackSets	9. November 2023

Detaillierte StackSet Drift-Informationen	Im Folgenden APIs können Sie sehen, welche Stack-Instances von der StackSet Vorlage abgewichen sind und welche Ressourcen sich geändert haben. ListStackInstanceResourceDrifts Gibt Abweichungsinformationen für Ressourcen in einer Stack-Instance zurück. StackInstanceResourceDriftsSummary Die Struktur, die zusammenfassende Informationen über Ressourcenabweichungen für eine Stack-Instance enthält.	24. Juli 2023

CloudFormation StackSets APIs um den vertrauen swürdigen Zugriff zu kontrolli eren AWS Organizations	CloudFormation StackSets bietet Kunden Folgendes APIs für die Verwaltung des AWS Organizations Vertrauen szugriffs: ActivateOrganizationsAccess Aktivieren Sie den vertrauenswürdigen Zugriff mit AWS Organizations. Wenn der vertrauen swürdige Zugriff zwischen StackSets und Organizat ions aktiviert ist, verfügt das Verwaltungskonto über Berechtigungen zum Erstellen und Verwalten StackSets für Ihre Organisation. DeactivateOrganizationsAcce ss Deaktiviert den vertrauen swürdigen Zugriff mit AWS Organizations. Wenn der vertrauenswürdige Zugriff deaktiviert ist, verfügt das Verwaltungskonto nicht über die erforderl ichen Berechtigungen zum Erstellen und Verwalten von Service-Managed Services StackSets für Ihre Organisation.	5. Juni 2023

	DescribeOrganizationsAccess Ruft Informationen über den `OrganizationAccess` -Status des Kontos ab. Diese API kann mithilfe des Parameters `CallAs` vom Verwaltungskonto oder vom delegierten Administrator aufgerufen werden. Diese API kann auch ohne den Parameter `CallAs` vom Verwaltungskonto aufgerufen werden.	
`DescribeStackSet` -API	Die `DescribeStackSet` API hat einen neuen Parameter in der Liste der Regionen, in denen ein bestimmtes Stack-Set bereitgestellt wird. Weitere Informationen finden Sie unter DescribeStackSet.	1. Februar 2023

Verwaltung von StackSets Veranstaltungen mit CloudFormation und Amazon EventBridge	CloudFormation StackSets Benachrichtigungen über Veranstaltungen über Amazon starten EventBridge. Sie können ereignisgesteuerte Aktionen auslösen, nachdem Sie Ihre CloudFormation Stack-Sets erstellt, aktualisiert oder gelöscht haben. Weitere Informationen findest du unter Überwachung CloudFormation und Git-Synchronisierung von Ereignissen mit EventBridge.	16. November 2022
Verbesserte Einblicke in Stack-Instances für Stack-Set-Operationen	CloudFormation StackSets bietet detailliertere Informationen zu Stack-Instances für Stack-Set-Operationen: DescribeStackSetOperation Sie können `DescribeStackSetOperation` jetzt verwenden, um die Anzahl der fehlgeschlagenen Stack-Instances für Stack-Set-Operationen während der Bereitstellung anzugeben. ListStackInstances Sie können jetzt die Filteroption `LastOperationID` verwenden, um Stack-Instances für Stack-Set-Operationen aufzulisten.	04. November 2022

Verwaltung von Veranstaltungen mit CloudFormation und Amazon EventBridge	Erhalten Sie Benachrichtigungen, wenn bestimmte CloudFormation Ereignisse eintreten. Weitere Informationen findest du unter Überwachung CloudFormation und Git-Synchronisierung von Ereignissen mit EventBridge.	20. Juli 2022
Kontoebene	CloudFormation kündigt die allgemeine Verfügbarkeit des Kontofiltertyps an, einer Funktion, die es Kunden ermöglicht, Bereitstellungsziele auf einzelne Konten zu beschränken oder zusätzliche Konten mit den bereitgestellten Konten hinzuzufügen OUs. Weitere Informationen finden Sie unter Ziele auf Kontoebene für servicemanaged StackSets.	7. Juli 2022
CloudFormation Registrierung	CloudFormation kündigt die allgemeine Verfügbarkeit von Hooks an, einer Funktion, mit der Kunden benutzerdefinierte Logik aufrufen können, um Aktionen zu automatisieren oder Ressourcenkonfigurationen zu überprüfen, bevor ein Stack erstellt, aktualisiert oder gelöscht wird. Weitere Informationen finden Sie im AWS CloudFormation Hooks User Guide.	10. Februar 2022

Stack-Fehleroptionen	Sie können Ihre Anwendungen iterativ entwickeln, wenn Provisionierungsfehler auftreten, indem Sie ab dem Punkt des Fehlers beginnen, ohne erfolgreich bereitgestellte Ressourcen rückgängig zu machen. Durch die Angabe von Optionen für den Stackausfall können Sie Probleme mit Ressourcen in `CREATE_FAILED` oder `UPDATE_FAILED` beheben. Sie können Fehleroptionen für alle Stack-Bereitstellungen und Änderungssatz-Vorgänge bereitstellen. Weitere Informationen finden Sie unter Wählen Sie aus, wie mit Fehlern bei der Bereitstellung von Ressourcen umgegangen werden soll.	30. August 2021
Stack in Stack-Set importieren	Sie können jetzt vorhandene Stacks in neue oder bestehende Stack-Sets importieren. Weitere Informationen finden Sie unter Stacks importieren in. CloudFormation StackSets	28. Juli 2021

Erhöhte Kontingente	Sie können jetzt ein standardmäßiges Maximum an 2000 Stacks in Ihrem angeben. AWS-KontoWeitere Informationen finden Sie unter Grundlegendes zu Kontingenten. CloudFormation	15. Juli 2021
Öffentliche Drittanbieter-Erweiterungen veröffentlichen	Sie können jetzt öffentliche Erweiterungen verwenden, die von Drittanbietern bereitgestellt werden, genauso wie Erweiterungen von AWS. Weitere Informationen finden Sie unter Öffentliche Erweiterungen von Drittanbietern aus der CloudFormation Registrierung verwenden.	21. Juni 2021
Referenzieren von Makros in Stack-Set-Vorlagen	StackSets unterstützt jetzt das Erstellen oder Aktualisieren von Stacksets mit selbstverwalteten Berechtigungen anhand von Vorlagen, die auf Makros verweisen. Weitere Informationen zu Makros finden Sie unter Durchführen einer benutzerdefinierten Verarbeitung von CloudFormation Vorlagen mit Vorlagenmakros.	14. April 2021

Verwenden Sie den neuesten Wert eines Systems Manager Manager-Parameters in einer dynamischen Referenz	Sie können jetzt immer die neueste Version eines Systems Manager Manager-Parameters CloudFormation verwenden, wenn Sie einen Stack erstellen oder aktualisieren. Sie müssen keine bestimmte Version mehr angeben. Weitere Informationen finden Sie unter Abrufen eines Klartext-Werts aus dem Systems Manager Manager-Parameterspeicher.	13. April 2021
Module unterstützen die Verwendung von Perioden-Trennzeichen in Ressourcennamen	Sie können jetzt einen Punkt als Trennzeichen verwenden, um den vollständig qualifizierten logischen Namen für eine in einem Modul enthaltene Ressource anzugeben. Weitere Informationen finden Sie unter Ressourcen für Referenzmodule in CloudFormation Vorlagen.	8. April 2021
CloudFormation StackSets unterstützt jetzt den Einsatz in parallel Regionen	Sie können jetzt wählen, ob Sie die Bereitstellung StackSets in Regionen sequentiell oder parallel durchführen möchten. Weitere Informationen finden Sie unter Optionen für Stack-Set-Operationen.	6. April 2021

CloudFormation StackSets unterstützt jetzt delegierte Administratoren mit AWS Organizations	Zusätzlich zu den Verwaltungskonto der Organisation können delegierte Administratorkonten Stack-Sets mit serviceverwalteten Berechtigungen für ihre Organisation erstellen und verwalten. Weitere Informationen finden Sie unter Registrieren eines delegierten Administrator-Mitgliedskontos und Erstellen CloudFormation StackSets mit vom Dienst verwalteten Rechten.	18. Februar 2021
CloudFormation StackSets Verfügbarkeit in Regionen	CloudFormation StackSets ist jetzt in der Region Asien-Pazifik (Osaka) verfügbar. Weitere Informationen finden Sie unter Konten- und regionsübergreifende Stacks verwalten mit StackSets.	10. Februar 2021

Module	Module sind eine Möglichkeit für Sie, Ressourcenkonfigurationen für die Aufnahme in Stack-Vorlagen auf transparente, verwaltbare und wiederholbare Weise zu erstellen. Module können gängige Servicekonfigurationen und Best Practices als modulare, anpassbare Bausteine zusammenfassen, die Sie in Ihre Stack-Vorlagen aufnehmen können. Weitere Informationen finden Sie unter Erstellen wiederverwendbarer Ressourcenkonfigurationen, die in allen Vorlagen mit CloudFormation Modulen eingebunden werden können.	24. November 2020
Änderungssätze für verschachtelte Stacks	Mit Änderungssätzen für verschachtelte Stacks können Sie eine Vorschau der Änderungen an Ihren Anwendungs- und Infrastrukturressourcen in der gesamten verschachtelten Stack-Hierarchie anzeigen und mit Aktualisierungen fortfahren, wenn Sie bestätigt haben, dass alle Änderungen wie beabsichtigt sind. Weitere Informationen finden Sie unter Änderungssätze für verschachtelte Stacks.	18. November 2020

Erhöhte Kontingente	Die folgenden AWS CloudFormation Kontingente wurden aktualisiert. • Sie können jetzt maximal `200` Zuordnungen in Ihrer AWS CloudFormation -Vorlage deklarieren. • Sie können jetzt für jedes `200` Mapping in Ihrer AWS CloudFormation Vorlage ein Maximum an Zuordnung sattributen deklarieren. • Sie können jetzt maximal `200` Ausgaben in Ihrer AWS CloudFormation -Vorlage deklarieren. • Sie können jetzt maximal `200` Parameter in Ihrer AWS CloudFormation -Vorlage deklarieren. • Sie können jetzt maximal `500` Ressourcen in Ihrer AWS CloudFormation -Vorlage deklarieren. • Sie können nun einen Vorlagenkörper mit einer maximalen Größe von `1 MB` in einem Amazon S3-Objekt übergeben.	22. Oktober 2020

Abweichungserkennung für private Ressourcen	CloudFormation unterstützt jetzt Operationen zur Erkennung von Abweichungen bei einer erweiterten Liste von AWS Ressourcen sowie bei privaten Ressourcen, die in der CloudFormation Registrierung als provisorisch definiert sind. Weitere Informationen finden Sie unter Unterstützung von Ressourcentypen.	1. Oktober 2020
Aktualisierte Berechtigungen für die Registrierung von Ressourcenanbietern	Für die Registrierung eines Ressourcenanbieters in Ihrem Konto müssen Sie jetzt über die Berechtigung verfügen, auf das Schema-Handler-Paket zuzugreifen, das in einen S3-Bucket für diesen Ressourcenanbieter hochgeladen wurde. Weitere Informationen finden Sie unter IAM-Berechtigungen für die Registrierung einer privaten Erweiterung eines Drittanbieters.	7. August 2020
Der Ressourcenimport unterstützt bereitstellbare private Ressourcentypen.	Importoperationen unterstützen jetzt private Ressourcentypen, die bereitstellbar sind, d. h. deren Bereitstellungstyp entweder `FULLY_MUTABLE` oder `IMMUTABLE` ist. Weitere Informationen finden Sie unter Ressourcen, die Importvorgänge unterstützen.	3. Juni 2020

blue/green ECS-Bereitstellungen über CodeDeploy	Sie können es jetzt verwenden CloudFormation , um blue/green ECS-Bereitstellungen durchzuführen über. CodeDeploy Blue/green Bereitstellungen sind eine sichere Bereitstellungsstrategie AWS CodeDeploy zur Minimierung von Unterbrechungen, die durch wechselnde Anwendungsversionen verursacht werden. Weitere Informationen finden Sie unter Durchführen von blue/green ECS-Bereitstellungen mithilfe von. CodeDeploy CloudFormation	19. Mai 2020
CloudFormation StackSets Verfügbarkeit in Regionen	CloudFormation StackSets ist jetzt in der Region AWS GovCloud (USA West) verfügbar.	18. Mai 2020

AWS CloudFormation StackSets integriert mit AWS Organizations	Sie können es jetzt verwenden StackSets , um Bereitstellungen für alle Konten in Ihrer Organisation oder für bestimmte Organisationseinheiten (OUs) in AWS Organizations zentral zu verwalten. Sie können automatische Bereitstellungen für alle neuen Konten aktivieren, die Ihrer Organisation hinzugefügt wurden oder. OUs Die für die kontenübergreifende Bereitstellung erforderlichen Berechtigungen werden automatisch von StackSets verwaltet. Weitere Informationen finden Sie unter Konten- und regionsübergreifende Stacks verwalten mit StackSets.	11. Februar 2020
Drift-Erkennung für StackSets	Sie können jetzt die Abweichungserkennung für ein Stack-Set und alle darin eingesetzten Stack-Instances ausführen. Weitere Informationen finden Sie unter Drifterkennung durchführen auf CloudFormation StackSets .	19. November 2019

CloudFormation Registrierung jetzt verfügbar	Sie können jetzt die CloudFormation Konsole verwenden, um private und öffentliche Ressourcen anzuzeigen, die für die Verwendung in Ihrem Konto verfügbar sind. Weitere Informationen finden Sie unter Verfügbare und aktivierte Erweiterungen in der CloudFormation Registrierung anzeigen.	18. November 2019

CloudFormation API-Aktionen für die Registrierung	Die folgenden API-Aktionen für die Verwaltung von Typen in der CloudFormation Registrierung sind jetzt verfügbar. DeregisterType Entfernt einen Typ oder eine Typversion aus der aktiven Verwendung in der CloudFormation Registrierung. DescribeType Gibt detaillierte Informationen zu einem registrierten Typ zurück. DescribeTypeRegistration Gibt Informationen zur Registrierung eines Typs zurück, einschließlich aktuellem Status sowie Typ und Versions-IDs. ListTypeRegistrations Gibt eine Liste von Registrierungsanforderungskennungen für den angegebenen Typ zurück. ListTypes Gibt zusammenfassende Informationen über Typen zurück, mit denen registriert wurde CloudFormation.	18. November 2019

ListTypeVersions

Gibt zusammenfassende Informationen zu den Versionen eines Typs zurück.

RegisterType

Registriert einen Typ in der CloudFormation Registrierung. Durch die Registrierung eines Typs steht er für die Verwendung in CloudFormation Vorlagen in Ihrem zur Verfügung AWS-Konto.

SetTypeDefaultVersion

Geben Sie die Standardversion eines Typs an. Die Standardversion eines Typs wird in CloudFormation Operationen verwendet.

Weitere Informationen zur CloudFormation Registrierung finden Sie unter Erweiterungen mit der CloudFormation Registrierung verwalten

Ressourcenimport hinzugefügt	Wenn Sie eine AWS Ressource außerhalb der CloudFormation Verwaltun g erstellt haben, können Sie diese vorhandene Ressource mit in die CloudFormation Verwaltung übernehme `nresource import`. Weitere Informationen finden Sie unter AWS Ressourcen mit einem Ressourcenimport in einen CloudFormation Stapel importieren.	11. November 2019
Erhöhungen des Stack-Set-Limits	Sie können jetzt maximal 100 Stack-Sets in Ihrem Administratorkonto erstellen, maximal 2000 Stack-Instances pro Stack-Set erstellen und maximal 3500 Stack-Instance-Operationen in jeder Region gleichzeitig pro Administr atorkonto ausführen. Weitere Informationen finden Sie unter Grundlegendes zu CloudForm ation Kontingenten.	2. August 2019
Limits für Ressourcen bei gleichzeitigen Stack-Vor gängen	CloudFormation erzwingt jetzt ein Kontolimit für die Anzahl der Ressourcen bei gleichzei tigen Stack-Vorgängen. Dieses Limit wird durch die Region bestimmt. Weitere Informationen finden Sie unter Grundlegendes zu Kontingen ten CloudFormation .	30. April 2019

Operative Begrenzung der Stack-Instance	Denn StackSets pro Administratorkonto können jetzt maximal 1500 Stack-Instance-Operationen in einer bestimmten Region gleichzeitig ausgeführt werden. Weitere Informationen finden Sie unter Grundlegendes zu CloudFormation Kontingenten.	13. Dezember 2018
Die `CAPABILITY_AUTO_EXPAND` Funktion ist jetzt verfügbar	Sie können die `CAPABILITY_AUTO_EXPAND` Funktion jetzt verwenden, um einen Stack direkt aus einer Stackvorlage, die Makros enthält, zu erstellen oder zu aktualisieren, ohne zuerst die resultierenden Änderungen in einem Änderungssatz überprüfen zu müssen. Weitere Informationen finden Sie im Abschnitt zu CreateStack oder UpdateStack in der API-Referenz zu AWS CloudFormation .	7. Dezember 2018

Erkennung von Stack-Abweichungen hinzugefügt	Sie können jetzt erkennen, ob die tatsächliche Konfiguration eines Stacks von der erwarteten Vorlagenkonfiguration, wie darin definiert, abgewichen ist. CloudFormation Sie können Abweichungen bei einem gesamten Stack oder bei einzelnen Stack-Ressourcen erkennen. Weitere Informationen finden Sie unter Erkennen nicht verwalteter Konfigurationsänderungen an Stacks und Ressourcen mithilfe der Drift-Erkennung.	13. November 2018

`secretsmanager` dynamische Referenz jetzt verfügbar	Sie können die `secretsmanager` dynamische Referenz jetzt verwenden, um ganze Geheimnisse oder geheime Werte abzurufen, die in gespeichert sind AWS Secrets Manager. Bei den Secrets kann es sich um Datenbank-Anmeldeinformationen, Passwörter, API-Schlüssel von Drittanbietern und sogar beliebigen Text handeln. Die Verwendung der `secretsmanager` dynamischen Referenz garantiert, dass weder Secrets Manager noch aufgelöste geheime Werte CloudFormation protokollieren oder beibehalten. Weitere Informationen finden Sie unter Abrufen eines geheimen oder geheimen Werts aus Secrets Manager.	9. November 2018

Makros sind jetzt verfügbar	Sie können jetzt Makros verwenden, um benutzerd efinierte Verarbeitungen an Vorlagen durchzuführen, von einfachen Aktionen wie find-and-replace Operation en bis hin zu umfangrei chen Transformationen ganzer Vorlagen. Weitere Informationen finden Sie unter Durchführen einer benutzerd efinierten Verarbeitung von CloudFormation Vorlagen mit Vorlagenmakros.	6. September 2018
CloudFormation unterstüt zt jetzt VPC-Endpunkte, die betrieben werden von PrivateLink	Sie können einen VPC-Endpu nkt verwenden, um eine private Verbindung zwischen Ihrer VPC und CloudForm ation herzustellen, ohne dass ein Zugriff über das Internet, über eine NAT-Instance, eine VPN-Verbindung oder AWS Direct Connect erforderlich ist. Weitere Informationen finden Sie unter Zugriff CloudForm ation über einen Schnittst ellenendpunkt ()AWS PrivateLi nk.	22. August 2018

Dynamische Referenzen unterstützen sichere Zeichenfolgen	Sie können jetzt neue dynamische Verweise verwenden, um Werte, die in anderen Diensten gespeichert und verwaltet werden, einschließlich Parametern `SecureString` vom Typ Systems Manager Parameter Store, in Ihren Stack-Vorlagen anzugeben. Weitere Informationen finden Sie unter Abrufen eines sicheren Zeichenkettenwerts aus dem Systems Manager Manager-Parameterspeicher.	16. August 2018
Stack-Sets unterstützen jetzt benutzerdefinierte Ausführungsrollen.	Sie können jetzt benutzerdefinierte Ausführungsrollen in Zielkonten verwenden, um die Stack-Ressourcen zu steuern, die Benutzer oder Gruppen in ihre Stack-Sets aufnehmen können. Weitere Informationen finden Sie unter Erweiterte Berechtigungsoptionen für Stack-Set-Operationen einrichten.	30. Mai 2018

Ausgewählte Aktualisierungen von Stack-Instances	Sie können jetzt die optionalen `Regions` Parameter `Accounts` und verwenden, um die Konten und Regionen anzugeben, in denen Stack-Instances während eines Stack-Set-Aktualisierungsvorgangs aktualisiert werden sollen. Weitere Informationen finden Sie unter UpdateStackSet in der AWS CloudFormation -API-Referenz.	30. Mai 2018
CloudFormation erstellt jetzt S3-Buckets mit aktivierter Verschlüsselung	Für Amazon S3 S3-Buckets, die zum Speichern hochgeladener Stack-Vorlagen CloudFormation erstellt werden, ist die serverseitige Verschlüsselung jetzt standardmäßig aktiviert, wodurch alle in diesen Buckets gespeicherten Objekte verschlüsselt werden. Weitere Informationen finden Sie unter Einen Stack von der Konsole aus erstellen. CloudFormation	24. Mai 2018

Hinzufügung von FIPS-Endpunkten	CloudFormation bietet jetzt neue Endpunkte, die FIPS 140-2-validierte kryptografische Module in den folgenden öffentlichen US-Regionen verwenden: US-East-1, US-East-2, US-West-1 und US-West-2. AWS CloudFormation Informationen zu Endpunkten und Kontingenten finden Sie URLs im für den neuen Allgemeine Amazon Web Services-ReferenzFIPS-kompatiblen Endpunkt.	17. Mai 2018

Aktualisierungen des AWS CloudFormation Hooks User Guide finden Sie im Hooks User Guide unter Document History for the AWS CloudFormation Hooks User Guide.AWS CloudFormation

Archivierte Updates

In der folgenden Tabelle werden wichtige Änderungen beschrieben, die in jeder Version des AWS CloudFormation Benutzerhandbuchs vor Mai 2018 vorgenommen wurden.

