



Benutzer-Leitfaden

AWS CloudFormation



API-Version 2010-05-15

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS CloudFormation: Benutzer-Leitfaden

Copyright © 2026 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 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 mit CloudFormation	12
Melden Sie sich an für ein AWS-Konto	12
Melden Sie sich für eine an AWS-Konto	12
Erstellen eines Benutzers mit Administratorzugriff	13
Erstellen Sie Ihren ersten Stapel	15
Voraussetzungen	15
Erstellen Sie einen CloudFormation Stack mit der Konsole	16
Stack-Erstellung überwachen	20
Testen Sie den Webserver	21
Fehlerbehebung	21
Bereinigen	22
Nächste Schritte	23
Best Practices	25
Verkürzen Sie die Feedbackschleife, um die Entwicklungsgeschwindigkeit zu erhöhen.	26
Stacks nach Lebenszyklus und Besitz organisieren	27
Verwenden Sie stapelübergreifende Referenzen, um den Wert einer Ausgabe zurückzugeben, die von einem anderen Stapel exportiert wurde	28
Wird CloudFormation StackSets für Bereitstellungen mit mehreren Konten und mehreren Regionen verwendet	29
Prüfen der Kontingente für alle Ressourcentypen	29
Verwenden von Vorlagen zum Replizieren von Stacks in mehreren Umgebungen	29
Verwenden von Modulen, um Ressourcenkonfigurationen wiederzuverwenden	30
Einführung von Infrastruktur als Code	30
Fügen Sie keine Anmeldedaten in Ihre Vorlagen ein	31

Verwenden Sie AWS-spezifische Parametertypen	31
Verwenden von Parameter-Einschränkungen	32
Verwenden Sie Pseudoparameter, um die Portabilität zu fördern.	32
Verwenden von <code>AWS::CloudFormation::Init</code> zur Bereitstellung von Software- Anwendungen auf Amazon-EC2-Instances	33
Verwenden der neuesten Hilfsskripts	33
Validieren von Vorlagen vor ihrer Verwendung	34
Prüfen von Vorlagen auf die Einhaltung von Organisationsrichtlinien	34
Verwendung von YAML oder JSON für die Erstellung von Vorlagen	34
Implementieren Sie eine umfassende Tagging-Strategie	35
Nutzen Sie Vorlagenmakros für erweiterte Transformationen	36
Verwalten Sie alle Stack-Ressourcen über CloudFormation	36
Erstellen von Änderungssätzen vor der Aktualisierung von Stacks	36
Verwenden Sie Stapelrichtlinien zum Schutz von Ressourcen	37
Wird verwendet AWS CloudTrail , um CloudFormation Anrufe zu protokollieren	37
Verwenden von Code-Bewertungen und Revisionskontrollen für die Verwaltung Ihrer Vorlagen	38
Aktualisieren Sie Ihre Amazon-EC2-Instances regelmäßig	38
Verwenden Sie die Drifterkennung regelmäßig	38
Konfigurieren Sie Rollback-Auslöser für die automatische Wiederherstellung	39
Implementieren Sie effektive Stack-Refactoring-Strategien	39
Verwenden Sie CloudFormation Hooks für das Lebenszyklusmanagement	40
Verwenden Sie den IaC-Generator, um Vorlagen aus vorhandenen Ressourcen zu erstellen.	40
AWS Infrastructure Composer Für die visuelle Vorlagengestaltung verwenden	41
Erwägen Sie AWS Cloud Development Kit (AWS CDK) die Verwendung für komplexe Infrastrukturen	41
Verwendung von IAM für die Zugriffskontrolle	42
Wenden Sie das Prinzip des geringsten Privilegs an	42
Sichern Sie sensible Parameter	43
Implementieren Sie Richtlinien als Code mit AWS CloudFormation Guard	43
Arbeiten mit -Vorlagen	44
Wo Vorlagen gespeichert werden	45
Vorlagen validieren	45
Erste Schritte mit Vorlagen	2
Planen Sie, die CloudFormation Vorlagenreferenz zu verwenden	46
Mustervorlagen	47