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	22. Juli 2019	Verwenden Sie die `encryptionOptions` -Eigenschaft, um einen AWS-eigener Schlüssel oder einen kundenverwalteten Schlüssel für Amazon-MQ-Broker anzugeben.	15.05.2010
Namenskonvention für Stack-Sets	10. April 2018	CloudFormation Stapel, die mit Stack-Sets erstellt wurden, folgen nun einer neuen Namenskonvention, bei der der Stack-Name den Namen des Stack-Sets enthält.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	10. April 2018	AWS::AppSync::ApiKey Verwenden Sie die `AWS::AppSync::ApiKey` Ressource, um einen eindeutigen Schlüssel zu erstellen, den Sie an Clients verteilen können, mit AWS AppSync denen GraphQL-Operationen ausgeführt werden. AWS::AppSync::DataSource Verwenden Sie die `AWS::AppSync::DataSource` Ressource, um Datenquellen für Resolver in zu erstellen. AWS AppSync AWS::AppSync::GraphQLApi Verwenden Sie die `AWS::AppSync::GraphQLApi` Ressource, um eine neue AWS AppSync GraphQL-API zu erstellen. AWS::AppSync::GraphQLSchema Verwenden Sie die `AWS::AppSync::GraphQLSchema` Ressource, um das Datenmodell für Ihre AWS AppSync GraphQL-API zu erstellen. AWS::AppSync::Resolver Verwenden Sie die Ressource `AWS::AppSync::Resolver` , um den logischen GraphQL-Resolver zu definieren, den Sie an Felder in einem Schema anfügen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressource	10. April 2018	AWS::Config::ConfigurationAggregator Verwenden Sie den `OrganizationAggregationSource` Eigenschaftstyp, um die AWS Config Datenbereiche anzugeben, die zu einem AWS Config Konfigurationsaggregator zusammengefasst werden sollen, und die IAM-Rolle, die zum Abrufen von Details verwendet werden soll. AWS Organizations	15.05.2010
Neue Ressourcen	4. April 2018	AWS::Config::AggregationAuthorization Verwenden Sie die `AWS::Config::AggregationAuthorization` Ressource, um einem Aggregatorkonto die Erlaubnis zur Erfassung Ihrer Daten zu erteilen. AWS Config AWS::Config::ConfigurationAggregator Verwenden Sie die `AWS::Config::ConfigurationAggregator` -Ressource , um einen Konfigurationsaggregator für AWS Config zu erstellen .	15.05.2010
Stack-Sets unterstützen jetzt benutzerdefinierte Administratorrollen.	29. März 2018	Steuern Sie mithilfe von benutzerdefinierten Administratorrollen, welche Benutzer oder Gruppen bestimmte StackSets innerhalb desselben Administratorkontos verwalten können. Weitere Informationen finden Sie unter Erweiterte Berechtigungsoptionen für Stack-Set-Operationen einrichten.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressource	29. März 2018	AWS::EC2::LaunchTemplate Verwenden Sie die `AWS::EC2::LaunchTemplate` Ressource, um eine Startvorlage für eine EC2 Amazon-Instance zu erstellen.	15.05.2010
Aktualisierte Ressourcen	29. März 2018	AWS::AutoScaling::AutoScalingGroup Verwenden Sie die Eigenschaft `LaunchTemplate` , um die Einführungsvorlage für das Starten von Instances festzulegen. AWS::EC2::SpotFleet Verwenden Sie im Eigenschaftstyp SpotFleetRequestConfigData die Eigenschaft `LaunchTemplateConfigs` , um eine Startvorlage und Überschreibungen zu beschreiben.	15.05.2010
Neue intrinsische Funktion `Fn::Cidr`	6. März 2018	Gibt den angegebenen Cidr-Adressblock zurück. Weitere Informationen finden Sie unter Fn::Cidr.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	6. März 2018	AWS::ApiGateway::VpcLink Verwenden Sie die Ressource `AWS::ApiGateway::VpcLink` zur Festlegung eines API Gateway VPC-Links für eine `AWS::ApiGateway::RestApi`, um auf Ressourcen in einer Amazon Virtual Private Cloud (VPC) zuzugreifen. AWS::GuardDuty::Master Verwenden Sie die `AWS::GuardDuty::Master` Ressource, um ein GuardDuty primäres Konto zu erstellen. AWS::GuardDuty::Member Verwenden Sie die `AWS::GuardDuty::Member` Ressource, um ein GuardDuty Mitgliedskonto zu erstellen. AWS::SES::ConfigurationSet Verwenden Sie die Ressource `AWS::SES::ConfigurationSet`, um Gruppen von Regeln zu erstellen, die Sie auf die E-Mails, die Sie senden, anwenden können. AWS::SES::ConfigurationSetEventDestination Mit der Ressource `AWS::SES::ConfigurationSetEventDestination` können Sie ein Konfigurationssatz-Ereigniszieł angeben. AWS::SES::ReceiptFilter Mit der Ressource `AWS::SES::ReceiptFilter` können Sie angeben, ob E-Mails, die von einer IP-	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		Adresse oder einem IP-Adressbereich stammen, akzeptiert oder abgelehnt werden sollen. AWS::SES::ReceiptRule Mit der Ressource `AWS::SES::ReceiptRule` können Sie angeben, welche Aktionen Amazon SES ausführen soll, wenn E-Mails im Namen eines oder mehrerer E-Mail-Adressen oder Domainn, die Sie besitzen, empfangen werden. AWS::SES::ReceiptRuleSet Mit der Ressource `AWS::SES::ReceiptRuleSet` können Sie einen leeren Regelsatz für Amazon SES angeben. AWS::SES::Template Verwenden Sie die Ressource `AWS::SES::Template`, um den Inhalt der E-Mail anzugeben, bestehend aus einer Betreffzeile, einem HTML-Teil und einem Nur-Text-Teil.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	6. März 2018	AWS::AutoScaling::AutoScalingGroup Mit der Eigenschaft `AutoScalingGroupName` können Sie den Namen der Auto-Scaling-Gruppe angeben. AWS::ApiGateway::RestApi Mit der Eigenschaft `ApiKeySourceType` können Sie die Quelle des API-Schlüssels für Messungsanforderungen nach einem Nutzungsplan angeben. Verwenden Sie die `MinimumCompressionSize` -Eigenschaft, um eine nullwertfähige Ganzzahl zur Aktivierung oder Deaktivierung der Komprimierung in einer API anzugeben. AWS::ApplicationAutoScaling::ScalingPolicy Verwenden Sie im Eigenschaftstyp TargetTrackingScalingPolicyConfiguration die Eigenschaft `DisableScaleIn` , um anzugeben, ob das Herunterskalieren durch die Richtlinie für die Ziel-Nachverfolgung deaktiviert ist. AWS::EC2::SpotFleet Verwenden Sie im LaunchSpecificationsEigenschaftstyp die `TagSpecifications` Eigenschaft, um die Tags anzugeben, die bei der SpotFleet Erstellung angewendet werden sollen. AWS::Elasticsearch::Domain Verwenden Sie das Attribut `Arn`, damit `Fn::GetAtt` den Amazon-Ressourcennamen (ARN) der Domain zurückgibt.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		Das Attribut `DomainArn` von `Fn::GetAtt` wurde als veraltet gekennzeichnet. AWS::RDS::DBCluster Mit der Eigenschaft `DBClusterIdentifier` können Sie die DB-Cluster-Kennung angeben. AWS::RDS::DBCluster Mit der Eigenschaft `DBClusterIdentifier` können Sie die DB-Cluster-Kennung angeben. AWS::Redshift::Cluster Mit der Eigenschaft `ClusterIdentifier` können Sie die eindeutige Kennung des Clusters angeben. AWS::Route53::HealthCheck Geben Sie im HealthCheckConfigEigenschaftstyp mithilfe der `Regions` Eigenschaft die Regionen an, von denen aus Route 53-Integritätsprüfungen den angegebenen Endpunkt überprüfen sollen. AWS::SSM::Document Mit der Eigenschaft `Tags` können Sie die CloudFormation-Ressourcen-Tags für das Dokument angeben.	
Aktualisierte Ressource	19. Februar 2	AWS::CodeBuild::Project Mit der Eigenschaft `Triggers` können Sie einen Webhook konfigurieren, mit dem das Projekt beginnt, den Quellcode automatisch neu zu erstellen, sobald eine Code-Änderung in das Repository verschoben wird. Dies ist nur für GitHub Projekte in CloudFormation verfügbar. Es ist nicht für GitHub Enterprise-Projekte verfügbar.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressource	8. Februar 2018	AWS::DynamoDB::Table Mit der Eigenschaft `SSESpecification` können Sie die Einstellungen zum Aktivieren der serverseitigen Verschlüsselung angeben.	15.05.2010
Aktualisierte Ressource	5. Februar 2018	AWS::CodeBuild::Project Im Eigenschaftstyp Source `CodeBuild Project Source`: • Mit der Eigenschaft `GitCloneDepth` können Sie die Tiefe des herunterzuladenden Verlaufs angeben. • Verwenden Sie die `InsecureSsl` Eigenschaft, um anzugeben, ob SSL-Warnungen ignoriert werden sollen, während Sie eine Verbindung zu Ihrem GitHub Enterprise-Projekt-Repository herstellen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	23. Januar 2018	AWS::AutoScaling::LifecycleHook Mit der Eigenschaft `LifecycleHookName` können Sie den Namen des Lebenszyklus-Hook anzugeben. AWS::DynamoDB::Table Die Eigenschaft `AttributeDefinitions` muss beim Aktualisieren nun ersetzt werden. AWS::EC2::Instance Mit der Eigenschaft `CreditSpecification` können Sie die Guthaben-Option für die CPU-Nutzung einer T2-Instance angeben. Verwenden Sie die `ElasticGpuSpecifications` Eigenschaft GPUs, um Elastic-GPU-Ressourcen anzugeben, die Sie Ihrer Instance zuordnen können, um die Grafikleistung Ihrer Anwendungen zu beschleunigen. AWS::EC2::VPC Für die Eigenschaft `InstanceTenancy` ist nun keine Unterbrechung erforderlich, wenn sie von `"dedicated"` auf `"default"` aktualisiert wird. AWS::ECS::Service Mit der Eigenschaft `HealthCheckGracePeriodSeconds` können Sie in Sekunden angeben, wie lange der Amazon ECS-Service-Scheduler fehlerhafte Elastic Load Balancing-Zustandsprüfungen des Ziels ignoriert, nachdem eine Aufgabe zum ersten Mal gestartet wurde.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::IoT::TopicRule Im Eigenschaftstyp DynamoDBAction werden die Eigenschaften `RangeKeyField` und `RangeKeyValue` nicht mehr benötigt. AWS::KinesisAnalytics::ApplicationOutput Im -Eigenschaftstyp ApplicationOutput können Sie beim Konfigurieren der Anwendungsausgabe mit der Eigenschaft `LambdaOutput` eine Lambda-Funktion als Ziel identifizieren. AWS::Kinesis::Stream Verwenden Sie die `StreamEncryption` Eigenschaft, um die serverseitige Verschlüsselung mithilfe von AWS KMS key für einen bestimmten Stream zu aktivieren oder zu aktualisieren. AWS::Lambda::Function Mit der Eigenschaft `ReservedConcurrentExecutions` können Sie die maximale Anzahl gleichzeitiger Ausführungen angeben, die für die Funktion reserviert werden sollen. AWS::RDS::DBSubnetGroup Mit der Eigenschaft `DBSubnetGroupName` können Sie den Namen für die DB-Subnetzgruppe anzugeben. AWS::S3::Bucket Verwenden Sie die `BucketEncryption` Eigenschaft, um die Standardverschlüsselung für einen Bucket anzugeben, der serverseitige Verschlüsselung mit	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		von Amazon S3 verwalteten Schlüsseln SSE-S3 oder AWS KMS keys (SSE-KMS) Bucket verwendet. Im Eigenschaftstyp ReplicationRule können Sie mit der Eigenschaft `SourceSelectionCriteria` zusätzliche Filter bei der Identifizierung von Quellobjekten angeben, die repliziert werden sollen. Im Eigenschaftstyp ReplicationDestination: • Verwenden Sie die `AccessControlTranslation` Eigenschaft, um den Besitz des Replikats für das Konto anzugeben, dem der Ziel-Bucket gehört. AWS • Mit der Eigenschaft `Account` können Sie die Konto-ID des Ziel-Bucket-Eigentümers angeben. • Mit der Eigenschaft `EncryptionConfiguration` können Sie verschlüsselungsbezogene Informationen für einen Bucket angeben, der ein Ziel für replizierte Objekte ist. AWS::SSM::Association Verwenden Sie die `AssociationName` Eigenschaft, um den Namen der Verknüpfung zwischen einem SSM-Dokument und EC2 Instanzen anzugeben, die einen Konfigurationsagenten zur Verarbeitung des Dokuments enthalten.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Rollback-Trigger wurden der CloudFormation Konsole hinzugefügt.	15. Januar 2018	Rollback-Trigger ermöglichen es Ihnen, den Status Ihrer Anwendung während der Stack-Erstellung und Aktualisierung zu CloudFormation überwachen und diesen Vorgang rückgängig zu machen, wenn die Anwendung den Schwellenwert eines der von Ihnen angegebenen Alarme überschreitet. Weitere Informationen finden Sie unter Überwachen und Zurücksetzen von Stack-Operationen.	15.05.2010
Aktualisierte Ressource	12. Januar 2018	AWS::SSM::Parameter Verwenden Sie die `AllowedPattern` -Eigenschaft, um einen regulären Ausdruck anzugeben, der für die Überprüfung des Parameterwerts verwendet wird.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	5. Dezember 2017	AWS::Inspector::AsssmentTarget Mit der Ressource `AWS::Inspector::AssmentTarget` können Sie ein Amazon Inspector -Bewertungsziel erstellen. AWS::Inspector::AssessmentTemplate Mit der Ressource `AWS::Inspector::AssessmentTemplate` können Sie eine Amazon Inspector-Bewertungsvorlage erstellen. AWS::Inspector::ResourceGroup Verwenden Sie die Ressource `AWS::Inspector::ResourceGroup` , um eine Amazon-Inspector-Ressourcengruppe zu erstellen, die Tags definiert, mit denen die AWS -Ressourcen eines Amazon-Inspector-Bewertungsziels identifiziert werden. AWS::ServiceDiscovery::Instance Mit der Ressource `AWS::ServiceDiscovery::Instance` können Sie Informationen zu einer Instance angeben, die von Amazon Route 53 erstellt wird. AWS::ServiceDiscovery::PrivateDnsNamespace Mit der Ressource `AWS::ServiceDiscovery::PrivateDnsNamespace` können Sie Informationen zu einem privaten Namespace für Amazon Route 53 angeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::ServiceDiscovery::PublicDnsNamespace Mit der Ressource `AWS::ServiceDiscovery::PublicDnsNamespace` können Sie Informationen zu einem öffentlichen Namespace für Amazon Route 53 angeben. AWS::ServiceDiscovery::Service Sie können mit der Ressource `AWS::ServiceDiscovery::Service` eine Vorlage für bis zu fünf Datensätze und eine optionale Zustandsprüfung definieren, die Amazon Route 53 erstellen soll, wenn Sie eine Instance registrieren.	
Aktualisierte Ressource	5. Dezember 2017	AWS::KinesisAnalytics::Application Im Eigenschaftstyp Input können Sie mit der Eigenschaft `InputProcessingConfiguration` Datensätze transformieren, wenn diese vom Stream empfangen werden.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressource	1. Dezember	AWS::CodeBuild::Project Mit der Eigenschaft `BadgeEnabled` können Sie eine öffentlich zugängliche URL für den Build-Auswahl eines Projekts erstellen. Mit der Eigenschaft `Cache` können Sie Cache-Einstellungen für Build-Abhängigkeiten konfigurieren. Verwenden Sie die `VpcConfig` Eigenschaft, um den Zugriff auf Ressourcen in einer Amazon VPC CodeBuild zu ermöglichen. Im Eigenschaftstyp EnvironmentVariable können Sie mit der Eigenschaft `Type` die Art der Umgebungsvariablen angeben.	15.05.2010
Neue Ressource	30. November 2017	AWS::Cloud9::EnvironmentEC2 Verwenden Sie die `AWS::Cloud9::EnvironmentEC2` Ressource, um eine EC2 Amazon-Entwicklungsumgebung in zu erstellen AWS Cloud9.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	29. November 2017	AWS::ECS::TaskDefinition Mit der Eigenschaft `Cpu` können Sie die Anzahl der CPU-Einheiten angeben, die für die zu erledigend Aufgabe erforderlich sind. Mit der Eigenschaft `ExecutionRoleArn` können Sie den ARN der Ausführungsrolle angeben. Mit der Eigenschaft `Memory` können Sie die Menge (in MiB) an Speicher angeben, die für die zu erledigend Aufgabe erforderlich ist. Mit der Eigenschaft `RequiresCompatibilities` können Sie angeben, welcher Starttyp für die Aufgabe benötigt wird. AWS::ECS::Service Mit der Eigenschaft `LaunchType` können Sie den Starttyp angeben, auf dem Ihr Service ausgeführt werden soll. Mit der Eigenschaft `NetworkConfiguration` können Sie die Netzwerkkonfiguration für den Service angeben. Mit der Eigenschaft `PlatformVersion` können Sie die Version der Plattform angeben, auf der Ihr Service ausgeführt wird.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	28. November 2017	AWS::GuardDuty::Detector Verwenden Sie die `AWS::GuardDuty::Detector` Ressource, um einen einzelnen GuardDuty Amazon-Detektor zu erstellen. AWS::GuardDuty::IPSet Verwenden Sie die `AWS::GuardDuty::IPSet` Ressource, um einen GuardDuty Amazon-IP-Satz zu erstellen. AWS::GuardDuty::ThreatIntelSet Verwenden Sie die `AWS::GuardDuty::ThreatIntelSet` Ressource, um eine zu erstellen ThreatIntelSet.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	28. November 2017	AWS::CodeDeploy::Application Verwenden Sie die `ComputePlatform` Eigenschaft, um eine AWS Lambda Rechenplattform anzugeben CodeDeploy , auf der eine Anwendung bereitgestellt werden soll. AWS::CodeDeploy::DeploymentGroup Im Eigenschaftstyp DeploymentStyle können Sie mit der Eigenschaft `DeploymentType` eine Blau/Grün-Bereitstellung einer Lambda-EDV-Plattform bereitstellen. AWS::EC2::SpotFleet Im Eigenschaftstyp SpotFleetRequestConfigData ist die Eigenschaft `SpotPrice` nun optional. AWS::Lambda::Alias Verwenden Sie die `RoutingConfig` Eigenschaft, um zwei verschiedene Versionen einer AWS Lambda Funktion anzugeben, sodass Sie festlegen können, wie viel Prozent des Datenverkehrs die einzelnen Versionen aufrufen.	15.05.2010
Neue `CodeDeployLambdaAliasUpdate` -Aktualisierungsrichtlinie	28. November 2017	Verwenden Sie die `CodeDeployLambdaAliasUpdate` Aktualisierungsrichtlinie, um eine CodeDeploy Bereitstellung durchzuführen, wenn sich die Version auf einer `AWS::Lambda::Alias` Ressource ändert. Weitere Informationen finden Sie unter UpdatePolicyAttribut.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue `SSM`-Parametertypen	21. November 2017	Mit den Parametertypen `SSM` können Sie vorhandene Parametern aus dem Systems Manager Parameter Store nutzen. Hinweis: CloudFormation unterstützt den Typ `SecureString` derzeit nicht. Weitere Informationen finden Sie unter SSM-Parametertypen.	15.05.2010
Neues `ResolvedValue` - Feld für `Parameter` - Datentyp	21. November 2017	Das Feld `ResolvedValue` gibt den Wert zurück, der in der Stack-Definition für einen `SSM`-Parameter verwendet wird. Weitere Informationen finden Sie im Abschnitt zum Datentyp Parameter in der API-Referenz zu AWS CloudFormation .	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	20. November 2017	AWS::ApiGateway::ApiKey Verwenden Sie die `CustomerId` -Eigenschaft, um eine AWS -Marketplace-Kunden-ID anzugeben. Mit der Eigenschaft `GenerateDistinctId` können Sie angeben ob sich die Schlüssel-ID von dem erstellten API-Schlüsselwert unterscheidet. AWS::ApiGateway::Authorizer Mit der Eigenschaft `AuthType` können Sie ein benutzerdefiniertes Feld angeben, das in Swagger-Importen und -Exporten ohne funktionale Auswirkungen verwendet wird. AWS::ApiGateway::DomainName Mit der Eigenschaft `EndpointConfiguration` können Sie die Endpunkttypen eines API Gateway-Domainnamens angeben. Mit der Eigenschaft `RegionalCertificateArn` können Sie auf Zertifikat für die Verwendung durch den regionalen Endpunkt für einen Domainnamen verweisen. AWS::ApiGateway::Method Bei den Eigenschaftstypen Integration und IntegrationResponse können Sie mit der Eigenschaft `ContentHandling` angeben, wie Umwandlungen des Anforderungs-Nutzlast-Inhaltstyps vollzogen werden sollen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::ApiGateway::RestApi Mit der Eigenschaft `EndpointConfiguration` können Sie die Endpunkttypen einer API Gateway-REST-API angeben. AWS::ApplicationAutoScaling::ScalableTarget Mit der Eigenschaft `ScheduledActions` können Sie geplante Aktionen für ein skalierbares Auto Scaling von Anwendungen-Ziel angeben. AWS::ECR::Repository Mit der Eigenschaft `LifecyclePolicy` können Sie eine Lebenszyklusrichtlinie für ein Amazon ECR-Repository angeben. AWS::ECS::TaskDefinition Im Eigenschaftstyp ContainerDefinition können Sie mit der Eigenschaft `LinuxParameters` Linux-spezifische Optionen für einen Amazon ECS-Container angeben. AWS::ElastiCache::ReplicationGroup Mit der Eigenschaft `AtRestEncryptionEnabled` können Sie die Verschlüsselung im Ruhezustand aktivieren. Mit der Eigenschaft `AuthToken` können Sie ein Passwort für den Zugriff auf einen passwortgeschützten Server angeben. Mit der Eigenschaft `TransitEncryptionEnabled` können Sie die Verschlüsselung während der Übertragung aktivieren.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::ElasticLoadBalancingV2::TargetGroup Sie können das Attribut `TargetGroupName` mit der Funktion `Fn::GetAtt` verwenden, um den Namen einer Elastic Load Balancing Zielgruppe abzurufen. AWS::Elasticsearch::Domain Verwenden Sie die `VPCOptions` Eigenschaft, um eine VPC-Konfiguration für die OpenSearch Service-Domain anzugeben. AWS::EMR::Cluster Mit der Eigenschaft `EbsRootVolumeSize` können Sie die Größe des EBS-Stamm-Volumes für einen Amazon EMR-Cluster angeben. AWS::RDS::DBInstance Mit den Eigenschaften `SourceRegion` und `KmsKeyId` können Sie eine verschlüsselte Read Replica aus einer regionsübergreifenden Quell-DB-Instance erstellen. AWS::Route53::HostedZone Mit der Eigenschaft `QueryLoggingConfig` können Sie eine Konfiguration für die DNS-Abfragegeprotokollierung angeben.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neues `NoEcho`-Feld für benutzerdefinierte Ressourcen-`Response`-Objekte	20. November 2017	Sie können nun das optionalen Feld `NoEcho` verwenden, um die Ausgabe einer benutzerdefinierten Ressource zu maskieren. Weitere Informationen finden Sie unter Benutzerdefinierte Ressourcenantwortobjekte. Der entsprechende `noEcho`-Parameter wird von der Methode `send` unterstützt. Weitere Informationen finden Sie unter cfn-response-Modul.	15.05.2010
Stack-Instance-Überschreibungen für Stack-Sets hinzugefügt.	17. November 2017	CloudFormation StackSets ermöglicht es Ihnen, Parameterwerte in Stack-Instanzen nach Konto und Region zu überschreiben. Sie können Parameterwerte beim Erstellen der Stack-Instances oder beim Aktualisieren vorhandener Stack-Instances überschreiben. Weitere Informationen finden Sie unter Überschreiben von Parametern auf Stack-Instances.	15.05.2010
Aktualisierte Ressource	15. November 2017	AWS::StepFunctions::StateMachine Sie können mit `AWS::StepFunctions::StateMachine` einen `StateMachineName` angeben, wenn Sie einen Zustandsautomaten erstellen, und sowohl `DefinitionString` als auch `RoleArn` kann aktualisiert werden, ohne den Zustandsautomaten zu ersetzen.	15.05.2010
StackSets unterstützt jetzt maximal 500 Stack-Instances pro Stack-Set.	6. November 2017	Sie können jetzt bis zu 500 Stack-Instances pro Stack-Set erstellen. Weitere Informationen zu AWS CloudFormation Limits finden Sie unter Grundlegendes zu CloudFormation Kontingenten.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	2. November 2017	AWS::CloudFront::CloudFrontOriginAccessIdentity Verwenden Sie die `AWS::CloudFront::CloudFrontOriginAccessIdentity` Ressource, um die CloudFront Amazon-Ursprungszugriffsidentität anzugeben, die dem Ursprung einer CloudFront Distribution zugeordnet werden soll. AWS::CloudFront::StreamingDistribution Verwenden Sie die `AWS::CloudFront::StreamingDistribution` Ressource, um eine Adobe Real-Time Messaging Protocol (RTMP) - Streaming-Verteilung für CloudFront anzugeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	2. November 2017	AWS::ApiGateway::Deployment Die `StageName` -Eigenschaft im StageDescription-Eigenschaftstyp ist als veraltet gekennzeichnet. AWS::ApiGateway::Method Verwenden Sie die `OperationName` -Eigenschaft, um einer API Gateway-Methode einen Anzeigenamen zuzuweisen. Verwenden Sie die `RequestValidatorId` -Eigenschaft, um einer Methode eine Anforderungsvalidierung zuzuordnen. AWS::AutoScaling::AutoScalingGroup Verwenden Sie die `LifecycleHookSpecificationList` -Eigenschaft, um Aktionen anzugeben, die ausgeführt werden sollen, wenn Auto Scaling Instances startet oder beendet. AWS::CloudFront::Distribution Verwenden Sie die `Tags` Eigenschaft, um eine beliebige Gruppe von Tags (Schlüssel-Wert-Paare) anzugeben, die einer Verteilung zugeordnet werden sollen. CloudFront Verwenden Sie in den DefaultCacheBehaviorEigenschaftstypen CacheBehaviorund die `LambdaFunctionAssociations` Eigenschaft, um Lambda-Funktionszuordnungen für eine CloudFront Verteilung anzugeben. Verwenden Sie im CustomOriginConfig-Eigenschaftstyp die `OriginKeepaliveTimeout` -Eigensch	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		aft, um ein benutzerdefiniertes Keepalive-Timeout anzugeben, und verwenden Sie die `OriginReadTimeout` -Eigenschaft, um ein Timeout für das Lesen des benutzerdefinierten Ursprungs anzugeben. Verwenden Sie im DistributionConfigEigenschaftstyp die `IPV6Enabled` Eigenschaft, um anzugeben , ob CloudFront auf IPv6 DNS-Anfragen eine IPv6 Adresse für Ihre Verteilung antwortet. AWS::CodeDeploy::DeploymentGroup Verwenden Sie im LoadBalancerInfo-Eigenschaftstyp die `TargetGroupInfoList` -Eigenschaft, um Informationen über eine Zielgruppe in Elastic Load Balancing zur Nutzung in einer Bereitstellung anzugeben. AWS::EC2::SecurityGroup, AWS::EC2::SecurityGroupEgress und AWS::EC2::SecurityGroupIngress Verwenden Sie die `Description` -Eigenschaft, um die Beschreibung einer Sicherheitsgruppenregel festzulegen. AWS::EC2::Subnet Die `Ipv6CidrBlock` -Eigenschaft unterstützt jetzt `No interruption` -Aktualisierungen. AWS::EC2::VPNGateway Verwenden Sie die `AmazonSideAsn` -Eigenschaft, um eine private Autonomous System Number (ASN) für die Amazon-Seite einer BGP-Sitzung anzugeben.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::EC2::VPNConnection Verwenden Sie die `VpnTunnelOptionsSpecifications` -Eigenschaft, um Tunneloptionen für eine VPN-Verbindung zu konfigurieren. AWS::ElasticBeanstalk::ConfigurationTemplate und AWS::ElasticBeanstalk::Environment Verwenden Sie in den Eigenschaftstypen ConfigurationOptionSetting und OptionSetting die `ResourceName` -Eigenschaft, um einen Ressourcennamen für eine zeitbasierte Skalierungskonfigurationsoption anzugeben. AWS::EMR::Cluster Verwenden Sie die `CustomAmiId` -Eigenschaft, um ein benutzerdefiniertes Amazon Linux-AMI für ein Cluster anzugeben. AWS::KinesisFirehose::DeliveryStream Verwenden Sie das `Arn`-Attribut mit der `Fn::GetAtt` -Funktion, um den Amazon-Ressourcennamen (ARN) des Bereitstellungs-Streams abzurufen. AWS::KMS::Key Verwenden Sie die `Tags`-Eigenschaft, um einen beliebigen Satz von Tags (Schlüssel-Wert-Paaren) anzugeben, die einen vom Kunden verwalteten Schlüssel zugeordnet sind. AWS::OpsWorks::Layer und AWS::OpsWorks::Stack Verwenden Sie die `Tags` Eigenschaft, um einen beliebigen Satz von Tags (Schlüssel-Wert-Paare)	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		anzugeben, die einer OpsWorks Ebene oder einem Stapel zugeordnet werden sollen. AWS::RDS::OptionGroup Verwenden Sie im OptionConfiguration-Eigenschaftstyp die `OptionVersion` -Eigenschaft, um eine Version für die Option anzugeben. AWS::S3::Bucket Verwenden Sie die `AnalyticsConfigurations` -Eigenschaft zum Konfigurieren eines Analysefilters für einen Amazon S3-Bucket.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	24. Oktober 2017	AWS::Glue::Classifier Verwenden Sie die `AWS::Glue::Classifier` -Ressource zum Erstellen eines AWS Glue -Classifiers. AWS::Glue::Connection Verwenden Sie die `AWS::Glue::Connection` Ressource, um eine AWS Glue Verbindung zu einer Datenquelle anzugeben. AWS::Glue::Crawler Mithilfe der `AWS::Glue::Crawler` -Ressource können Sie einen AWS Glue -Crawler festlegen. AWS::Glue::Database Verwenden Sie die `AWS::Glue::Database` -Ressource zum Erstellen einer AWS Glue -Datenbank. AWS::Glue::DevEndpoint Verwenden Sie die `AWS::Glue::DevEndpoint` -Ressource, um einen Entwicklungs-Endpunkt anzugeben, mit dem ETL-Skripts remote debuggt werden. AWS::Glue::Job Verwenden Sie die `AWS::Glue::Job` -Ressource, um einen AWS Glue -Job im Datenkatalog anzugeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::Glue::Partition Verwenden Sie die `AWS::Glue::Partition` -Ressource, um eine AWS Glue -Partition zu erstellen, die für einen Slice der Tabellendaten steht. AWS::Glue::Table Verwenden Sie die `AWS::Glue::Table` -Ressource zum Erstellen einer AWS Glue -Tabelle. AWS::Glue::Trigger Verwenden Sie die `AWS::Glue::Trigger` Ressource, um Trigger anzugeben, die AWS Glue Jobs ausführen.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	11. Oktober 2017	AWS::SSM::MaintenanceWindow Verwenden Sie die `AWS::SSM::MaintenanceWindow` Ressource, um ein AWS Systems Manager Wartungsfenster zu erstellen. AWS::SSM::MaintenanceWindowTarget Verwenden Sie die `AWS::SSM::MaintenanceWindowTarget` -Ressource zum Registrieren eines Ziels bei einem Wartungsfenster. AWS::SSM::MaintenanceWindowTask Verwenden Sie die `AWS::SSM::MaintenanceWindowTask` -Ressource zum Definieren einer Wartungsfenster-Aufgabe. AWS::SSM::PatchBaseline Verwenden Sie die `AWS::SSM::PatchBaseline` -Ressource zum Definieren einer Systems Manager-Patch-Baseline.	15.05.2010
Neue Ressource	10. Oktober 2017	AWS::ElasticLoadBalancingV2::ListenerCertificate Verwenden Sie die `AWS::ElasticLoadBalancingV2::ListenerCertificate` -Ressource zum Angeben von Zertifikaten für einen Elastic Load Balancing-Listener.	15.05.2010
Neue Ressource	27. Septembe 2017	AWS::Athena::NamedQuery Verwenden Sie zum Erstellen einer Amazon Athena-Abfrage die `AWS::Athena::NamedQuery` -Ressource.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	27. September 2017	AWS::EC2::NatGateway Verwenden Sie die Eigenschaft `Tags`, um Ressourcen-Tags für ein NAT-Gateway anzugeben. AWS::ElasticBeanstalk::Application Verwenden Sie die Eigenschaft `ResourceLifecycleConfig`, um die Lebenszykluseinstellungen für Ressourcen zu definieren, die zur Anwendung gehören, sowie die Servicerolle, die Elastic Beanstalk unterstellt, um die Lebenszykluseinstellungen anzuwenden. AWS::ElasticBeanstalk::ConfigurationTemplate und AWS::ElasticBeanstalk::Environment Verwenden Sie die Eigenschaft `PlatformArn`, um eine benutzerdefinierte Plattform für Elastic Beanstalk anzugeben. AWS::ElasticLoadBalancingV2::TargetGroup Geben Sie im TargetDescriptionEigenschaftstyp mithilfe der `AvailabilityZone` Eigenschaft die Availability Zone an, in der die IP-Adresse registriert werden soll. AWS::Events::Rule Verwenden Sie im Eigenschaftstyp Target die folgenden Eigenschaften für die Eingabeumwandlung von Ereignissen und zum Festlegen von Amazon ECS-Aufgabe und Kinesis-Stream-Zielen. • `EcsParameters` • `InputTransformer` • `KinesisParameters`	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		• `RunCommandParameters` AWS::KinesisFirehose::DeliveryStream Verwenden Sie die `DeliveryStreamType` Eigenschaft, um den Streamtyp anzugeben, und die `KinesisStreamSourceConfiguration` Eigenschaft, um den Stream und die Rolle ARNs für einen Kinesis-Stream anzugeben, der als Quelle für einen Lieferstream verwendet wird. AWS::RDS::DBInstance Wenn Sie für die Eigenschaft `Engine` den Wert `oracle-se` oder `oracle-se1` angegeben haben, können Sie auf `oracle-se2` aktualisieren, ohne dass die Datenbank-Instance ersetzt wird. AWS::S3::Bucket Verwenden Sie die Eigenschaft `AccelerateConfiguration`, um den Transfer Acceleration-Status für einen Amazon S3-Bucket zu konfigurieren.	
Beendigungsschutz für Stacks hinzugefügt.	26. September 2017	Das Aktivieren des Beendigungsschutzes für einen Stack verhindert, dass der Stack versehentlich gelöscht wird. Ein Benutzer kann einen Stack mit aktiviertem Beendigungsschutz nicht löschen. Weitere Informationen finden Sie unter Schützen eines Stacks vor dem Löschen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Der Standardwert von `umask` wurde ab Version 1.4-22 geändert	14. Septembe 2017	Der `umask`-Standardparameterwert für die Konfigurationsdatei cfn-hup.conf lautet jetzt `022`. Weitere Informationen finden Sie unter cfn-hup.	
Aktualisierte Ressourcen	7. September 2017	AWS::ElasticLoadBalancingV2::LoadBalancer Verwenden Sie die `SubnetMappings` Eigenschaft, um die IDs Subnetze anzugeben, die an den Load Balancer angeschlossen werden sollen. Verwenden Sie die `Type` Eigenschaft, um den Typ des zu erstellenden Load Balancers anzugeben. AWS::ElasticLoadBalancingV2::TargetGroup Verwenden Sie die `TargetType` -Eigenschaft, um den Registrierungstyp der Ziele in dieser Zielgruppe anzugeben.	15.05.2010
Rollback-Trigger wurden der API hinzugefügt CloudFormation	31. August 2017	Rollback-Trigger ermöglichen es Ihnen, den Status Ihrer Anwendung während der Stack-Erstellung und Aktualisierung zu CloudFormation überwachen und diesen Vorgang rückgängig zu machen, wenn die Anwendung den Schwellenwert eines der von Ihnen angegebenen Alarme überschreitet. Weitere Informationen finden Sie unter RollbackConfiguration in der AWS CloudFormation -API-Referenz.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue `umask` Parameter für die cfn-hup.conf-Datei	31. August 2017	Verwenden Sie den Parameter `umask` in der Konfigurationsdatei cfn-hup.conf, um Dateiberechtigungen festzulegen, die vom Daemon cfn-hup genutzt werden (Version 1.4-21). Weitere Informationen finden Sie unter cfn-hup.	
Aktualisierte Ressourcen für die Unterstützung des VPC-Sizings	29. August 2017	AWS::EC2::VPCCidrBlock Verwenden Sie die `CidrBlock` Eigenschaft, um einen IPv4 CIDR-Block einer VPC zuzuordnen. AWS::EC2::VPC Verwenden Sie das `CidrBlockAssociations` Attribut mit der `Fn::GetAtt` Funktion, um eine Liste der IPv4 CIDR-Blockzuordnungen abzurufen, die der VPC IDs zugeordnet sind.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	23. August 2017	AWS::S3::Bucket Verwenden Sie im Rule Eigenschaftstyp die `TagFilters` -Eigenschaft , um Tags anzugeben , die für die Identifizierung einer Untergruppe von Objekten für einen Amazon S3-Bucket verwendet werden. Verwenden Sie die `MetricsConfiguration` Eigenschaft, um eine Metrikkonfiguration für die CloudWatch Anforderungsmetriken aus einem Amazon S3 S3-Bucket anzugeben. AWS::IoT::TopicRule Verwenden Sie im Action Eigenschaftstyp die `DynamoDBv2Action` -Eigenschaft, um eine AWS IoT -Aktion zu beschreiben, die Daten in eine DynamoDB-Tabelle schreibt. Im Action-Eigenschaftstyp unterstützt die Eigenschaft `DynamoDBAction` jetzt die Eigenschaften `HashKeyType` und `RangeKeyType` . AWS::Lambda::Permission Verwenden Sie die `EventSourceToken` -Eigenschaft, um eindeutiges Token anzugeben, das von dem Prinzipal bereitgestellt werden muss, der die Funktion aufruft.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Pseudoparameter	23. August 2017	Verwenden Sie die `AWS::Partition` -Pseudo-Parameter, um die Partition zurückzugeben, in der sich eine Ressource befindet. Verwenden Sie den `AWS::URLSuffix` -Pseudo-Parameter, um das Suffix für eine Domain zurückzugeben. Weitere Informationen finden Sie unter Pseudoparameter-Referenz.	15.05.2010
Neue Ressourcen für den DAX-Support	22. August 2017	AWS::DAX::Cluster Erstellen Sie mithilfe der `AWS::DAX::Cluster` -Ressource einen DAX-Cluster zur Verwendung mit Amazon DynamoDB. AWS::DAX::ParameterGroup Verwenden Sie die `AWS::DAX::ParameterGroup` -Ressource , um eine Parametergruppe zur Verwendung mit Amazon DynamoDB zu erstellen. AWS::DAX::SubnetGroup Verwenden Sie die `AWS::DAX::SubnetGroup` -Ressource, um eine Subnetzgruppe zur Verwendung mit DAX (DynamoDB Accelerator) zu erstellen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	18. August 2017	AWS::ApiGateway::DocumentationPart und AWS::ApiGateway::DocumentationPart Verwenden Sie die `AWS::ApiGateway::DocumentationPart` - und `AWS::ApiGateway::DocumentationVersion` -Ressourcen, um eine Dokumentation für Ihre API Gateway API zu erstellen. AWS::ApiGateway::GatewayResponse Verwenden Sie die `AWS::ApiGateway::GatewayResponse` -Ressource, um eine benutzerdefinierte Antwort für Ihre API Gateway API zu erstellen. AWS::ApiGateway::RequestValidator Verwenden Sie die `AWS::ApiGateway::RequestValidator` -Ressource, um Validierungsregeln für eingehende Anforderungen an Ihre API Gateway API einzurichten. AWS::EC2::NetworkInterfacePermission Verwenden Sie die `AWS::EC2::NetworkInterfacePermission` Ressource, um einem AWS Konto die Berechtigung für eine Netzwerkschnittstelle zu erteilen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	18. August 2017	AWS::ApiGateway::Stage Verwenden Sie die `DocumentationVersion` Eigenschaft, um einen versionierten Snapshot der API-Dokumentation anzugeben. AWS::AutoScaling::ScalingPolicy Verwenden Sie die `TargetTrackingConfiguration` -Eigenschaft, um eine Auto Scaling-Ziel-Tracking-Skalierungsrichtlinienkonfiguration anzugeben. AWS::CloudTrail::Trail Verwenden Sie die `EventSelectors` -Eigenschaft, um Amazon S3 Data Events zu unterstützen. AWS::CodeDeploy::DeploymentGroup Verwenden Sie die `LoadBalancerInfo` - und `DeploymentStyle` -Eigenschaften, um einen Elastic Load Balancing Load Balancer für eine In-Place-Bereitstellung anzugeben. Verwenden Sie die `AutoRollbackConfiguration` -Eigenschaft, um das automatische Zurücksetzen für die Bereitstellung zu konfigurieren. AWS::EC2::SpotFleet Verwenden Sie im SpotFleetRequestConfigData-Eigenschaftstyp die `ReplaceUnhealthyInstances` -Eigenschaft, um anzugeben, ob die Spot-Flotte fehlerhafte Instances ersetzen soll, und die `Type`-Eigenschaft, um den Anforderungstyp anzugeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::EC2::Subnet Verwenden Sie die `Ipv6CidrBlock` Eigenschaften `AssignIpv6AddressOnCreation` und, um ein Subnetz mit einem IPv6 CIDR-Block zu erstellen. AWS::KinesisFirehose::DeliveryStream Verwenden Sie die `ExtendedS3DestinationConfiguration` -Eigenschaft, um in Amazon S3 ein Ziel zu konfigurieren. Verwenden Sie die `ProcessingConfiguration` -Untereigenschaft in jeder Ziel-Konfiguration, um Lambda-Funktionen aufzurufen, die eingehende Quelldaten umwandeln und die umgewandelten Daten an die Ziele liefern. AWS::RDS::DBCluster und AWS::RDS::DBInstance Die Standardeinstellung `DeletionPolicy` gilt jetzt `Snapshot` für `AWS::RDS::DBCluster` Ressourcen und für `AWS::RDS::DBInstance` Ressourcen, die die Eigenschaft nicht angeben. `DBClusterIdentifier` Weitere Informationen finden Sie unter DeletionPolicyAttribut. AWS::S3::Bucket Verwenden Sie im Rule-Eigenschaftstyp die `AbortIncompleteMultipartUpload` -Eigenschaft, um eine Lebenszyklusregel anzugeben, die unvollständige mehrteilige Uploads in ein Amazon S3-Bucket abbricht.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::SQS::Queue Verwenden Sie die `KmsMasterKeyId` - und `KmsDataKeyReusePeriodSeconds` -Eigenschaften, eine serverseitige Verschlüsselung für Amazon SQS zu konfigurieren. Hinzufügen des `Arn`-Attributs zur intrinsischen Funktion `Fn::GetAtt` für die folgenden Ressourcen: • AWS::CloudTrail::Trail. Außerdem wurde `SnsTopicArn` hinzugefügt. • AWS::CloudWatch::Alarm • AWS::DynamoDB::Table • AWS::ECS::Cluster • AWS::IoT::Policy • AWS::IoT::TopicRule • AWS::Logs::Destination	
Support für Stack-Tags in CodePipeline Artefakten	18. August 2017	Sie können jetzt Tags für Stacks in Vorlagenkonfigurationsdateien angeben, um sie als Artefakte für CodePipeline Pipelines zu verwenden. Angegebene Tags werden auf Stacks angewendet, die mit der Vorlagen-Konfigurationsdatei erstellt wurden. Weitere Informationen finden Sie unter CloudFormation Artefakte.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Erstellen verschlüsselter Dateisysteme	14. August 2017	AWS::EFS::FileSystem Verwenden Sie die `Encrypted` -Eigenschaft zum Verschlüsseln eines Amazon EFS-Dateisystems während der Erstellung. Verwenden Sie die `KmsKeyId`-Eigenschaft, um optional einen benutzerdefinierten kundenverwalteten Schlüssel anzugeben, um das verschlüsselten Dateisystem zu schützen.	15.05.2010