Format der Vorlage	47
Struktur der Vorlage	48
Kommentare	50
Technische Daten	51
Weitere Informationen	51
Verwenden von regulären Ausdrücken	51
Abschnitte einer Vorlage	53
Resources	54
Parameters	63
Outputs	77
Mappings	83
Metadata	91
Rules	97
Conditions	118
Transform	130
Formatversion	131
Description	131
Infrastructure Composer	132
Warum Infrastructure Composer im CloudFormation Konsolenmodus verwenden?	133
Wie unterscheidet sich dieser Modus von dem der Infrastructure Composer-Konsole?	133
AWS CloudFormation Sprachserver	133
IDEs Integration mit dem Sprachserver AWS CloudFormation	134
Erste Schritte	134
Initialisieren eines CloudFormation Projekts in der IDE	141
Open-Source	142
Benötigen Sie Hilfe?	143
IaC-Generator	143
Überlegungen	144
Berechtigungen	145
Häufig verwendete Befehle	146
Migrieren Sie eine Vorlage auf die AWS CDK	146
Starten Sie eine Ressourcenprüfung	147
Zusammenfassung des Scans anzeigen	151
Erstellen Sie eine Vorlage aus gescannten Ressourcen	152
Erstellen Sie einen Stapel aus gescannten Ressourcen	157
Auflösen von schreibgeschützten Eigenschaften	159

In anderen Diensten gespeicherte Werte abrufen	165
Allgemeine Überlegungen	165
Systemmanager-Klartextwert abrufen	166
Systems Manager sichere Zeichenfolge abrufen	170
Secrets Manager Wert abrufen	173
Werte abrufen AWS	178
Syntax	178
Verfügbare Pseudoparameter	179
Beispiele	181
Stapelausgaben abrufen	185
Exportieren von Stack-Ausgabewerten im Vergleich zur Verwendung geschachtelter	
Stacks	186
Überlegungen	186
Auflistung von exportierten Ausgabewerten	187
Auflistung von Stacks, die einen exportierten Ausgabewert importieren	188
Vorhandene Ressourcen zur Laufzeit spezifizieren	189
-Übersicht	189
Beispiel	190
Überlegungen	194
Unterstützte AWS-spezifische Parametertypen	195
Unterstützte Systems Manager Parametertypen	197
Nicht unterstützte Systems Manager Parametertypen	197
Anleitungen	198
Verweisen Sie auf Ressourcenausgaben in einem anderen CloudFormation Stack	199
Bereitstellen von Anwendungen auf Amazon EC2	204
Einen Stack aktualisieren	214
Erstellen Sie eine skalierte Anwendung mit Load Balancing	231
Verknüpfen mit einer VPC in einem anderen Konto	245
Führen Sie ECS-Bereitstellungen blue/green durch	256
Vorlagentausschnitte	281
General	282
Auto-Scaling	293
AWS Abrechnungskonsolle	325
CloudFormation	329
CloudFront	336
CloudWatch	349

CloudWatch Logs	355
DynamoDB	389
Amazon EC2	393
Amazon ECS	431
Amazon EFS	454
Elastic Beanstalk	480
Elastic Load Balancing	485
IAM	490
AWS Lambda	508
Amazon Redshift	514
Amazon RDS	524
Route 53	533
Amazon S3	541
Amazon SNS	550
Amazon SQS	551
Amazon Timestream	551
Windows-basierte Stapel	580
Bootstrapping Windows-Stack	582
Verwenden Sie von CloudFormation -bereitgestellte Ressourcentypen	590
Benutzerdefinierte Ressourcen	591
Vorlagenmakros	638
Verschachtelte Stacks	657
Bedingungen warten	664
Erstellen Sie wiederverwendbare Ressourcenkonfigurationen mit Modulen	670
Überlegungen zur Verwendung von Modulen	672
Verstehen der Modulversionierung	672
Module in einer Vorlage verwenden	674
Verwenden Sie Parameter, um Modulwerte anzugeben	676
Referenzmodul Ressourcen	678
Stapel erstellen und verwalten	681
Schnittstellen für die Verwaltung Ihrer Stacks	683
Erstellen eines Stacks	684
Erstellen eines Stacks	684
Stapeloptionen konfigurieren	687
Vorschau der Konfiguration Ihres Stacks	690
Stapelinformationen anzeigen	691