Neue Ressourcen für den AWS Batch Support	8. August 2017	AWS::Batch::ComputeEnvironment Verwenden Sie die `AWS::Batch::ComputeEnvironment` Ressource, um Ihre AWS Batch Rechenumgebung zu definieren. AWS::Batch::JobDefinition Verwenden Sie die `AWS::Batch::JobDefinition` Ressource, um die Parameter für eine AWS Batch Jobdefinition anzugeben. AWS::Batch::JobQueue Verwenden Sie die `AWS::Batch::JobQueue` Ressource, um Ihre AWS Batch Job-Warteschlange zu definieren.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen für den Support von Amazon Managed Service für Apache Flink	28. Juli 2017	AWS::KinesisAnalytics::Application Verwenden Sie die Ressource `AWS::KinesisAnalytics::Application`, um eine Amazon-Managed-Service-für-Apache-Flink-Anwendung zu erstellen. AWS::KinesisAnalytics::ApplicationOutput Verwenden Sie die Ressource `AWS::KinesisAnalytics::ApplicationOutput`, um Ihrer Amazon-Managed-Service-für-Apache-Flink-Anwendung ein externes Ziel hinzuzufügen. AWS::KinesisAnalytics::ApplicationReferenceDataSource Verwenden Sie die Ressource `AWS::KinesisAnalytics::ApplicationReferenceDataSource`, um einer bestehenden Amazon-Managed-Service-für-Apache-Flink-Anwendung eine Referenzdatenquelle hinzuzufügen.	15.05.2010
Verwenden Sie StackSets es, um Stacks über Konten und Regionen hinweg zentral zu verwalten	25. Juli 2017	StackSets ermöglicht es Ihnen, Stacks für mehrere Konten und Regionen in einem einzigen Vorgang zu erstellen, zu aktualisieren oder zu löschen. Mithilfe eines Administratorkontos definieren und verwalten Sie eine CloudFormation Vorlage und verwenden die Vorlage als Grundlage für die Bereitstellung von Stacks für ausgewählte Zielkonten in bestimmten Regionen. Weitere Informationen finden Sie unter Konten- und regionsübergreifende Stacks verwalten mit. StackSets	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Anzeigen von Stack-Ereignissen durch Client-Anforderungs-Token	14. Juli 2017	In der Konsole zeigen Stack-Operationen das Client-Anforderungs-Token auf der Registerkarte Ereignisse. Allen Ereignissen, die von einer bestimmten Stack-Operation ausgelöst werden, wird dasselbe Client-Anfrage-Token zugewiesen, das Sie verwenden können, um Operationen zu verfolgen. Weitere Informationen finden Sie unter CloudFormation Stack-Daten und -Ressourcen anzeigen auf der AWS Management Console und StackEventin der AWS CloudFormation API-Referenz.	15.05.2010
Verwenden von Quick-Links	14. Juli 2017	Verwenden Sie Quick-Links, um Stacks einzurichten und schnell zu starten. Sie können die Vorlage-URL, den Stack-Namen und die Vorlagenparameter angeben, um eine Create Stack-Assistentenseite mit Daten zu füllen. Weitere Informationen finden Sie unter Schnellverknüpfungen für Stacks erstellen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen für den AWS Database Migration Service Support	12. Juli 2017	AWS::DMS::Certificate Verwenden Sie die `AWS::DMS::Certificate` Ressource, um ein SSL-Zertifikat zu erstellen, das Verbindungen zwischen AWS DMS Endpunkten und der Replikationsinstanz verschlüsselt. AWS::DMS::Endpoint Verwenden Sie die `AWS::DMS::Endpoint` -Ressource zum Erstellen eines AWS DMS -Endpunkts. AWS::DMS::EventSubscription Verwenden Sie die `AWS::DMS::EventSubscription` Ressource, um Benachrichtigungen über AWS DMS Ereignisse über den Amazon Simple Notification Service zu erhalten. AWS::DMS::ReplicationInstance Verwenden Sie die `AWS::DMS::ReplicationInstance` Ressource, um eine AWS DMS Replikationsinstanz zu erstellen. AWS::DMS::ReplicationSubnetGroup Verwenden Sie die `AWS::DMS::ReplicationSubnetGroup` Ressource, um eine AWS DMS Replikationssubnetzgruppe zu erstellen. AWS::DMS::ReplicationTask Verwenden Sie die `AWS::DMS::ReplicationTask` Ressource, um eine AWS DMS Replikationsaufgabe zu erstellen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	5. Juli 2017	AWS::CloudWatch::Dashboard Verwenden Sie die `AWS::CloudWatch::Dashboard` Ressource, um ein benutzerdefiniertes CloudWatch Dashboard für Ihre CloudWatch Konsole anzugeben. AWS::ApiGateway::DomainName Verwenden Sie die `AWS::ApiGateway::DomainName` -Ressource, um einen benutzerdefinierten URL-Anzeigenamen für Ihre API anzugeben, der API Amazon API Gateway bereitgestellt wird. AWS::EC2::EgressOnlyInternetGateway Verwenden Sie die `AWS::EC2::EgressOnlyInternetGateway` -Ressource, um ein Internet-Gateway nur für ausgehende Datenverkehr für Ihre VPC zu erstellen. InstanceFleetConfig Verwenden Sie die `InstanceFleetConfig` -Ressource, um eine Spot-Instance-Flotte für einen Amazon-EMR-Cluster zu konfigurieren.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	5. Juli 2017	AWS::ApiGateway::RestApi Verwenden Sie die `BinaryMediaTypes` -Eigenschaft, um unterstützte binäre Medientypen anzugeben. AWS::ApplicationAutoScaling::ScalingPolicy Verwenden Sie die `TargetTrackingScalingPolicyConfiguration` -Eigenschaft, um eine Skalierungsrichtlinien-Konfiguration für die Ziel-Nachverfolgung anzugeben. AWS::CloudTrail::Trail Verwenden Sie die `TrailName` Eigenschaft, um einen benutzerdefinierten Namen für eine AWS CloudTrail Ressource anzugeben. Verwenden Sie die `Tags`-Eigenschaft, um Ressourcen-Tags anzugeben. AWS::CodeDeploy::DeploymentGroup Verwenden Sie die `AlarmConfiguration` -Eigenschaft, um Alarme für die Bereitstellungsgruppe zu konfigurieren. Verwenden Sie die `TriggerConfigurations` -Eigenschaft, um Benachrichtigungsauslöser für die Bereitstellungsgruppe zu konfigurieren. AWS::EMR::Cluster Verwenden Sie die `CoreInstanceFleet` Eigenschaft und die `MasterInstanceFleet` Eigenschaft im Eigenschaftstyp, um die JobFlowInstancesConfigSpot-Instance-Flotte für einen Amazon EMR-Cluster zu konfigurieren.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::DynamoDB::Table Verwenden Sie die `TimeToLiveSpecification` -Eigenschaft, um die TTL-Einstellungen (Time to Live) für eine Amazon DynamoDB-Tabelle anzugeben. Verwenden Sie die `Tags`-Eigenschaft, um Ressourcen-Tags für eine DynamoDB-Tabelle anzugeben. AWS::EC2::Instance Die `IamInstanceProfile` -Eigenschaft unterstützt jetzt `No interruption` -Aktualisierungen. AWS::EC2::Route Verwenden Sie die `EgressOnlyInternetGatewayId` Eigenschaft, um ein Internet-Gateway für ausgehende Verbindungen für eine Route anzugeben. EC2 AWS::Kinesis::Stream Verwenden Sie die `RetentionPeriodHours` -Eigenschaft, um anzugeben, wie viele Stunden lang die in Shards gespeicherten Datensätze zur Verfügung stehen. AWS::RDS::DBCluster Verwenden Sie die `ReplicationSourceIdentifier` -Eigenschaft, um ein DB-Cluster als Read Replica eines anderen DB-Clusters oder einer Amazon RDS MySQL-DB-Instance zu erstellen.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::Redshift::Cluster Verwenden Sie die `LoggingProperties` -Eigenschaft, um Audit-Protokolldateien zu erstellen und in Amazon S3 zu speichern.	
Neue Ressourcen	6. Juni 2017	AWS::EMR::SecurityConfiguration Verwenden Sie die `AWS::EMR::SecurityConfiguration` -Ressource zum Erstellen einer Sicherheitskonfiguration, die im Service gespeichert wird und angegeben werden kann, wenn ein Cluster erstellt wird.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	6. Juni 2017	AWS::AutoScaling::LifecycleHook Die Eigenschaften `NotificationTargetARN` und `RoleARN` sind jetzt optional. AWS::CloudWatch::Alarm Sie können jetzt die Eigenschaften `EvaluateLowSampleCountPercentile`, `ExtendedStatistic` und `TreatMissingData` verwenden, wenn Sie `AWS::CloudWatch::Alarm`-Ressourcen erstellen. AWS::EC2::SpotFleet CloudFormation unterstützt veränderbare Änderungen an den Eigenschaften der Spot-Flotte. Die folgenden Eigenschaften der `SpotFleetRequestConfigData`-Eigenschaft unterstützen `Replacement`-Aktualisierungen: • `AllocationStrategy` • `IamFleetRole` • `LaunchSpecifications` • `SpotPrice` • `TerminateInstancesWithExpiration` • `ValidFrom` • `ValidUntil` Die folgenden Eigenschaften der `SpotFleetRequestConfigData`-Eigenschaft unterstützen `No interruption`-Aktualisierungen: • `ExcessCapacityTerminationPolicy` • `TargetCapacity`	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::EMR::InstanceGroupConfig CloudFormation unterstützt jetzt Auto Scaling für Amazon EMR-Task-Instance-Gruppen. AWS::Events::Rule Die `RoleArn`-Eigenschaft ist für die `Rule`-Ressource als veraltet gekennzeichnet. Verwenden Sie die `RoleArn`-Eigenschaft des `Target`-Eigenschaftentyps, um die IAM-Rolle für die Verwendung für ein Ziel anzugeben. AWS::Kinesis::Stream Die `ShardCount` -Eigenschaft unterstützt jetzt No `interruption` -Aktualisierungen. AWS::Lambda::Function Verwenden Sie die `TracingConfig` -Eigenschaft zum Konfigurieren von Protokollierungseinstellungen für Lambda Funktionen. AWS::Redshift::Cluster, AWS::Redshift::ClusterParameterGroup, AWS::Redshift::ClusterSecurityGroup, und AWS::Redshift::ClusterSubnetGroup Verwenden Sie die `Tags`-Eigenschaft, um Ressourcen-Tags anzugeben. AWS::RDS::DBCluster Das `ReadEndpoint.Address` -Attribut wurde zur intrinsischen `Fn::GetAtt` -Funktion hinzugefügt.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::S3::Bucket Das `Arn`-Attribut wurde zur intrinsischen `Fn::GetAtt` -Funktion hinzugefügt.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	11. Mai 2017	Die folgenden neuen Ressourcen unterstützen die Verwendung AWS WAF mit Elastic Load Balancing (ELB) Application Load Balancers. AWS::WAFRegional::ByteMatchSet Verwenden Sie die `AWS::WAFRegional::ByteMatchSet` -Ressource zum Identifizieren eines Teils einer Webanforderung, die Sie prüfen möchten. AWS::WAFRegional::IPSet Verwenden Sie die `AWS::WAFRegional::IPSet` -Ressource, um auf Basis der IP-Adressen, von denen die Anforderungen stammen, anzugeben , welche Webanforderungen zugelassen und welche blockiert werden. AWS::WAFRegional::Rule Verwenden Sie die `AWS::WAFRegional::Rule` -Ressource, um eine Kombination von `IPSet`-, `ByteMatchSet` - und `SqlInjectionMatchSet` -Objekten anzugeben, welche die Webanforderungen zum Zulassen, Blockieren oder Zählen identifizieren. AWS::WAFRegional::SizeConstraintSet Verwenden Sie die `AWS::WAFRegional::SizeConstraintSet` -Ressource, um eine Größenbeschränkung zum Prüfen der Größe einer Webanforderung sowie die zu prüfenden Teile der Anforderung anzugeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::WAFRegional::SqlInjectionMatchSet Verwenden Sie die `AWS::WAFRegional::SqlInjectionMatchSet` -Ressource, um Anforderungen zuzulassen, zu blockieren oder zu zählen, die schädlichen SQL-Code in einem bestimmten Teil der Webanforderungen enthalten. AWS::WAFRegional::WebACL Verwenden Sie die `AWS::WAFRegional::WebACL` -Ressource zum Identifizieren der Webanforderungen, die Sie zulassen, blockieren oder zählen möchten. AWS::WAFRegional::WebACLAssociation Verwenden Sie die `AWS::WAFRegional::WebACLAssociation` -Ressource, um eine Web-Zugriffskontrollgruppe (ACL) mit einer Ressource zu verknüpfen. AWS::WAFRegional::XssMatchSet Verwenden die `AWS::WAFRegional::XssMatchSet` -Ressource, um die Teile von AWS WAF -Web-Anforderungen anzugeben, die auf Cross-Site-Scripting-Angriffe hin überprüfen soll, sowie den Namen des zu prüfenden Headers.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	28. April 2017	AWS::Cognito::IdentityPool Verwenden Sie die `AWS::Cognito::IdentityPool` -Ressource zum Erstellen eines Amazon Cognito-Identitäten-Pools. AWS::Cognito::IdentityPoolRoleAttachment Verwenden Sie die `AWS::Cognito::IdentityPoolRoleAttachment` -Ressource zum Verwalten der Rollenkonfiguration für einen Amazon Cognito Identitäten-Pool. AWS::Cognito::UserPool Verwenden Sie die `AWS::Cognito::UserPool` -Ressource zum Erstellen eines Amazon Cognito-Benutzerpools. AWS::Cognito::UserPoolClient Verwenden Sie `AWS::Cognito::UserPoolClient` -Ressource zum Erstellen eines Benutzerpool-Clients. AWS::Cognito::UserPoolGroup Verwenden Sie die `AWS::Cognito::UserPoolGroup` -Ressource zum Erstellen einer Benutzergruppe in einem Amazon Cognito-Benutzerpool. AWS::Cognito::UserPoolUser Verwenden Sie die `AWS::Cognito::UserPoolUser` -Ressource zum Erstellen eines Amazon Cognito-Benutzerpool-Benutzers.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::Cognito::UserPoolUserToGroupAttachment Verwenden Sie die `AWS::Cognito::UserPoolUserToGroupAttachment` -Ressource, um einen Benutzer an eine Amazon Cognito-Benutzerpool-Gruppe anzuhängen.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	28. April 2017	SourceDetails Verwenden Sie die `MaximumExecutionFrequency` -Untereigenschaft der `AWS::Config::ConfigRule` -Ressource, um Bewertungen für eine benutzerdefinierte Regel mithilfe eines regelmäßigen Auslösers auszuführen. AWS::EC2::Volume Wir unterstützen jetzt Elastic-Volumes für Amazon Elastic Block Store (Amazon EBS) in CloudFormation. Wir unterstützen jetzt `No interruption` -Aktualisierungen für drei Eigenschaften: `VolumeType` , `Size` und `Iops`. AWS::EC2::SecurityGroup Verwenden Sie die `GroupName` Eigenschaft, um einen Namen für Ihre EC2 Amazon-Sicherheitsgruppe anzugeben. AWS::ECS::Service Es gibt drei neue Eigenschaften für `AWS::ECS::Service` : `PlacementConstraints` , `PlacementStrategies` und `ServiceName` . AWS::ECS::TaskDefinition Verwenden Sie die `PlacementConstraints` -Eigenschaft zum Definieren von Platzierungseinschränkungen für Aufgaben im Service. AWS::ElastiCache::ReplicationGroup Das `ConfigurationEndPoint.Address` -Attribut und das `ConfigurationEndPo`	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		`int.Port` -Attribut wurden zur intrinsischen `Fn::GetAtt` -Funktion hinzugefügt. AWS::ElasticLoadBalancingV2::LoadBalancer Verwenden Sie die `IpAddressType` -Eigenschaft, um den Typ der IP-Adressen anzugeben, die von den Load Balancer-Subnetzen verwendet werden. AWS::EMR::Cluster CloudFormation unterstützt jetzt Auto Scaling für Amazon EMR-Cluster. AWS::IAM::ManagedPolicy Verwenden Sie die `ManagedPolicyName` -Eigenschaft, um einen benutzerdefinierten Namen für Ihre verwaltete IAM-Richtlinie anzugeben. AWS::Lambda::Function Verwenden Sie die `Tags`-Eigenschaft zum Hinzufügen von Tags zu Ihrer Lambda-Funktion. AWS::OpsWorks::Instance Die folgenden Attribute wurden zur intrinsischen `Fn::GetAtt` -Funktion hinzugefügt: `AvailabilityZone` , `PrivateDnsName` , `PrivateIp` und `PublicDnsName` . AWS::OpsWorks::UserProfile Verwenden Sie die `SshUsername` -Eigenschaft, um den SSH-Namen eines Benutzers anzugeben. Das `SshUsername` -Attribut wurde zur intrinsischen `Fn::GetAtt` -Funktion hinzugefügt.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::Redshift::Cluster Verwenden Sie die `IamRoles` Eigenschaft, um eine Liste mit einer oder mehreren AWS Identity and Access Management Rollen bereitzustellen, die der Amazon Redshift Redshift-Cluster für den Zugriff auf andere AWS Services verwenden kann.	
Bearbeiten Sie Vorlagen in YAML und JSON mit Designer AWS CloudFormation	6. April 2017	Wenn Sie CloudFormation Vorlagen mit Designer erstellen, können Sie Ihre Vorlage jetzt sowohl in YAML als auch in JSON im integrierten Editor bearbeiten. Sie können auch JSON-Vorlagen in YAML konvertieren und umgekehrt, je nach Ihrer bevorzugten Sprache für die Vorlagenerstellung. Weitere Informationen finden Sie unter Was ist CloudFormation Designer? .	15.05.2010
Neue Ressource	6. April 2017	AWS::SSM::Parameter Verwenden Sie die `AWS::SSM::Parameter` -Ressource zum Erstellen eines SSM-Parameters im Parameter Store.	15.05.2010
`AWS::Include` -Transformation	28. März 2017	Verwenden Sie die `AWS::Include` -Transformation, um wiederverwendbare, in einem Amazon S3-Bucket gespeicherte Codeaussschnitte wiederzuverwenden. Weitere Informationen finden Sie unter AWS::Include-Transformation.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Verknüpfen Ihrer Amazon VPC mit einem anderen Konto	28. März 2017	Sie können CloudFormation jetzt Ihre Amazon VPC mit einer VPC in einem anderen AWS Konto verbinden. Weitere Informationen finden Sie unter Peer mit einer Amazon VPC in einem anderen AWS Konto.	15.05.2010
Neue Ressource	28. März 2017	AWS::ApiGateway::UsagePlanKey Verwenden Sie die `AWS::ApiGateway::UsagePlanKey` -Ressource, um einen Nutzungsplanschlüssel zu verknüpfen und festzulegen, auf welche Benutzer der Nutzungsplan angewendet wird.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	28. März 2017	AWS::EC2::VPCPeeringConnection Verwenden Sie die `PeerOwnerId` -Eigenschaft und die `PeerRoleArn` -Eigenschaft zum Verknüpfen mit einer VPC in einem anderen AWS -Konto. Weitere Informationen finden Sie unter Peer mit einer Amazon VPC in einem anderen AWS Konto. AWS::IAM::InstanceProfile Verwenden Sie die `InstanceProfileName` -Eigenschaft zum Konfigurieren eines Instance-Profils. AWS::Lambda::Function Verwenden Sie die `DeadLetterConfig` -Eigenschaft, um zu konfigurieren, wie AWS Lambda Ereignisse handhabt, die nicht verarbeitet werden können. Node.js v0.10 wird für die `Runtime`-Eigenschaft nicht mehr unterstützt. AWS::Route53::HealthCheck Es gibt sieben neue Ressourcen für die HealthCheckConfig `HealthCheckConfig` -Untereigenschaftstypen: `AlarmIdentifier` , `ChildHealthChecks` , `EnableSNI` , `HealthThreshold` , `InsufficientDataHealthStatus` , `Inverted` und `MeasureLatency` . AWS::SQS::Queue Verwenden Sie die `FifoQueue` Eigenschaften `ContentBasedDeduplication` und, um First-In-First-Out (FIFO) Amazon Simple Queue Service-Warteschlangen zu erstellen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::S3::Bucket Sie können jetzt IPv6 Domainnamen für Ihre Amazon S3 S3-Buckets angeben.	
Neue Ressourcen	10. Februar 2017	AWS::StepFunctions::Activity Verwenden Sie die `AWS::StepFunctions::Activity` Ressource, um eine AWS Step Functions Aktivität zu erstellen. AWS::StepFunctions::StateMachine Verwenden Sie die `AWS::StepFunctions::StateMachine` -Ressource zum Erstellen eines Step Functions-Zustandsautomaten.	15.05.2010
Neue intrinsische Funktion	17. Januar 2017	Verwenden Sie die `Fn::Split` -Funktion zum Aufteilen einer Zeichenfolge in eine Liste von Zeichenfolgenwerten. Weitere Informationen finden Sie unter Fn::Split.	15.05.2010
Konsolensupport zum Auflisten von Importen	17. Januar 2017	Verwenden Sie die CloudFormation Konsole, um alle Stapel zu sehen, die einen exportierten Ausgabewert importieren. Weitere Informationen finden Sie unter Auflisten von Stacks, die einen exportierten Ausgabewert importieren.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	17. Januar 2017	AWS::AutoScaling::AutoScalingGroup Die `LoadBalancerNames` -Eigenschaft kann aktualisiert werden, ohne die Auto-Scaling-Gruppe zu ersetzen. AWS::ECS::TaskDefinition Die Eigenschaften `NetworkMode` und `MemoryReservation` wurden hinzugefügt. AWS::RDS::DBCluster CloudFormation unterstützt Aktualisierungen der `Tags` Eigenschaft. AWS::RDS::DBInstance Die `Timezone`-Eigenschaft wurde hinzugefügt. FirehoseAction Die `Separator` -Eigenschaft wurde hinzugefügt. AWS::OpsWorks::Instance Das `PublicIp`-Attribut für die intrinsische `Fn::GetAtt` -Funktion wurde hinzugefügt.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	01. Dezember 2016	AWS::CodeBuild::Project Verwenden Sie die `AWS::CodeBuild::Project` Ressource, um ein AWS CodeBuild Projekt zu erstellen, das definiert, wie Ihr Quellcode CodeBuild erstellt wird. AWS::SSM::Association Verwenden Sie die `AWS::SSM::Association` Ressource, um ein Amazon EC2 Systems Manager Manager-Dokument mit EC2 Instances zu verknüpfen. AWS::EC2::SubnetCidrBlock Verwenden Sie die `AWS::EC2::SubnetCidrBlock` Ressource, um einen einzelnen IPv6 CIDR-Block einem Amazon VPC-Subnetz zuzuordnen. AWS::EC2::VPCCidrBlock Verwenden Sie die `AWS::EC2::VPCCidrBlock` Ressource, um einen einzelnen von Amazon bereitgestellten IPv6 CIDR-Block einer Amazon VPC zuzuordnen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Die Ressourcen für den Support wurden aktualisiert IPv6	01. Dezember 2016	AWS::EC2::Instance Die Eigenschaften `Ipv6AddressCount` und `Ipv6Addresses` wurden hinzugefügt. AWS::EC2::NetworkAclEntry Die `Ipv6CidrBlock` -Eigenschaft wurde hinzugefügt. AWS::EC2::NetworkInterface Die Eigenschaften `Ipv6AddressCount` und `Ipv6Addresses` wurden hinzugefügt. AWS::EC2::Route Die `DestinationIpv6CidrBlock` -Eigenschaft wurde hinzugefügt. AWS::EC2::SecurityGroupEgress Die `CidrIpv6`-Eigenschaft wurde hinzugefügt. AWS::EC2::SecurityGroupIngress Die `CidrIpv6`-Eigenschaft wurde hinzugefügt. AWS::EC2::SpotFleet Die Eigenschaften `Ipv6AddressCount` und `Ipv6Addresses` für die Startspezifikations-Netzwerkschnittstellen wurden hinzugefügt. AWS::EC2::Subnet Das `Ipv6CidrBlocks` -Attribut für die `Fn::GetAtt` -Funktion wurde hinzugefügt.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::EC2::VPC Das `Ipv6CidrBlocks` -Attribut für die `Fn::GetAtt` -Funktion wurde hinzugefügt. AWS::SSM::Document Die `DocumentType` -Eigenschaft wurde hinzugefügt.	
Ressourcenspezifikation	22. November 2016	Verwenden Sie die CloudFormation Ressourcenspezifikation, um Tools zu erstellen, mit denen Sie CloudFormation Vorlagen erstellen können. Die Spezifikation ist eine maschinenlesbare, JSON-formatierte Textdatei. Weitere Informationen finden Sie unter CloudFormation Ressourcenspezifikation.	15.05.2010
Neue Ressourcen	22. November 2016	AWS::OpsWorks::UserProfile Verwenden Sie die `AWS::OpsWorks::UserProfile` Ressource, um den SSH-Zugriff für Benutzer zu konfigurieren, die Zugriff auf Instanzen in einem OpsWorks Stack benötigen. AWS::OpsWorks::Volume Verwenden Sie die `AWS::OpsWorks::Volume` Ressource, um ein Amazon Elastic Block Store-Volume bei einem OpsWorks Stack zu registrieren.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	22. November 2016	AWS::OpsWorks::App Die `DataSources` -Eigenschaft wurde hinzugefügt. AWS::OpsWorks::Instance Die Eigenschaften `BlockDeviceMappings` , `AgentVersion` , `ElasticIps` , `Hostname`, `Tenancy` und `Volumes` wurden hinzugefügt. AWS::OpsWorks::Layer Die Eigenschaften `CustomJson` und `VolumeConfigurations` wurden hinzugefügt. AWS::OpsWorks::Stack Die Eigenschaften `ElasticIps` , `EcsClusterArn` , `RdsDbInstances` , `CloneAppIds` , `ClonePermissions` und `SourceStackId` wurden hinzugefügt. AWS::RDS::DBInstance Die `CopyTagsToSnapshot` -Eigenschaft wurde hinzugefügt.	15.05.2010
Auflisten von Importen	22. November 2016	Listet Importe eines exportierten Ausgabewerts auf, um nachzuverfolgen, welche CloudFormation Stapel den Wert importieren. Weitere Informationen finden Sie unter Auflisten von Stacks, die einen exportierten Ausgabewert importieren.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Transformationen	17. November 2016	Geben Sie das AWS Serverless Application Model (AWS SAM) an, das zur Verarbeitung der AWS SAM-Syntax für serverlose CloudFormation Anwendungen verwendet wird. Weitere Informationen finden Sie unter -Transformation.	15.05.2010
Neue Ressource	17. November 2016	AWS::SNS::Subscription Verwenden Sie die `AWS::SNS::Subscription` -Ressource, um einen Endpunkt für ein Amazon Simple Notification Service-Thema zu abonnieren.	15.05.2010
Aktualisierte Ressource	17. November 2016	AWS::Lambda::Function Verwenden Sie die `Environment` Eigenschaft, um Schlüssel-Wert-Paare (Umgebungsvariablen) anzugeben, auf die Ihre AWS Lambda Funktion zugreifen kann. Verwenden Sie die `KmsKeyArn` Eigenschaft, um einen KMS-Schlüssel anzugeben, der zum Verschlüsseln und Entschlüsseln von Umgebungsvariablen AWS Lambda verwendet wird.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue CLI-Befehle	17. November 2016	Hochladen lokaler Artefakte in einen S3-Bucket Verwenden Sie den `package` Befehl, um lokale Artefakte, auf die in einer CloudFormation Vorlage verwiesen wird, in einen S3-Bucket hochzuladen. Schnelles Bereitstellen von Vorlagen mit Transformationen Verwenden Sie den `deploy`-Befehl, um die Änderungssatzaktionen zum Erstellen und Ausführen in einem einzigen Befehl zu kombinieren. Dieser Befehl ist hilfreich, um schnell Stacks mit Transformationen zu erstellen oder zu aktualisieren.	15.05.2010
Aktualisierte Ressource	03. November 2016	AWS::CloudFront::Distribution Verwenden Sie für die DistributionConfigEigenschaft die `HttpVersion` Eigenschaft, um die neueste HTTP-Version anzugeben, die Zuschauer für die Kommunikation mit Amazon verwenden können CloudFront. Verwenden Sie für die ForwardedValues`QueryStringCacheKeys` Eigenschaft die Abfragezeichenfolge-Parameter, CloudFront anhand derer bestimmt wird, welcher Inhalt zwischengespeichert werden soll.	15.05.2010
Auflisten von Stack-Exporten	03. November 2016	Verwenden Sie die CloudFormation Konsole, die API oder, AWS CLI um eine Liste aller exportierten Ausgabewerte für eine Region anzuzeigen. Weitere Informationen finden Sie unter Exportieren von Stack-Ausgabewerten.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Fortlaufende Zustellung mit Stacks	03. November 2016	Wird verwendet AWS CodePipeline , um Workflows für die kontinuierliche Bereitstellung mit CloudFormation Stacks zu erstellen. Weitere Informationen finden Sie unter Continuous Delivery mit CodePipeline.	15.05.2010
Überspringen von Ressourcen während des Rollbacks	03. November 2016	Wenn sich ein Stack in einem `UPDATE_ROLLBACK_FAILED` -Status befindet, verwenden Sie den `ResourcesToSkip` -Parameter für die `ContinueUpdateRollback` -Aktion, um Ressourcen zu überspringen, für die CloudFormation kein Rollback ausführen kann. Weitere Informationen finden Sie im Abschnitt zur Fehlerbehebung unter Update Rollback fehlgeschlagen.	15.05.2010
Verbesserung für Änderungssätze	03. November 2016	Sie können einen neuen Stack mithilfe eines Änderungssatzes erstellen.	15.05.2010
Aktualisierte Ressource	12. Oktober 2016	AWS::ElastiCache::CacheCluster Aktualisieren Sie die `CacheNodeType` -Eigenschaft ohne Ersetzen des Clusters. AWS::ElastiCache::ReplicationGroup Sie können eine (Clustermodus-aktivierte) Redis-Replikationsgruppe mit mehreren Knotengruppen (Shards) erstellen, jede mit einem primären Cluster und Read Replicas. AWS::ElastiCache::SubnetGroup Verwenden Sie die `CacheSubnetGroupName` Eigenschaft, um einen Namen für eine ElastiCache Amazon-Subnetzgruppe anzugeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	06. Oktober 2016	AWS::ApiGateway::UsagePlan Verwenden Sie die `AWS::ApiGateway::UsagePlan` Ressource, um einen Nutzungsplan für das bereitgestellte Amazon API Gateway anzugeben APIs. AWS::CodeCommit::Repository Verwenden Sie die `AWS::CodeCommit::Repository` Ressource, um ein CodeCommit Repository zu erstellen, das von Amazon Web Services gehostet wird.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	06. Oktober 2016	AWS::ApiGateway::Authorizer Verwenden Sie die `ProviderARNs` -Eigenschaft, um Amazon Cognito-Benutzerpools als Amazon API Gateway-API-Genehmiger zu verwenden. AWS::ApiGateway::Deployment Die `StageName` -Eigenschaft ist nicht mehr erforderlich. AWS::ElasticLoadBalancingV2::TargetGroup Verwenden Sie für die `GetAtt` Funktion das `LoadBalancerArns` Attribut, um die Amazon-Ressourcennamen (ARNs) der Load Balancer abzurufen, die den Verkehr an die Zielgruppe weiterleiten. AWS::RDS::DBInstance Verwenden Sie die Eigenschaften `Domain` und `DomainIAMRoleName` , um Windows-Authentifizierung zu verwenden, wenn Benutzer eine Verbindung zu der RDS DB-Instance herstellen. AWS::EC2::SecurityGroupEgress Verwenden Sie die `DestinationPrefixListId` Eigenschaft, um das AWS Servicepräfix eines Amazon VPC-Endpunkts anzugeben.	15.05.2010
Verbesserung der Stack-übergreifenden Referenz	06. Oktober 2016	Verwenden Sie intrinsische Funktionen, um den `Name`-Wert eines Exports anzupassen oder auf einen Wert in der `ImportValue` -Funktion zu verweisen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
CloudFormation Service-Rolle	26. September 2016	Verwenden Sie eine AWS Identity and Access Management (IAM) -Servicerolle für CloudFormation Stack-Operationen. CloudFormation verwendet die Anmeldeinformationen der Rolle, um Stack-Ressourcen in Ihrem Namen aufzurufen. Weitere Informationen finden Sie unter AWS CloudFormation Servicerolle.	15.05.2010
Neues Feature	19. September 2016	Sie können das `Export`-Ausgabefeld und die intrinsische `Fn::ImportValue` -Funktion verwenden, damit ein Stack auf Ressourcenausgaben in einem anderen Stack verweist. Weitere Informationen finden Sie unter Ausgaben und Exemplarische Vorgehensweise: Weitere Informationen finden Sie unter Ressourcenausgaben in einem anderen CloudFormation Stapel. Fn::ImportValue	15.05.2010
YAML-Unterstützung	19. September 2016	Sie können das YAML-Format verwenden, um Vorlagen zu erstellen. CloudFormation In YAML können Sie außerdem z. B. Kommentare zu den Vorlagen hinzufügen oder die Kurzform für intrinsische Funktionen verwenden. Weitere Informationen finden Sie unter CloudFormationVorlagenformat.	15.05.2010
Neue intrinsische Funktion	19. September 2016	Verwenden Sie die `Fn::Sub`-Funktion, um Variablen in einer Eingabezeichenfolge mit Werten zu ersetzen, die Sie angeben. Weitere Informationen finden Sie unter Fn::Sub.	15.05.2010
Neue Ressourcen	19. September 2016	AWS::KMS::Alias Verwenden Sie die Ressource `AWS::KMS::Alias` , um für den AWS KMS key einen Alias zu erstellen.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	19. September 2016	AWS::EC2::SpotFleet Verwenden Sie für die `LaunchSpecifications` -Eigenschaft die `SpotPrice` -Eigenschaft, um einen Angebotspreis für einen bestimmten Instance-Typ anzugeben. AWS::ECS::Cluster Verwenden Sie die `ClusterName` -Eigenschaft, um einen Namen für einen Amazon Elastic Container Service-Cluster anzugeben. AWS::ECS::TaskDefinition Verwenden Sie die `TaskRoleArn` Eigenschaft, um eine AWS Identity and Access Management Rolle anzugeben, die Amazon Elastic Container Service-Container verwenden, um in Ihrem Namen AWS Aufrufe zu tätigen. Verwenden Sie die `Family`-Eigenschaft für die Registrierung einer Aufgabendefinition an eine bestimmte Familie. AWS::Elasticsearch::Domain Verwenden Sie die `ElasticsearchVersion` -Eigenschaft, um anzugeben, welche Version von Elasticsearch verwendet werden soll.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	11. August 2016	Verwenden Sie die folgenden Elastic Load Balancing Balancer-Ressourcen für den Application Load Balancer, um den eingehenden Anwendungsdatenverkehr auf mehrere Ziele, z. B. EC2 Instances, in mehreren Availability Zones zu verteilen: • AWS::ElasticLoadBalancingV2::Listener • AWS::ElasticLoadBalancingV2::ListenerRule • AWS::ElasticLoadBalancingV2::LoadBalancer • AWS::ElasticLoadBalancingV2::TargetGroup	15.05.2010
Aktualisierte Ressource	11. August 2016	AWS::AutoScaling::AutoScalingGroup Verwenden Sie die `TargetGroupARNs` -Eigenschaft, um die Auto-Scaling-Gruppe mit einer oder mehreren Application Load Balancer-Zielgruppen zu verknüpfen. AWS::ECS::Service Verwenden Sie für die `LoadBalancers` -Lasteigenschaft die `TargetGroupArn` -Eigenschaft, um einen Amazon Elastic Container Service-Service mit einer Application Load Balancer-Zielgruppe zu verknüpfen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	09. August 2016	CloudFormation hat die folgenden Ressourcen hinzugefügt: AWS::ApplicationAutoScaling::ScalableTarget und AWS::ApplicationAutoScaling::ScalingPolicy Verwenden Sie eine Auto Scaling von Anwendungen-Skalierungsrichtlinie, um zu definieren, wann und wie eine Zielressource skaliert wird. AWS::CertificateManager::Certificate Stellen Sie ein AWS Certificate Manager Zertifikat bereit, das Sie mit anderen AWS Diensten verwenden können, um sichere Verbindungen zu ermöglichen.	15.05.2010
Aktualisierte Ressourcen	09. August 2016	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::CloudFront::Distribution Für die `ViewerCertificate` -Eigenschaft der Verteilungskonfiguration können Sie ein AWS Certificate Manager -Zertifikat angeben. Für die `Origin`-Eigenschaft der Verteilungskonfiguration können Sie benutzerdefinierte Header und die SSL-Protokolle für benutzerdefinierte Ursprünge angeben. AWS::EFS::FileSystem Sie können den Leistungsmodus für ein Amazon Elastic File System-Dateisystem angeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	20. Juli 2016	AWS IoT Wird verwendet, AWS IoT um eine AWS IoT Richtlinie, ein X.509-Zertifikat, eine Zuordnung zwischen einer Richtlinie und einem Prinzipal (ein X.509-Zertifikat oder andere Anmeldeinformationen), eine AWS IoT Sache, eine Zuordnung zwischen einem Prinzipal und einem Ding oder eine Regel zu deklarieren. AWS IoT • AWS::IoT::Certificate • AWS::IoT::Policy • AWS::IoT::PolicyPrincipalAttachment • AWS::IoT::Thing • AWS::IoT::ThingPrincipalAttachment • AWS::IoT::TopicRule	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	20. Juli 2016	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::IAM::Group, AWS::IAM::Role, AWS::IAM::User Verwenden Sie die Namens-Eigenschaften, um einen benutzerdefinierten Namen für AWS Identity and Access Management (IAM)-Ressourcen anzugeben. AWS::ApiGateway::Method Für die `Integration` -Eigenschaft können Sie die `PassthroughBehavior` -Eigenschaft verwenden, um anzugeben, wann Amazon API Gateway Anforderungen an das spezifische Back-End übergibt. AWS::ApiGateway::Model und AWS::ApiGateway::RestApi Sie können JSON-Objekte für die Eigenschaften `Schema` und `Body` angeben.	15.05.2010
Bereich Auto Scaling UpdatePolicy	9. Juni 2016	Verwenden Sie für das `UpdatePolicy` -Attribut die `AutoScalingReplacingUpdate` -Eigenschaft, um anzugeben, ob eine Auto-Scaling-Gruppe und die darin enthaltenen Instances ersetzt werden, wenn Sie die Auto-Scaling-Gruppe aktualisieren. CloudFormation Behält während eines Austauschs die alte Auto Scaling Scaling-Gruppe bei, bis die neue erfolgreich erstellt wurde, sodass ein Rollback zur alten Auto Scaling Scaling-Gruppe durchgeführt werden CloudFormation kann, falls das Update fehlschlägt. Weitere Informationen finden Sie unter UpdatePolicy Attribute.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressource	9. Juni 2016	CloudFormation hat die folgenden Ressourcen hinzugefügt: AWS::EC2::FlowLog Erstellt ein Amazon Elastic Compute Cloud-Flow-Protokoll, mit dem IP-Datenverkehr für eine bestimmte Netzwerkschnittstelle, ein Subnetz oder eine VPC erfasst wird. AWS::KinesisFirehose::DeliveryStream Erstellt einen Lieferstream, der Streaming-Daten in Echtzeit an ein Ziel wie Amazon Simple Storage Service, Amazon Redshift oder Amazon OpenSearch Service liefert.	15.05.2010
Aktualisierte Ressourcen	9. Juni 2016	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::Kinesis::Stream Verwenden Sie die `Name`-Eigenschaft, um einen Namen für einen Amazon Kinesis-Stream anzugeben. AWS::Lambda::Function Für die `Code`-Eigenschaft können Sie die `ZipFile`-Eigenschaft und das cfn-Antwortmodul für `nodejs4.3` -Laufzeitumgebungen verwenden. AWS::SNS::Topic CloudFormation Updates für die Themenressource Amazon Simple Notification Service aktiviert.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressource	25. April 2016	Verwenden Sie die AWS::EC2::HostRessource, um einen vollständig dedizierten physischen Server für das Starten von EC2 Instances zuzuweisen.	15.05.2010
Aktualisierte Ressourcen	25. April 2016	AWS::EC2::Instance Verwenden Sie die Eigenschaften `Affinity` und `HostId`, um Instances auf einem Dedicated Host von Amazon Elastic Compute Cloud zu starten. AWS::ECS::Service Verwenden Sie die `DeploymentConfiguration` -Eigenschaft, um zu konfigurieren, wie viele Aufgaben während einer Bereitstellung ausgeführt werden können. AWS::ECS::TaskDefinition CloudFormation Unterstützung für zusätzliche Eigenschaften der Amazon Elastic Container Service-Container-Definition hinzugefügt. AWS::GameLift::Fleet Verwenden Sie die `MinSize` Eigenschaften `MaxSize` und, um die maximale und minimale Anzahl von EC2 Instances anzugeben, die in Ihrer Amazon GameLift Server-Flotte zulässig sind. AWS::Lambda::Function Verwenden Sie die `FunctionName` Eigenschaft, um einen Namen für Ihre AWS Lambda Funktion anzugeben. Sie können auch Python 2.7 verwenden, um eine Inlinefunktion anzugeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	18. April 2016	Amazon API Gateway Verwenden Sie die Amazon API Gateway Gateway-Ressourcen, um Inhalte in beliebiger Größenordnung zu veröffentlichen, APIs zu verwalten und zu überwachen. Sie können festlegen APIs , dass Kunden per Anruf auf Ihre Back-End-Services zugreifen können, z. B. auf Anwendungen, auf EC2 denen Instances ausgeführt werden, oder Code, auf dem AWS Lambda ausgeführt wird. • AWS::ApiGateway::Account • AWS::ApiGateway::ApiKey • AWS::ApiGateway::Authorizer • AWS::ApiGateway::BasePathMapping • AWS::ApiGateway::ClientCertificate • AWS::ApiGateway::Deployment • AWS::ApiGateway::Method • AWS::ApiGateway::Model • AWS::ApiGateway::Resource • AWS::ApiGateway::RestApi • AWS::ApiGateway::Stage AWS::Events::Rule Erstellen Sie eine Amazon CloudWatch Events-Regel, die Änderungen an AWS Ressourcen in Ihrem Konto überwacht (Ereignisse). Wenn ein eingehendes Ereignis den Bedingungen entspricht, die Sie in der Regel beschrieben haben, sendet Amazon CloudWatch Events Nachrichten an Ihre angegebenen Ziele, wie AWS Lambda Funktionen oder Amazon	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		Simple Notification Service-Themen, und aktiviert diese. AWS::WAF::SizeConstraintSet und AWS::WAF::XssMatchSet Verwenden Sie die beiden AWS WAF Regeln, um die Größe einer Webanfrage zu überprüfen oder um Cross-Site-Scripting-Angriffe zu verhindern.	
Neue Ressourcen	31. März 2016	Verwenden Sie die AWS::Lambda::AliasRessource , um Aliase für Ihre AWS Lambda Funktionen zu erstellen, und die AWS::Lambda::VersionRessource, um Versionen Ihrer Funktionen zu erstellen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	31. März 2016	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::EMR::Cluster und AWS::EMR::InstanceGroupConfig Verwenden Sie die `EbsConfiguration` -Eigenschaft zum Konfigurieren von Amazon Elastic Block Store-Speicher-Volumes für Ihre Amazon EMR-Cluster oder Instance-Gruppen. AWS::Lambda::Function Verwenden Sie die `VpcConfig` -Eigenschaft, um AWS Lambda -Funktionen den Zugriff auf Ressourcen in einer VPC zu ermöglichen. AWS::S3::Bucket Für die Amazon Simple Storage Service-Lebenszyklusregeln können Sie mehrere Übergangsregeln festlegen, die angeben, wann Objekte in eine bestimmte Speicherklasse übergehen.	15.05.2010
Änderungssätze	29. März 2016	Verwenden Sie vor dem Aktualisieren von Stacks Änderungssätze, um zu sehen, wie sich Ihre Änderungen auf laufende Ressourcen auswirken könnten. Weitere Informationen finden Sie unter Aktualisieren von Stacks mithilfe von Änderungssätzen.	15.05.2010