Aktualisieren Sie Ihre Stapelvorlage	694
Verstehen des Aktualisierungsverhaltens von Stack-Ressourcen	696
Aktualisieren von Stapeln mithilfe von Änderungssätzen	698
Ein Änderungsset erstellen	700
Ein Änderungsset anzeigen	707
Änderungssätze, die Drift berücksichtigen	716
Ausführen eines Änderungssatzes	723
Löschen eines Änderungssatzes	725
Exemplarische Änderungssätze	727
Änderungssätze für verschachtelte Stacks	740
Validieren Sie Stack-Bereitstellungen	747
So funktioniert die Validierung vor der Bereitstellung	747
Überlegungen	748
Voraussetzungen	748
Validieren Sie eine Stack-Bereitstellung (Konsole)	749
Validieren Sie eine Stack-Bereitstellung (AWS CLI)	750
Arten der Validierung	751
Begrenzte Ressourcen	752
Stapel direkt aktualisieren	766
Eine Stapelaktualisierung abbrechen	770
So brechen Sie eine Stack-Aktualisierung ab (Konsole)	770
So brechen Sie eine Stapelaktualisierung ab (AWS CLI)	771
Löschen eines Stacks	771
Zugehörige Ressourcen	773
Gelöschte Stapel ansehen	773
Fortschritt des Stapels überwachen	774
Stack-Ereignisse ansehen	774
Stack-Ereignisse nach Vorgang anzeigen	783
Diagramm der Zeitachse der Stack-Bereitstellung anzeigen	788
Verstehen Sie die Ereignisse bei der Stapelerstellung	791
Stack-Updates überwachen	793
Rollback einer Aktualisierung fortsetzen	795
Ermitteln Sie die Ursache eines Stapelausfalls	800
Stack-Fehleroptionen	801
Verringern Sie Ihren Stapel bei Alarmverletzungen	812
Rollback-Auslöser bei der Erstellung oder Aktualisierung von Stapeln hinzufügen	813

Rollback-Auslöser zu einem Änderungssatz hinzufügen	814
Rollback-Auslöser für einen Stapel anzeigen	814
Rollback-Auslöser für einen Änderungssatz anzeigen	815
Erkennen Sie nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen	815
Was ist Abweichung?	816
Statuscodes für die Abweichungserkennung	817
Überlegungen zur Erkennung von Abweichungen	820
Drift auf einem ganzen CloudFormation Stack erkennen	822
Erkennen von Abweichungen bei einzelnen Stack-Ressourcen	828
Beheben von Abweichungen mit einer Importoperation	832
AWS Ressourcen importieren	843
Manuelles Importieren von AWS -Ressourcen	844
Automatisch AWS -Ressourcen importieren	883
Wiederherstellen eines Importvorgangs	885
Stapel-Refactoring	890
So funktioniert das Stack-Refactoring	891
Überlegungen zum Stack-Refactoring	892
Voraussetzungen	892
Stacks umgestalten (Konsole)	893
Stacks umgestalten ()AWS CLI	894
Begrenzte Ressourcen	899
Unterstützung für Ressourcentypen	903
Verwenden Sie Schnellerstellungslinks, um Stapel zu erstellen CloudFormation	1122
URL-Format	1122
Beispiel	1124
Erstellen eines Stapels mit Hilfe eines Schnellzugriffs	1124
Beispielbefehle für AWS CLI und PowerShell	1126
CancelUpdateStack	1126
ContinueUpdateRollback	1127
CreateStack	1129
Bereitstellen	1133
DeleteStack	1134
DescribeStackEvents	1135
DescribeStackResource	1137
DescribeStackResources	1138
DescribeStacks	1142