Neue Ressourcen	15. März 2016	Verwenden Sie die AWS::GameLift::FleetRessourcen AWS::GameLift::AliasAWS::GameLift::Build, und, um Multiplayer-Spieleserver in bereitzustellen AWS.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	26. Februar 2016	CloudFormation hat die folgenden Ressourcen hinzugefügt: AWS::ECR::Repository Erstellen Sie Amazon Elastic Container Registry-Repositorys, in denen Benutzer Docker-Images per Push und Pull übertragen können. AWS::EC2::NatGateway Verwenden Sie das NAT-Gateway (Network Address Translator), um EC2 Instances in einem privaten Subnetz die Verbindung zum Internet zu ermöglichen. AWS::Elasticsearch::Domain Erstellen Sie Amazon OpenSearch Service-Domains, auf denen ältere Elasticsearch OSS-Cluster ausgeführt werden. AWS::EMR::Cluster, AWS::EMR::InstanceGroupConfig, AWS::EMR::Step Verwenden Sie die Amazon-EMR-Ressourcen als Unterstützung beim Analysieren und Verarbeiten riesiger Datenmengen. Sie können Cluster erstellen und dann darin Aufträge ausführen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierte Ressourcen	26. Februar 2016	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::CloudTrail::Trail Verwenden Sie die `IsMultiRegionTrail` Eigenschaft, um anzugeben, ob in der Region, in der Sie einen Stapel erstellen, oder in allen Regionen ein AWS CloudTrail Trail erstellt werden soll. AWS::Config::ConfigurationRecorder Verwenden Sie für die Aufzeichnungsgruppe die `IncludeGlobalResourceTypes` -Eigenschaft, um alle globalen Ressourcentypen aufzuzeichnen. AWS::RDS::DBCluster Verwenden Sie die Eigenschaften `KmsKeyId` und `StorageEncrypted` zum Verschlüsseln von Datenbank-Instances im Cluster.	15.05.2010
Beibehalten von Ressourcen	26. Februar 2016	Verwenden Sie für Stacks im `DELETE_FAILED` -Status den `RetainResources` -Parameter, um Ressourcen beizubehalten, die von CloudFormation nicht gelöscht werden können. Weitere Informationen finden Sie unter Löschen Sie die Stack-Fehler.	15.05.2010
Aktualisieren von Stack-Tags	26. Februar 2016	Sie können Stack-Tags hinzufügen, ändern oder entfernen, wenn Sie einen Stack aktualisieren. Weitere Informationen finden Sie unter CloudFormation Stacks-Updates.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Fortgesetztes Rollback von fehlgeschlagenen Aktualisierungs-Rollbacks	25. Januar 2016	Für einen Stack, der sich im `UPDATE_ROLLBACK_FAILED` -Status befindet, können Sie das Rollback der Aktualisierung fortsetzen, um den Stack in einen funktionierenden Zustand zu versetzen. Auf diese Weise können Sie den Stack auf seine ursprünglichen Einstellungen zurücksetzen und dann erneut versuchen, ihn zu aktualisieren. Weitere Informationen finden Sie unter Fortsetzen des Rollback einer Aktualisierung.	15.05.2010
Es sind neue Beispielvorlagen für die Region Asien-Pazifik (Seoul) verfügbar.	7. Januar 2016	Die folgende Sammlung von CloudFormation Beispielvorlagen bezieht sich auf die Region ap-northeast-2: • Lösungsbeispiele • Anwendungs-Frameworks • Services Weitere Informationen finden Sie unter Arbeiten mit Vorlagen. CloudFormation	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	28. Dezember 2015	CloudFormation hat die folgenden Ressourcen hinzugefügt: AWS::DirectoryService::MicrosoftAD Verwenden Sie die Microsoft Active Directory-Ressource, um ein Microsoft Active Directory-Verzeichnis in zu erstellen AWS. AWS::Logs::Destination und AWS::Logs::LogStream Verwenden Sie die Amazon CloudWatch Logs-Ressourcen, um ein Ziel für die Echtzeitverarbeitung von Protokolldaten bzw. Protokollstreams zu erstellen. AWS::WAF::ByteMatchSet, AWS::WAF::IPSet, AWS::WAF::Rule, AWS::WAF::SqlInjectionMatchSet, und AWS::WAF::WebACL Verwenden Sie die AWS WAF Ressourcen, um Webanfragen zu Ihren Inhalten zu kontrollieren und zu überwachen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Ressourcenaktualisierungen	28. Dezember 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::CloudFront::Distribution Verwenden Sie für die Distributionskonfiguration die `WebACLId` Eigenschaft, um einer CloudFront Amazon-Distribution eine AWS WAF Web Access Control List (ACL) zuzuordnen. Für das Cache-Verhalten und Standard-Cache-Verhalten können Sie einen Standardwert und einen Höchstwert für Time-to-Live (TTL) angeben. AWS::DynamoDB::Table Sie können einen globalen sekundären Index erstellen, aktualisieren oder löschen, ohne die Amazon DynamoDB-Tabelle zu ersetzen. AWS::S3::Bucket Verwenden Sie die `ReplicationConfiguration` -Eigenschaft, um anzugeben, welche Objekte repliziert werden und wo sie gespeichert werden. Verwenden Sie die Eigenschaften in der `NotificationConfiguration` -Eigenschaft, um Filter anzugeben, damit Amazon Simple Storage Service Benachrichtigungen für von Ihnen angegebene Objekte sendet.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Gruppieren und Sortieren von Parametern	3. Dezember 2015	Verwenden Sie den AWS::CloudFormation::InterfaceMetadatenschlüssel, um Parameter in der CloudFormation Konsole zu gruppieren und zu sortieren, wenn Benutzer einen Stack mit Ihrer Vorlage erstellen oder aktualisieren.	15.05.2010
UpdatePolicy-Attribut	3. Dezember 2015	Für ein Auto Scaling -Aktualisierungsrichtlinien-Attribut verwenden Sie die `MinSuccessfulInstancesPercent` -Eigenschaft, um den Prozentsatz der Instances anzugeben, die Erfolg signalisieren müssen, damit die Aktualisierung erfolgreich verläuft.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	3. Dezember 2015	CloudFormation hat die folgenden Ressourcen hinzugefügt: AWS::CodePipeline::Pipeline und AWS::CodePipeline::CustomActionType Verwenden Sie die CodePipeline Ressourcen, um eine Pipeline zu erstellen, die beschreibt, wie Softwareänderungen einen Release-Prozess durchlaufen. AWS::Config::ConfigurationRecorder, AWS::Config::DeliveryChannel und AWS::Config::ConfigRule Verwenden Sie die AWS Config Ressourcen, um Konfigurationsänderungen an bestimmten AWS Ressourcen zu überwachen. AWS::KMS::Key Verwenden Sie die Ressource AWS Key Management Service (AWS KMS), um vom Kunden verwaltete Schlüssel zu erstellen AWS KMS , mit denen Benutzer kleine Datenmengen verschlüsseln können. AWS::SSM::Document Verwenden Sie den Amazon EC2 Systems Manager, um ein Dokument zu erstellen, das On-Instance-Konfigurationen spezifiziert.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Ressourcenaktualisierung	3. Dezember 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::AutoScaling::LaunchConfiguration Geben Sie an, ob EBS-Volumes verschlüsselt werden. AWS::AutoScaling::ScalingPolicy Sie können zwei verschiedene Richtlinientypen (einfache und schrittweise Skalierung) verwenden , um festzulegen, wie eine Auto Scaling-Gruppe skaliert, wenn ein Amazon CloudWatch (CloudWatch) -Alarm verletzt wird. AWS::CloudTrail::Trail Verwenden Sie die CloudWatch Eigenschaften, um Protokolle an eine CloudWatch Protokollgruppe zu senden. Sie können einem Trail Tags hinzufügen und eines angeben AWS KMS key , das Sie zum Verschlüsseln von Protokollen verwenden möchten. AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentConfig und AWS::CodeDeploy::DeploymentGroup Verwenden Sie die `DeploymentGroupName` Eigenschaften `ApplicationName` `DeploymentConfigName` , und, um benutzerdefinierte Namen für CodeDeploy Ressourcen anzugeben. AWS::DynamoDB::Table Verwenden Sie die `StreamSpecification` -Eigenschaft, um Einstellungen zum Erfassen von Änderungen an Elementen anzugeben, die in einer	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		Amazon DynamoDB (DynamoDB)-Tabelle gespeichert sind. AWS::EC2::Instance Verwenden Sie die `SsmAssociations` Eigenschaft, um ein Amazon EC2 Systems Manager Manager-Dokument mit einer Instance zu verknüpfen. AWS::EC2::SpotFleet Verwenden Sie die `AllocationStrategy` -Eigenschaft, um festzulegen, wie Zielkapazität über Spot-Pools hinweg zugeordnet werden soll. Verwenden Sie die `ExcessCapacityTerminationPolicy` -Eigenschaft, um anzugeben, wie Instances beendet werden, wenn die Zielkapazität unter der Größe der Spot-Flotte liegt. AWS::Redshift::Cluster Verwenden Sie die `KmsKeyId`-Eigenschaft zum Verschlüsseln von Daten um AWS KMS key in einem Amazon-Redshift-Cluster anzugeben. AWS::WorkSpaces::Workspace Verwenden Sie die Verschlüsselungseigenschaften zum Verschlüsseln von auf Volumes gespeicherten Daten.	
Ressourcenaktualisierung	4. November 2015	Verwenden Sie für die AWS::EC2::VolumeRessource die `AutoEnableIO` Eigenschaft, um den I/O Betrieb automatisch wieder aufzunehmen, wenn die Daten eines Volumes inkonsistent werden.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	1. Oktober 2015	CloudFormation hat die folgenden Ressourcen hinzugefügt: AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentGroup und AWS::CodeDeploy::DeploymentConfig Verwenden Sie die CodeDeploy Ressourcen, um Bereitstellungen für EC2 oder lokale Instanzen zu erstellen und anzuwenden. AWS::DirectoryService::SimpleAD Verwenden Sie die Simple Active Directory-Ressource, um ein AWS Directory Service Simple AD zu erstellen. Dabei handelt es sich um ein mit Microsoft Active Directory kompatibles Verzeichnis. AWS::EC2::PlacementGroup Verwenden Sie eine Placement-Gruppe, um einen Cluster von Instances in einem Netzwerk mit geringer Latenz zu erstellen. AWS::EC2::SpotFleet Verwenden Sie eine Spot-Flotte, um eine Sammlung von Spot Instances zu starten, die unterbrechbare Aufgaben ausführen. AWS::Lambda::EventSourceMapping Verwenden Sie die Ressource zur Zuordnung von Ereignisquellen, um einen Stream als Ereignisquelle für eine AWS Lambda (Lambda-) Funktion anzugeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		AWS::Lambda::Permission Verwenden Sie eine Lambda-Berechtigung zum Hinzufügen einer Anweisung zur Richtlinie einer Lambda-Funktion. AWS::Logs::SubscriptionFilter Verwenden Sie den Abonnementfilter, um zu definieren, welche Protokollereignisse an Ihren Kinesis-Stream zugestellt werden. AWS::RDS::DBCluster und AWS::RDS::DBClusterParameterGroup Verwenden Sie die Cluster- und Cluster-Parametergruppen-Ressourcen, um einen Amazon Aurora-DB-Cluster zu erstellen. AWS::WorkSpaces::Workspace Wird verwendet WorkSpaces , um cloudbasierte Desktop-Erlebnisse zu erstellen.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Ressourcenaktualisierungen	1. Oktober 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::ElastiCache::ReplicationGroup Verwenden Sie die intrinsische `Fn::GetAtt` -Funktion, um eine Liste der schreibgeschützten Replikatadressen und -Ports abzurufen. AWS::OpsWorks::Stack Verwenden Sie die `AgentVersion` Eigenschaft, um einen bestimmten OpsWorks Agenten anzugeben. AWS::OpsWorks::App Verwenden Sie die `Environment` Eigenschaft, um Umgebungsvariablen für eine OpsWorks App anzugeben. AWS::S3::Bucket Für die NotificationConfigurationEigenschaft können Sie Benachrichtigungseinstellungen für Lambda-Funktionen und Amazon Simple Queue Service (Amazon SQS) -Warteschlangen konfigurieren.	15.05.2010
IAM-Bedingungsschlüssel	1. Oktober 2015	Verwenden Sie für AWS Identity and Access Management (IAM-) Richtlinien CloudFormation spezifische Bedingungsschlüssel, um anzugeben, wann eine IAM-Richtlinie wirksam wird. Weitere Informationen finden Sie unter Zugriffskontrolle mit AWS Identity and Access Management.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
AWS CloudFormation Designer	1. Oktober 2015	Verwenden Sie AWS CloudFormation Designer, um Vorlagen über eine drag-and-drop Benutzeroberfläche zu erstellen und zu ändern.	15.05.2010
Neue Ressource	24. August 2015	Verwenden Sie die AWS::EC2::VPCEndpointRessource, um eine private Verbindung zwischen Ihrer VPC und einem anderen AWS Service herzustellen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Ressourcenaktualisierungen	24. August 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::ElasticBeanstalk::Environment Verwenden Sie die `Tags` Eigenschaft, um Tags (Schlüssel-Wert-Paare) für eine AWS Elastic Beanstalk (Elastic Beanstalk-) Umgebung anzugeben . AWS::Lambda::Function Verwenden Sie für die Code-Eigenschaft die `ZipFile`-Eigenschaft, um den Quellcode Ihrer Lambda-Funktion direkt in eine Vorlage zu schreiben . Derzeit können Sie die `ZipFile`-Eigenschaft nur für `nodejs`-Laufzeitumgebungen verwenden. Sie können weiterhin auf eine Datei in einem S3-Bucket für alle Laufzeitumgebungen zeigen, z. B. `java8` und `nodejs`. AWS::OpsWorks::Instance Verwenden Sie die `EbsOptimized` -Eigenschaft, um anzugeben, ob eine Instance für Amazon Elastic Block Store (Amazon EBS)-E/A optimiert ist. AWS::RDS::DBInstance Für die `SourceDBInstanceIdentifier` -Eigenschaft können Sie eine Datenbank-Instance in einer anderen Region angeben, um eine regionsübergreifende Read Replica zu erstellen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Amazon S3-Vorlagen-URL	24. August 2015	Für Versioning-fähige Buckets können Sie eine Versions-ID in einer Amazon S3-Vorlagen-URL angeben, wenn Sie einen Stack wie `https://s3.amazonaws.com/templates/myTemplate.template?versionId=123ab1cdeKdOW5IH4GAcYbEngcpTJTDW` erstellen oder aktualisieren.	15.05.2010
Neue Ressource	3. August 2015	Verwenden Sie die Ressource AWS::EFS::FileSystem, um ein Amazon Elastic File System (Amazon EFS)-Dateisystem zu erstellen, und die Ressource AWS::EFS::MountTarget, um einen Mountpoint für ein Dateisystem zu erstellen.	15.05.2010
Änderung von Berechtigungsanforderungen	11. Juni 2015	Wenn Sie eine AWS::RDS::DBInstance-Ressource erstellen oder aktualisieren, benötigen Sie auch die Berechtigung zum Aufrufen der `ec2:DescribeAccountAttributes` -Aktion.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	11. Juni 2015	CloudFormation hat die folgenden Ressourcen hinzugefügt: AWS::DataPipeline::Pipeline Verwenden Sie Daten-Pipelines, um das Verschieben und Transformieren von Daten zu automatisieren. Amazon Elastic Container Service-Ressourcen Verwenden Sie die AWS::ECS::TaskDefinitionRessourcen AWS::ECS::ServiceAWS::ECS::Cluster, und, um Docker-Container auf einem Cluster von EC2 Instanzen zu erstellen. AWS::ElastiCache::ReplicationGroup Verwenden Sie Replikationsgruppen, um eine Sammlung von Knoten mit einem primären Lese-Schreib-Cluster und maximal fünf sekundären schreibgeschützten Clustern zu erstellen. AWS::IAM::ManagedPolicy Verwenden Sie verwaltete Richtlinien, um Richtlinien in Ihrem AWS Konto zu erstellen, mit denen Sie IAM-Benutzern, -Gruppen und -Rollen Berechtigungen zuweisen können. AWS::Lambda::Function Verwenden Sie Lambda Funktionen für die Ausführung von Code als Reaktion auf Ereignisse. AWS::RDS::OptionGroup Verwenden Sie Optionsgruppen, um Sie beim Erstellen und Verwalten von Amazon Relational	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		Database Service (Amazon RDS)-Datenbanken zu unterstützen.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Ressourcenaktualisierungen	11. Juni 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::EC2::Subnet Verwenden Sie die `MapPublicIpOnLaunch` -Eigenschaft, um automatisch öffentliche IP-Adressen zu Instances in einem Subnetz zuzuweisen. AWS::ElastiCache::CacheCluster Verwenden Sie die `SnapshotName` -Eigenschaft für die Wiederherstellung von Snapshot-Daten in einem neuen Redis-Cache-Cluster. AWS::IAM::User Verwenden Sie für die `LoginProfile` -Eigenschaft die `PasswordResetRequired` -Eigenschaft, damit Benutzer ein neues Passwort festlegen müssen, wenn sie sich bei der AWS Management Console anmelden. AWS::OpsWorks::Layer Verwenden Sie die `LifecycleEventConfiguration` Eigenschaft, um Lebenszyklusereignisse für eine OpsWorks Ebene zu konfigurieren. AWS::S3::Bucket Verwenden Sie für die `LifecycleConfiguration` -Eigenschaft die Eigenschaften `NoncurrentVersionExpirationInDays` und `NoncurrentVersionTransition` , um Lebenszyklusregeln für nicht aktuelle Objektversionen anzugeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Parametertypen	19. Mai 2015	Wann immer Sie die CloudFormation Konsole verwenden, um einen Stack zu erstellen oder zu aktualisieren, können Sie anhand des Kennworts ID, Name oder Name nach Werten für AWS spezifische Parametertypen suchen. CloudFormation Außerdem wurde Unterstützung für die folgenden AWS-spezifischen Parametertypen hinzugefügt. Weitere Informationen finden Sie unter Parameter. • `AWS::EC2::AvailabilityZone::Name` • `List<AWS::EC2::AvailabilityZone::Name>` • `AWS::EC2::Instance::Id` • `List<AWS::EC2::Instance::Id>` • `AWS::EC2::Image::Id` • `List<AWS::EC2::Image::Id>` • `AWS::EC2::SecurityGroup::GroupName` • `List<AWS::EC2::SecurityGroup::GroupName>` • `AWS::EC2::Volume::Id` • `List<AWS::EC2::Volume::Id>` • `AWS::Route53::HostedZone::Id` • `List<AWS::Route53::HostedZone::Id>`	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Ressourcen	16. April 2015	CloudFormation die folgenden Ressourcen wurden hinzugefügt: AWS::AutoScaling::LifecycleHook Verwenden Sie Auto Scaling-Lebenszyklus-Hooks zum Steuern des Status einer Instance, nachdem sie gestartet oder beendet wurde. AWS::RDS::EventSubscription Verwenden Sie Ereignisabonnements, um Benachrichtigungen zu Amazon RDS-Ereignissen zu erhalten.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Ressourcenaktualisierungen	16. April 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::AutoScaling::AutoScalingGroup Verwenden Sie die `NotificationConfigurations` -Eigenschaft, um mehrere Benachrichtigungen anzugeben. AWS::AutoScaling::LaunchConfiguration Verwenden Sie die `PlacementTenancy` -Eigenschaft zum Angeben der Tenancy von Instances. Verwenden Sie die `ClassicLinkVPCSecurityGroups` Eigenschaften `ClassicLinkVPCId` und, um EC2 -Classic-Instances mit einer ClassicLink -fähigen VPC zu verknüpfen. AWS::AutoScaling::ScalingPolicy Verwenden Sie die `MinAdjustmentStep` -Eigenschaft, um eine Mindestanzahl von Instances festzulegen, die während eines Skalierungsereignisses hinzugefügt oder entfernt werden. AWS::CloudFront::Distribution Für Betrachterzertifikate verwenden Sie die `MinimumProtocolVersion` -Eigenschaft, um eine Mindestprotokollversion anzugeben. Verwenden Sie für Cache-Verhalten die `CachedMethods` Eigenschaft, um anzugeben, für welche Methoden Amazon CloudFront (CloudFront) Antworten zwischenspeichert. Verwenden Sie für Ursprünge den, `OriginPath` um einen Pfad anzugeben, der	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		zum Anfordern von Inhalten CloudFront verwendet wird. AWS::ElastiCache::CacheCluster Verwenden Sie für Memcached-Cache-Cluster die `PreferredAvailabilityZones` Eigenschaften `AZMode` und, um Knoten in mehreren Availability Zones () AZs anzugeben. AWS::EC2::Volume Verwenden Sie die `KmsKeyId`-Eigenschaft, um einen vom Kunden verwalteten Schlüssel für verschlüsselte Volumes anzugeben. AWS::OpsWorks::Instance Verwenden Sie die `TimeBasedAutoScaling` -Eigenschaft für die automatische Skalierung der Instances basierend auf einem von Ihnen festgelegten Zeitplan. AWS::OpsWorks::Layer Verwenden Sie die `LoadBasedAutoScaling` -Eigenschaft, um lastbasierte Skalierungsrichtlinien anzugeben. Verwenden Sie für Volume-Konfigurationen die `Iops` Eigenschaften `VolumeType` und, um jeweils einen Volume-Typ und die Anzahl der I/O Operationen pro Sekunde anzugeben. AWS::RDS::DBInstance Verwenden Sie die `CharacterSetName` -Eigenschaft, um einen Zeichensatz für unterstützte Datenbank-Engines anzugeben.	

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		Verwenden Sie die `StorageEncrypted` -Eigenschaft, um anzugeben, ob Datenbank-Instances verschlüsselt werden, und `KmsKeyId`, um einen vom Kunden verwalteten Schlüssel für verschlüsselte Datenbank-Instances anzugeben. AWS::Route53::HealthCheck Verwenden Sie die `HealthCheckTags` -Eigenschaft, um Tags mit Zustandsprüfungen zu verknüpfen. AWS::Route53::HostedZone Verwenden Sie die `VPCs`-Eigenschaft zum Erstellen von privaten gehosteten Zonen. Verwenden Sie die `HostedZoneTags` -Eigenschaft, um Tags mit gehosteten Zonen zu verknüpfen.	
Neuer Vorlagenabschnitt	16. April 2015	Fügen Sie den Metadaten-Abschnitt zu Ihren Vorlagen hinzu, um zufällige JSON-Objekte einzuschließen, die Ihre Vorlagen beschreiben, wie z. B. die Design- oder Implementierungsdetails.	15.05.2010
Ressourcenaktualisierung	8. April 2015	Für die AWS::CloudFormation::Custom ResourceRessource können Sie die Lambda-Funktion Amazon Resource Names (ARNs) in der `ServiceToken` Eigenschaft angeben.	15.05.2010
Amazon RDS update	24. Dezember 2014	CloudFormation hat zwei neue Eigenschaften für RDS-DB-Instances hinzugefügt. Sie können eine Optionsgruppe mit einer DB-Instance verknüpfen und den DB Instance-Speichertyp angeben. Weitere Informationen finden Sie unter AWS::RDS::DBInstance.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Elastic Load Balancing update	24. Dezember 2014	Sie können die `ConnectionSettings` -Eigenschaft verwenden, um anzugeben, wie lange Verbindungen inaktiv bleiben können. Weitere Informationen finden Sie unter AWS::ElasticLoadBalancing::LoadBalancer.	15.05.2010
Route 53 aktualisieren	6. November 2014	Sie können jetzt für Route 53 gehostete Zonen, Zustandsprüfungen, Failover-Datensätze und Geolocation-Datensätze bereitstellen und verwalten.	15.05.2010
Verbesserung der fortlaufenden Auto Scaling-Aktualisierung	6. November 2014	Während eines Updates können Sie das `WaitOnResourceSignals` Flag verwenden, um anzuweisen, darauf CloudFormation zu warten, dass Instances erfolgreich signalisieren. Auf diese Weise aktualisiert CloudFormation die nächsten Instances erst dann, nachdem der aktuelle Stapel bereit ist. Weitere Informationen finden Sie unter UpdatePolicyAttribut.	15.05.2010
Neue VPC Fn: Attribute GetAtt	6. November 2014	Bei gegebener VPC-ID können Sie die Standardsicherheitsgruppe und Netzwerk-ACL für diese VPC abrufen. Weitere Informationen finden Sie unter Fn::GetAtt.	15.05.2010
Neue AWS-spezifische Parametertypen	6. November 2014	Sie können AWS-spezifische Parametertypen in Ihren CloudFormation Vorlagen angeben. In der CloudFormation Konsole stellen diese Parametertypen eine Dropdownliste mit gültigen Werten bereit. Mit der API oder AWS CLI CloudFormation können Werte für diese Parametertypen schnell validiert werden, bevor ein Stack erstellt oder aktualisiert wird. Weitere Informationen finden Sie unter Parameter.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
CreationPolicy Attribut	6. November 2014	Mit dem CreationPolicy Attribut können Sie anweisen, CloudFormation zu warten, bis Anwendungen auf EC2 Instanzen bereit sind, bevor Sie mit der Stack-Erstellung fortfahren. Sie können eine Erstellungsrichtlinie anstelle einer Wartebedingung und eines Wartebedingungs-Handles verwenden. Weitere Informationen finden Sie unter CreationPolicyAttribut.	15.05.2010
Amazon hat Werte CloudFront weitergeleitet	29. Septembe 2014	Für Cache-Verhaltensweisen können Sie Forward-Header an den Ursprung weiterleiten. Siehe Forwarded Values.	15.05.2010
AWS OpsWorks update	29. Septembe 2014	Für Chef 11.10 können Sie die `ChefConfiguration` -Eigenschaft verwenden, um Berkshelf zu aktivieren. Sie können auch die AWS OpsWorks integrierten Sicherheitsgruppen mit Ihren AWS OpsWorks Stacks verwenden. Weitere Informationen finden Sie unter AWS::OpsWorks::Stack.	15.05.2010
Elastic Load Balancing -Markierungsunterstützung	29. Septembe 2014	AWS CloudFormation kennzeichnet Elastic Load Balancing Load Balancer mit Tags auf Stackebene. Sie können auch Ihre eigenen Tags zu einem Load Balancer hinzufügen. Siehe AWS::ElasticLoadBalancing::LoadBalancer.	15.05.2010
Aktualisierung der Amazon Simple Notification Service-Themenrichtlinie	29. Septembe 2014	Sie können jetzt Amazon SNS-Themenrichtlinien aktualisieren. Weitere Informationen finden Sie unter AWS::SNS::TopicPolicy.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierung der RDS DB-Instance	5. September 2014	Sie können festlegen, ob eine DB-Instance mit dem Internet verbunden ist, indem Sie die `PubliclyAccessible` -Eigenschaft in der AWS::RDS::DBInstance-Ressource verwenden.	15.05.2010
UpdatePolicy Aktualisierung der Attribute	05. September 2014	Sie können eine Aktualisierungsrichtlinie für eine Auto-Scaling-Gruppe mit einer verknüpften geplanten Aktion angeben. Weitere Informationen finden Sie unter UpdatePolicyAttribut.	15.05.2010
CloudWatch Amazon-Unterstützung	10. Juli 2014	Sie können CloudFormation es verwenden, um Amazon CloudWatch Logs (CloudWatch Logs) -Protokollgruppen und Metrikfilter bereitzustellen und zu verwalten. Weitere Informationen finden Sie unter AWS::Logs::LogGroup oder AWS::Logs::MetricFilter.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Aktualisierung der CloudFront Amazon-Vertriebskonfiguration	17. Juni 2014	Sie können zusätzliche Eigenschaften für die CloudFront Vertriebskonfiguration angeben: • Benutzerdefinierte Fehlerantworten definieren benutzerdefinierte Fehlermeldungen für 4xx- und 5xx HTTP-Statuscodes. • Die Preisklasse definiert den Höchstpreis, den Sie für den CloudFront -Service zahlen möchten. • Einschränkungen definieren, wer Ihre Inhalte anzeigen kann. • Betrachterzertifikate geben das zu verwendende Zertifikat an, wenn Betrachter HTTPS verwenden. • Für Cache-Verhaltensweisen können Sie zulässige HTTP-Methoden angeben und festlegen, ob Cookies weitergeleitet werden. Weitere Informationen finden Sie unter AWS::CloudFront::Distribution.	15.05.2010
EC2 Instanz-Update	17. Juni 2014	Sie können angeben, ob eine Instance angehalten oder beendet wird, wenn Sie den Ausschaltbefehl des Betriebssystems der Instance aufrufen. Weitere Informationen finden Sie unter AWS::EC2::Instance.	15.05.2010
EBS-Volume-Aktualisierung	17. Juni 2014	Sie können verschlüsselte EBS-Volumes mit unterstützten Instance-Typen verwenden. Weitere Informationen finden Sie unter AWS::EC2::Volume.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Amazon VPC Peering-Verbindung	17. Juni 2014	Sie können CloudFormation damit eine Amazon Virtual Private Cloud (Amazon VPC) -Peering-Verbindung erstellen, die eine Netzwerkverbindung zwischen zwei VPCs herstellt. Weitere Informationen finden Sie unter AWS::EC2::VPCPeeringConnection.	15.05.2010
Gruppenupdate EC2 für Amazon Auto Scaling	17. Juni 2014	Sie können eine bestehende Cluster-Platzierungsgruppe angeben, in der Instances für eine Amazon EC2 Auto Scaling Scaling-Gruppe gestartet werden sollen. Weitere Informationen finden Sie unter AWS::AutoScaling::AutoScalingGroup.	15.05.2010
AWS CloudTrail Unterstützung	17. Juni 2014	CloudFormation unterstützt AWS CloudTrail, das API-Aufrufe von Ihrem AWS Konto erfassen und die Protokolle an einem von Ihnen angegebenen Ort veröffentlichen kann. Weitere Informationen finden Sie unter AWS::CloudTrail::Trail.	15.05.2010
Verbesserungen bei der Stack-Aktualisierung	12. Mai 2014	CloudFormation unterstützt zusätzliche Funktionen für die Aktualisierung von Stacks: • Sie können CloudFormation Stack-Parameter aktualisieren, ohne die Vorlage des Stacks erneut einzureichen. • Sie können Amazon-SNS-Benachrichtigungsthemen für einen CloudFormation-Stack hinzuzufügen oder entfernen. Weitere Informationen finden Sie unter CloudFormation Stacks-Updates.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Amazon Kinesis-Unterstützung	6. Mai 2014	Sie können CloudFormation damit Amazon Kinesis Kinesis-Streams erstellen, die Datensätze aus Datenquellen erfassen und transportieren. Weitere Informationen finden Sie unter AWS::Kinesis::Stream.	15.05.2010
Neue S3-Bucket-Eigenschaften	5. Mai 2014	CloudFormation unterstützt zusätzliche S3-Bucket-Eigenschaften: • Cross-Origin Resource Sharing (CORS) definiert die ursprungsübergreifende gemeinsame Nutzung von Objekten in einem Bucket. • Der Lebenszyklus definiert, wie Amazon S3 Objekte während ihrer Lebensdauer verwaltet. • Die Zugriffsprotokollierungsrichtlinie erfasst Informationen zu Anforderungen an das Bucket. • Benachrichtigungen definieren, welche Ereignisse gemeldet werden und an welches Amazon SNS-Thema Nachrichten gesendet werden. • Versioning aktiviert mehrere Varianten aller Objekte in einem Bucket. • Umleitungs- und Routing-Regeln regeln das Umleitverhalten von Anforderungen, die an den Website-Endpunkt eines Buckets gestellt werden. Weitere Informationen finden Sie unter AWS::S3::Bucket.	15.05.2010
Unterstützung für Amazon EC2 Auto Scaling	5. Mai 2014	CloudFormation unterstützt die Erfassung von Metriken für eine Auto Scaling Scaling-Gruppe. Weitere Informationen finden Sie unter AWS::AutoScaling::AutoScalingGroup.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
`Fn::If` update	5. Mai 2014	Sie können die intrinsische `Fn::If`-Funktion im Ausgabeabschnitt einer Vorlage verwenden. Weitere Informationen finden Sie unter Bedingungsfunktionen.	15.05.2010
API-Protokollierung mit AWS CloudTrail	2. April 2014	Sie können AWS CloudTrail (CloudTrail) verwenden, um CloudFormation Anfragen zu protokollieren. Mit können CloudTrail Sie einen Verlauf der CloudFormation API-Aufrufe für Ihr Konto abrufen. Weitere Informationen finden Sie unter Protokollieren von CloudFormation API-Aufrufen mit AWS CloudTrail.	15.05.2010
Elastic Load Balancing update	20. März 2014	Sie können eine Zugriffsprotokollierungsrichtlinie angeben, um Informationen zu Anforderungen an Ihren Load Balancer zu erfassen. Sie können auch eine Connection-Draining-Richtlinie angeben, die beschreibt, wie in Übertragung befindliche Anforderungen gehandhabt werden, wenn Instances abgemeldet oder instabil werden. Weitere Informationen finden Sie unter AWS::ElasticLoadBalancing::LoadBalancer.	15.05.2010
OpsWorks Unterstützung	3. März 2014	Sie können ihn CloudFormation für die Bereitstellung und Verwaltung von OpsWorks Stacks verwenden. Weitere Informationen finden Sie unter AWS::OpsWorks::Stackoder AWS OpsWorks Template Snippets.	15.05.2010
Erhöhung des Amazon S3-Vorlagengrößenlimits	18. Februar 2014	Sie können Vorlagengrößen von bis zu 460.800 Byte in Amazon S3 angeben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Amazon Redshift-Unterstützung	10. Februar 2014	Sie können es CloudFormation für die Bereitstellung und Verwaltung von Amazon Redshift Redshift-Clustern verwenden. Weitere Informationen finden Sie unter Amazon-Redshift-Vorlagenausschnitte oder AWS::Redshift::Cluster.	15.05.2010
Aktualisierung von S3-Buckets und Bucket-Richtlinien	10. Februar 2014	Sie können einige Eigenschaften der S3-Bucket- und Bucket-Richtlinienressourcen aktualisieren. Weitere Informationen finden Sie unter AWS::S3::Bucket oder AWS::S3::BucketPolicy.	15.05.2010
Aktualisierung von Elastic Beanstalk-Umgebungen und Anwendungsversionen	10. Februar 2014	Sie können die Elastic Beanstalk-Umgebungskonfigurationen und Anwendungsversionen aktualisieren. Weitere Informationen finden Sie unter AWS::ElasticBeanstalk::Environment, AWS::ElasticBeanstalk::ConfigurationTemplate oder AWS::ElasticBeanstalk::ApplicationVersion.	15.05.2010
Amazon SQS update	29. Januar 2014	Sie können eine Warteschlange für unzustellbare Nachrichten für eine Amazon SQS-Warteschlange angeben. Weitere Informationen finden Sie unter AWS::SQS::Queue.	15.05.2010
Geplante Auto Scaling-Aktionen	27. Januar 2014	Sie können die Anzahl der EC2 Instances in einer Auto Scaling Scaling-Gruppe nach einem Zeitplan skalieren. Wenn Sie einen Zeitplan verwenden, können Sie Anwendungen entsprechend vorhersagbarer Änderungen bei der Auslastung skalieren. Weitere Informationen finden Sie unter AWS::AutoScaling::ScheduledAction.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Sekundäre DynamoDB-Indizen	27. Januar 2014	Sie können lokale und globale sekundäre Indizen für DynamoDB-Datenbanken erstellen. Durch die Nutzung von sekundären Indizen können Sie effizient auf Daten mit anderen Attributen als dem Primärschlüssel zugreifen . Weitere Informationen finden Sie unter AWS::DynamoDB::Table.	15.05.2010
Auto Scaling update	2. Januar 2014	Sie können eine Instance-ID für eine Auto-Scaling-Gruppe oder Startkonfiguration angeben. Sie können auch zusätzliche Auto Scaling-Blockgeräteigenschaften angeben. Weitere Informationen finden Sie unter AWS::AutoScaling::AutoScalingGroup oder AWS::AutoScaling::LaunchConfiguration.	15.05.2010
Amazon SQS update	2. Januar 2014	Sie können SQS-Warteschlangen aktualisieren und zusätzliche Eigenschaften angeben. Weitere Informationen finden Sie unter AWS::SQS::Queue.	15.05.2010
Limiterhöhungen	2. Januar 2014	Sie können bis zu 60 Parameter und bis zu 60 Ausgaben in Ihren CloudFormation-Vorlagen angeben.	15.05.2010
New console	19. Dezember 2013	Die neue CloudFormation Konsole bietet Funktionen wie die automatische Aktualisierung von Stack-Ereignissen und die alphabetische Reihenfolge der Stack-Parameter.	15.05.2010
Zonenübergreifendes Load Balancing	19. Dezember 2013	Mit zonenübergreifendem Load Balancing können Sie den Datenverkehr über alle Availability Zones hinweg an Back-End-Instances weiterleiten (). AZs Weitere Informationen finden Sie unter AWS::ElasticLoadBalancing::LoadBalancer.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
AWS Elastic Beanstalk Ebenen der Umgebung	19. Dezember 2013	Sie können angeben, ob AWS Elastic Beanstalk Ressourcen zur Unterstützung eines Webservers oder zur Ausführung von Hintergrundverarbeitungsaufgaben bereitgestellt werden. Weitere Informationen finden Sie unter AWS::ElasticBeanstalk::Environment.	15.05.2010
Ressourcennamen	19. Dezember 2013	Sie können den folgenden Ressourcen Namen (physisch IDs) zuweisen: • ElastiCache Cluster • Elastic Load Balancing-Load Balancer • RDS DB-Instances Weitere Informationen finden Sie unter Name Type.	15.05.2010
VPN-Unterstützung	22. November 2013	Sie können ein Virtual Private Gateway (VGW) aktivieren, um Routen an die Routing-Tabellen einer VPC weiterzuleiten. Weitere Informationen finden Sie unter AWS::EC2::VPNGatewayRoutePropagation.	15.05.2010