GetTemplate	1146
ListStackResources	1147
ListStacks	1149
UpdateStack	1151
ValidateTemplate	1155
Lokale Artefakte in einen S3-Bucket hochladen	1158
Verwaltung von Stacks mit StackSets	1161
StackSets Konzepte	1162
Administrator- und Ziel-Konten	1163
CloudFormation StackSets	1163
Berechtigungsmodelle für StackSets	1163
Stack-Instances	1164
StackSet Operationen	1165
StackSet Operationsoptionen	1166
Tags (Markierungen)	1168
StackSets Statuscodes	1169
Stack-Instance-Statuscodes	1170
Voraussetzungen	1171
Aktivieren Sie AWS-Regionen , die standardmäßig deaktiviert sind	1172
Selbstverwaltete Berechtigungen erteilen	1173
Aktivieren Sie den vertrauenswürdigen Zugriff	1191
Beginnen Sie mit einer Mustervorlage	1196
Voraussetzungen	1197
Erstellen Sie eine Vorlage StackSet mit einer Beispielvorlage aus der Konsole	1197
Überwachen Sie StackSet die Erstellung	1200
StackSet Ergebnisse anzeigen	1201
Aktualisieren Sie Ihre StackSet	1201
Füge Stapel zu deinem hinzu StackSet	1202
Bereinigen	1203
Nächste Schritte	1205
Erstellen StackSets (selbstverwaltete Berechtigungen)	1205
Erstellen Sie eine StackSet mit selbstverwalteten Berechtigungen (Konsole)	1206
Erstellen Sie eine StackSet mit selbstverwalteten Berechtigungen (AWS CLI)	1208
Erstellen StackSets (vom Service verwaltete Berechtigungen)	1210
Überlegungen	1210
Erstellen Sie eine StackSet mit vom Dienst verwalteten Berechtigungen (Konsole)	1211

Erstellen Sie eine StackSet mit vom Service verwalteten Berechtigungen (AWS CLI)	1214
Automatische Bereitstellungen aktivieren/deaktivieren	1218
Wie automatische Bereitstellungen funktionieren	1219
Überlegungen	1219
Automatische Bereitstellungen aktivieren oder deaktivieren (Konsole)	1220
Automatische Bereitstellungen aktivieren oder deaktivieren (AWS CLI)	1220
Aktualisieren StackSets	1221
Aktualisiere deine StackSet (Konsole)	1221
Aktualisieren Sie Ihre StackSet (AWS CLI)	1224
Stapel hinzufügen zu StackSets	1225
Stapel zu einer StackSet (Konsole) hinzufügen	1226
Füge Stapel zu einem StackSet (AWS CLI) hinzu	1228
Parameter auf Stapeln außer Kraft setzen	1229
Parameter auf Stapeln außer Kraft setzen (Konsole)	1230
Parameter auf Stapeln außer Kraft setzen (AWS CLI)	1232
Stapel löschen von StackSets	1233
Löschen Sie Stacks von Ihrer StackSet (Konsole)	1233
Lösche Stacks von deinem StackSet (AWS CLI)	1235
Löschen StackSets	1237
Lösche eine StackSet (Konsole)	1237
Löschen Sie ein StackSet (AWS CLI)	1238
Löschen von Servicerollen (optional)	1238
Zielkonto-Gates	1239
Voraussetzungen	1240
CloudFormation Vorlagen für die Erstellung von Lambda-Funktionen	1240
Wählen Sie den Gleichzeitigkeitsmodus	1241
So funktionieren die einzelnen Parallelitätsmodi	1242
Wählen zwischen strikter und loser Fehlertoleranz auf Grundlage der Bereitstellungsgeschwindigkeit	1244
Auswahl des Gleichzeitigkeitsmodus (Konsole)	1247
Auswahl des Gleichzeitigkeitsmodus (AWS CLI)	1247
Drift wird erkannt an StackSets	1248
Wie CloudFormation funktioniert die Drifterkennung auf einem StackSet	1248
Erkennen Sie Drift auf einer StackSet (Konsole)	1249
Erkennt Drift auf einem StackSet (AWS CLI)	1251
Stoppen der Drifterkennung auf einem StackSet	1258