Bedingte Erstellung von Ressourcen und Zuweisung von Eigenschaften	8. November 2013	Mit Eingabeparametern können Sie die Erstellung und Einstellungen von designierten Stack-Ressourcen steuern, indem Sie Bedingungen in Ihren CloudFormation-Vorlagen definieren. Beispiel: Sie können Bedingungen verwenden, um Stack-Ressourcen für eine Produktionsumgebung zu erstellen. Mit derselben Vorlage können Sie ähnliche Stack-Ressourcen mit geringerer Kapazität für eine Testumgebung erstellen. Weitere Informationen finden Sie unter Bedingungsfunktionen.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Verhindern von versehentlichen Aktualisierungen der Stack-Ressourcen	8. November 2013	Sie können Stack-Aktualisierungen verhindern, die möglicherweise zu unbeabsichtigten Änderungen an Stack-Ressourcen führen. Wenn Sie beispielsweise einen Stack mit einem Datenbank-Layer haben, der nur selten aktualisiert werden sollte, können Sie eine Stack-Richtlinie festlegen, die für die meisten Benutzer eine Aktualisierung dieses Datenbank-Layers verhindert. Weitere Informationen finden Sie unter Verhindern von Aktualisierungen der Stack-Ressourcen.	15.05.2010
Namensressourcen	8. November 2013	Anstatt physisch CloudFormation generierte Ressourcen zu verwenden IDs, können Sie bestimmten Ressourcen Namen zuweisen. Die folgenden CloudFormation Ressourcen unterstützen die Benennung • CloudWatch Amazon-Alarme • DynamoDB-Tabellen • AWS Elastic Beanstalk Anwendungen und Umgebungen • Amazon-S3-Buckets • Amazon SNS-Themen • Amazon SQS-Warteschlangen Weitere Informationen finden Sie unter Name Type.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Zuweisen von benutzerdefinierten Ressourcentypen	8. November 2013	In Ihren Vorlagen können Sie Ihren eigenen Ressourcentyp für benutzerdefinierte CloudFormation-Ressourcen angeben (`AWS::CloudFormation::CustomResource`). Durch die Verwendung Ihres eigenen benutzerdefinierten Ressourcentypnamens können Sie schnell identifizieren, welcher Typ benutzerdefinierter Ressourcen in Ihrem Stack vorhanden ist. Sie können beispielsweise `"Type": "Custom::` *`MyCustomResource`* `"` angeben. Weitere Informationen finden Sie unter AWS::CloudFormation::CustomResource.	15.05.2010
Hinzufügen von Pseudoparametern	8. November 2013	Sie können jetzt auf die AWS AccountID in CloudFormation Vorlagen verweisen, indem Sie auf den `AWS::AccountID` Pseudo-Parameter verweisen . Weitere Informationen finden Sie unter Pseudoparameter-Referenz.	15.05.2010
Angeben von Stacks in IAM-Richtlinien	8. November 2013	Sie können IAM-Benutzern, -Gruppen oder -Rollen erlauben oder verweigern, auf bestimmten Stacks zu arbeiten. CloudFormation Beispielsweise können Sie die Aktion zum Löschen des Stacks für eine bestimmte Stack-ID verweigern. Weitere Informationen finden Sie unter Zugriffskontrolle mit AWS Identity and Access Management.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Verbundunterstützung	14. Oktober 2013	CloudFormation unterstützt temporäre Sicherheitsanmeldedaten von IAM-Rollen, die Szenarien wie Federation und Single Sign-On bei der ermöglichen. AWS Management Console Mithilfe CloudFormation von IAM-Rollen können Sie auch EC2 Instances aufrufen, ohne langfristige Sicherheitsanmeldedaten einzubetten. Weitere Informationen zu CloudFormation und IAM finden Sie unter Zugriffskontrolle mit AWS Identity and Access Management.	15.05.2010
Amazon RDS Read Replica-Unterstützung	24. September 2013	Sie können jetzt Amazon RDS Read Replicas von einer DB-Quell-Instance erstellen. Weitere Informationen finden Sie unter der `SourceDBInstanceIdentifier` -Eigenschaft in der AWS::RDS::DBInstance-Ressource.	15.05.2010
Verknüpfen von öffentlichen IP-Adressen mit Instances in einer Auto-Scaling-Gruppe	19. September 2013	Sie können jetzt öffentliche IP-Adressen mit Instances in einer Auto-Scaling-Gruppe verknüpfen. Weitere Informationen finden Sie unter AWS::AutoScaling::LaunchConfiguration.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Zusätzliche VPC-Unterstützung	17. September 2013	CloudFormation fügt mehrere Verbesserungen zur Unterstützung von VPC- und VPN-Funktionen hinzu • Sie können EC2 Amazon-Netzwerkschnittstellen eine öffentliche IP-Adresse und mehrere private IP-Adressen zuordnen. Weitere Informationen finden Sie unter AWS::EC2::NetworkInterface. Sie können auch eine primäre private IP-Adresse mit einer elastischen IP-Adresse (EIP) verknüpfen. • Sie können DNS-Unterstützung aktivieren und DNS-Hostnamen angeben. Weitere Informationen finden Sie unter AWS::EC2::VPC. • Sie können eine statische Route zwischen einem Virtual Private Gateway und Ihrem VPN-Gateway angeben. Weitere Informationen finden Sie unter AWS::EC2::VPNConnectionRoute.	15.05.2010
Unterstützung von Redis- und VPC-Sicherheitsgruppen für Amazon ElastiCache	3. September 2013	Sie können jetzt Redis als Cache-Engine für einen Amazon ElastiCache (ElastiCache) -Cluster angeben. Sie können jetzt auch VPC-Sicherheitsgruppen ElastiCache Clustern zuweisen. Weitere Informationen finden Sie unter AWS::ElastiCache::CacheCluster.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Parallele Stack-Erstellung, Aktualisierung und Löschung und verschachtelte Stack-Aktualisierungen	12. August 2013	AWS CloudFormation erstellt, aktualisiert und löscht jetzt Ressourcen parallel, wodurch die Leistung der Operationen verbessert wird. Wenn Sie eine Vorlage der obersten Ebene aktualisieren, aktualisiert AWS CloudFormation verschachtelte Stacks, die geändert wurden, automatisch. Weitere Informationen finden Sie unter Stacks-Updates. CloudFormation	15.05.2010
VPC-Sicherheitsgruppen können jetzt in RDS-DB-Instances festgelegt werden	28. Februar 2013	Sie können jetzt mit CloudFormation VPC-Sicherheitsgruppen zu einer RDS-DB-Instance zuweisen. Weitere Informationen finden Sie unter der Eigenschaft VPCSecurityGroups unter AWS::RDS::DBInstance.	15.05.2010
Fortlaufende Bereitstellungen für Amazon EC2 Auto Scaling Scaling-Gruppen	20. Februar 2013	CloudFormation unterstützt jetzt Aktualisierungsrichtlinien für Amazon EC2 Auto Scaling Scaling-Gruppen, die beschreiben, wie Instances in der Amazon EC2 Auto Scaling Scaling-Gruppe ersetzt oder geändert werden, wenn die Amazon EC2 Auto Scaling Scaling-Gruppe Instances hinzufügt oder entfernt. Sie können diese Einstellungen während der Stack-Erstellung oder bei einer Stack-Aktualisierung ändern. Weitere Informationen und ein Beispiel finden Sie unter UpdatePolicyAttribut.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Abbrechen und Rollback-Aktion für Stack-Aktualisierungen	20. Februar 2013	CloudFormation unterstützt die Möglichkeit, ein Stack-Update abzubrechen. Der Stack muss sich im UPDATE_IN_PROGRESS-Status befinden, wenn die Aktualisierungsanforderung gestellt wird. Weitere Informationen finden Sie unter den folgenden Themen: • Stornieren einer Stack-Aktualisierung • cancel-update-stack • CancelUpdateStack in der CloudFormation -API-Referenz	15.05.2010
EBS-optimierte Instances für Amazon EC2 Auto Scaling Scaling-Gruppen	20. Februar 2013	Sie können jetzt EBS-optimierte Instances in Amazon EC2 Auto Scaling Scaling-Gruppen für einen dedizierten Durchsatz für Amazon Elastic Block Store (Amazon EBS) in automatisch skalierten Instances bereitstellen. Die Implementierung ähnelt der der zuvor veröffentlichten Unterstützung für optimierte Amazon EC2 EBS-Instances. Weitere Informationen finden Sie unter der neuen `EbsOptimized` -Eigenschaft in AWS::AutoScaling::LaunchConfiguration.	15.05.2010
Neue Dokumentation	21. Dezember 2012	AWS::EC2::Instancebietet jetzt eine `BlockDeviceMappings` Eigenschaft, mit der Sie Blockgerätezuordnungen für Ihre Instance festlegen können. EC2 Mit dieser Änderung wurden zwei neue Typen hinzugefügt: • BlockDeviceMapping • Blockgerät	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue Dokumentation	21. Dezember 2012	Es wurden neue Abschnitte zur Beschreibung der Verfahren für das Erstellen und Anzeigen von Stacks mit der kürzlich neu gestalteten AWS Management Console hinzugefügt. Sie finden diese hier: • Einen Stack von der Konsole aus erstellen CloudFormation • Stack-Informationen von der CloudFormation Konsole aus anzeigen	15.05.2010
Neue Dokumentation	15. November 2012	Informationen zu benutzerdefinierten Ressourcen finden Sie in den folgenden Themen: • Angepasste Ressourcen • AWS::CloudFormation::CustomResource • Referenz für benutzerdefinierte Ressourcen	15.05.2010
Aktualisierte Dokumentation	15. November 2012	CloudFormation unterstützt jetzt die Angabe bereitgestellter I/O Operationen pro Sekunde (IOPS) für RDS-DB-Instances. Sie können einen Wert von 1 000 bis 10 000 in 1 000-IOPS-Schritten festlegen, indem Sie die neue Iops-Eigenschaft in AWS::RDS::DBInstance verwenden. Weitere Informationen über das Festlegen von IOPS für RDS-DB-Instances finden Sie unter Provisioned IOPS (Bereitgestellte IOPS) im Amazon Relational Database Service User Guide (Amazon Relational Database Service-Benutzerhandbuch).	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neue und aktualisierte Dokumentation	27. August 2012	Die Themen wurden neu organisiert, um spezifische Informationen zur Verwendung der AWS Management Console und zur Verwendung der CloudFormation Befehlszeilenschnittstelle (CLI) übersichtlicher bereitzustellen. Informationen zum Taggen von CloudFormation Stacks wurden hinzugefügt, darunter neue Leitfäden und aktualisierte Referenzthemen: • Neues Thema: Stack-Optionen konfigurieren. • Neue Informationen zu Tags finden Sie in der API-Referenz zu CloudFormation : CreateStack, Stack und Tag. Neue Informationen zum Arbeiten mit Windows-Stacks: • Microsoft Windows Amazon Machine Images (AMIs) und AWS CloudFormation Vorlagen • Bootstrapping AWS CloudFormation von Windows-Stacks Neues Thema: Verwendung regulärer Ausdrücke in Vorlagen. CloudFormation	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neues Feature	25. April 2012	CloudFormation bietet jetzt volle Unterstützung für Virtual Private Cloud (VPC) -Sicherheit bei Amazon EC2. Sie können jetzt mithilfe einer einzigen Vorlage eine gesamte VPC mit allen Arten von VPC-Ressourcen (Subnetze , Gateways, Netzwerk ACLs, Routing-Tabellen usw.) erstellen und auffüllen. CloudFormation Die Dokumentation für die folgenden Ressourcentypen wurde aktualisiert: • AWS::EC2::SecurityGroup • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::Instance • AWS::AutoScaling::AutoScalingGroup • AWS::EC2::EIP • AWS::EC2::EIPAssociation • AWS::ElasticLoadBalancing::LoadBalancer Neue Ressourcentypen wurden der Dokumentation hinzugefügt: • AWS::EC2::VPC • AWS::EC2::InternetGateway • AWS::EC2::DHCPOptions • AWS::EC2::DHCPOptions • AWS::EC2::RouteTable • AWS::EC2::NetworkAcl • AWS::EC2::NetworkAclEntry • AWS::EC2::Subnet	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		• AWS::EC2::VPNGateway • AWS::EC2::CustomerGateway	
Neues Feature	13. April 2012	CloudFormation ermöglicht es Ihnen jetzt, Elemente zu einem Stapel hinzuzufügen oder zu entfernen, wenn Sie ihn aktualisieren. CloudFormation Stacks-Updates wurden aktualisiert und der Komplettlösung wurde ein neuer Abschnitt hinzugefügt: Ändern der Ressourcen des Stacks, in dem beschrieben wird, wie Ressourcen hinzugefügt und entfernt werden, wenn der Stack aktualisiert wird.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neues Feature	2. Februar 2012	CloudFormation bietet jetzt Unterstützung für Ressourcen in einer bestehenden Amazon Virtual Private Cloud (Amazon VPC). Mit dieser Version können Sie: • Starten Sie eine EC2 Dedicated Instance in einer vorhandenen Amazon VPC. Weitere Informationen finden Sie unter AWS::EC2::Instance. • Legen Sie das `SourceDestCheck` Attribut einer EC2 Instance fest, die sich in einer vorhandenen Amazon VPC befindet. Weitere Informationen finden Sie unter AWS::EC2::Instance. • Elastische IP-Adressen in einer vorhandenen Amazon VPC erstellen Weitere Informationen finden Sie unter AWS::EC2::EIP. • Wird verwendet CloudFormation , um Amazon VPC-Sicherheitsgruppen und Eingangs-/Ausgangsregeln in einer vorhandenen VPC zu erstellen. Weitere Informationen finden Sie unter AWS::EC2::SecurityGroup. • Verknüpfen Sie eine Auto-Scaling-Gruppe mit einer bestehenden Amazon VPC verknüpfen, indem Sie die `VPCZoneIdentifier` -Eigenschaft Ihrer `AWS::AutoScaling::AutoScalingGroup` -Ressource festlegen. Weitere Informationen finden Sie unter AWS::AutoScaling::AutoScalingGroup. • Weisen Sie einen Elastic Load Balancing-Load Balancer einem Amazon VPC-Subnetz zu und erstellen Sie Sicherheitsgruppen für den Load Balancer. Weitere Informationen finden Sie unter AWS::ElasticLoadBalancing::LoadBalancer.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
		• Eine RDS-DB-Instance in einer vorhandenen Amazon VPC erstellen Weitere Informationen finden Sie unter AWS::RDS::DBInstance.	
Neues Feature	2. Februar 2012	Sie können jetzt Eigenschaften für die folgenden Ressourcen in einem vorhandenen Stack aktualisieren: • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::EIPAssociation • AWS::RDS::DBSubnetGroup • AWS::RDS::DBSecurityGroup • AWS::RDS::DBSecurityGroupIngress • AWS::Route53::RecordSetGroup Eine vollständige Liste der aktualisierbaren Ressourcen und Einzelheiten darüber, was bei der Aktualisierung eines Stacks zu beachten ist, finden Sie unter Stacks-Updates. CloudFormation	15.05.2010
Neu strukturiertes Handbuch	2. Februar 2012	Bestehende Abschnitte wurden in neue Abschnitte umstrukturiert: Mit CloudFormation Vorlagen arbeiten und Stacks verwalten. Verschieben der Vorlagenreferenz auf die oberste Ebene des Inhaltsverzeichnisses. Die Schätzung der Kosten Ihres CloudFormation Stacks wurde in den Abschnitt Erste Schritte verschoben.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Neuer Inhalt	2. Februar 2012	Neue Abschnitte hinzugefügt: • Schrittweise Anleitung: Aktualisieren eines Stacks ist ein Lernprogramm, das durch den Prozess der Aktualisierung eines LAMP-Stacks führt. • Deploying Applications on Amazon EC2 with AWS CloudFormation beschreibt, wie Sie CloudFormation Hilfsskripte verwenden, um Anwendungen mithilfe von Metadaten bereitzustellen, die in Ihrer Vorlage gespeichert sind. • CloudFormationHelper Scripts Reference bietet Referenzmaterial für die CloudFormation Hilfsskripte (cfn-init cfn-get-metadata, cfn-signal und cfn-hup).	15.05.2010
Neues Feature	26. Mai 2011	CloudFormation stellt jetzt den `list-stacks` Befehl bereit, mit dem Sie nach Stack-Status gefilterte Stacks auflisten können. Gelöschte Stacks können bis zu 90 Tage nach dem Löschen aufgelistet werden. Weitere Informationen finden Sie unter Ihre Stacks beschreiben und auflisten.	15.05.2010
Neue Features	26. Mai 2011	Mit den Befehlen `describe-stack-resources` und `get-template` können Sie jetzt Informationen aus gelöschten Stacks während 90 Tagen nach der Löschung abrufen. Weitere Informationen finden Sie unter Ressourcen auflisten und Vorlage abrufen.	15.05.2010
Neuer Link	1. März 2011	CloudFormation Endpunktinformationen befinden sich jetzt im Allgemeine AWS-Referenz. Weitere Informationen finden Sie unter „Regions and Endpoints “ (Regionen und Endpunkte) in der Allgemeinen AWS - Referenz.	15.05.2010

Änderung	Veröffentlichungsdatum	Beschreibung	API-Version
Erstversion	25. Februar 2011	Die erste öffentliche Veröffentlichung von CloudFormation.	15.05.2010

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.