Stapel importieren in StackSets	1259
Selbstverwalteter Stack-Import	1260
Service-verwalteter Stack-Import	1263
Stapelimporte rückgängig machen	1269
Best Practices	1270
Definieren der Vorlage	1270
Erstellen oder Hinzufügen von Stacks zum StackSet	1270
Aktualisierung von Stacks in einem StackSet	1271
Mustervorlagen	1271
Fehlerbehebung	1273
Häufige Gründe für Störungen der Stack-Operationen	1273
Wiederholen fehlgeschlagenen Operationen zum Erstellen und Aktualisieren von Stacks ..	1274
Löschen von Stack-Instances schlägt fehl	1275
Stack-Importvorgang schlägt fehl	1275
Anzahl der Stack-Instance-Fehler für StackSets Operationen	1276
Synchronisierung von Stacks mit Git-Quellcode	1283
So funktioniert die Git-Synchronisierung	1284
So funktioniert die Git-Synchronisierung	1284
Kommentare zu Pull-Anforderungen	1286
Stack-Bereitstellungsdatei	1286
CloudFormation Vorlagendatei	1287
Vorlagendefinitions-Repository	1287
Voraussetzungen	1288
Git-Repository	1288
CloudFormation Vorlage	1288
Git Sync Dienst Rolle	1288
IAM-Berechtigungen für Konsolenbenutzer	1292
Erstellen Sie einen Stapel aus dem Repository-Quellcode	1295
Erstellen Sie einen Stapel aus dem Repository-Quellcode	1295
Aktualisieren Sie Ihren Stack aus Ihrem Git-Repository	1298
Kommentare für Pull-Anfragen aktivieren	1299
Status-Dashboard	1299
Git-Sync-Status	1300
Letzte Synchronisierungsereignisse	1301
Unterstützte Stack-Status	1301
Verwaltung von Erweiterungen mit der CloudFormation Registrierung	1306

Zugehörige Dokumentation	1306
Konzepte	1307
Erweiterungstypen	1307
Öffentliche Erweiterungen	1308
Aktivierte Erweiterungen	1308
Anzeigen der verfügbaren und aktivierten Erweiterungen	1309
Öffentliche Erweiterungen	1310
IAM role (IAM-Rolle)	1312
Automatisch neue Versionen von Erweiterungen verwenden	1314
Verwenden Sie Aliase, um auf Erweiterungen zu verweisen	1315
Häufig verwendete AWS CLI Befehle für die Arbeit mit öffentlichen Erweiterungen	1315
Aktivieren Sie eine öffentliche Erweiterung	1316
Eine öffentliche Erweiterung aktualisieren	1321
Deaktivieren Sie öffentliche Erweiterungen	1322
Private Erweiterungen	1323
IAM-Berechtigungen	1324
Häufig verwendete AWS CLI Befehle für die Arbeit mit privaten Erweiterungen	1324
Eine private Nebenstelle registrieren	1324
Deregistrieren Sie private Nebenstellen	1327
Konfigurationsdaten für Erweiterungen bearbeiten	1329
Erforderliche Berechtigungen für die Verwendung dynamischer Referenzen	1329
Konfigurationsdaten für eine Nebenstelle bearbeiten (Konsole)	1330
Konfigurationsdaten für eine Nebenstelle bearbeiten (AWS CLI)	1331
Aufzeichnen von Ressourcentypen in AWS Config	1332
Verhindern, dass vertrauliche Eigenschaften in einem Konfigurationselement aufgezeichnet werden	1333
Fortlaufende Lieferung	1334
Walkthrough: Erstellen einer Pipeline für Test- und Produktions-Stacks	1334
Voraussetzungen	1335
Walkthrough-Übersicht	1335
Schritt 1: Artefakt bearbeiten und in einen S3-Bucket hochladen	1336
Schritt 2: Pipeline-Stack erstellen	1338
Schritt 3: Sehen Sie sich den WordPress Stapel an	1344
Schritt 4: Bereinigen von Ressourcen	1345
Weitere Informationen finden Sie auch unter	1346
Referenz der Konfigurationseigenschaften	1346

Konfigurationseigenschaften (Konsole)	1347
Konfigurationseigenschaften (JSON-Objekt)	1350
Weitere Informationen finden Sie auch unter	1355
CloudFormation Artefakte	1355
Stack-Vorlagendatei	1356
Vorlagenkonfigurationsdatei	1356
Weitere Informationen finden Sie auch unter	1357
Verwenden von Funktionen zum Überschreiben von Parametern mit CodePipeline Pipelines	1358
Fn::GetArtifactAtt	1358
Fn::GetParam	1360
Weitere Informationen finden Sie auch unter	1363
Referenzhandbuch für Vorlagen	1364
Sicherheit	1365
Stapel vor dem Löschen schützen	1366
Benutzer festlegen, die den Beendigungsschutz von Stacks ändern dürfen	1368
Verhindern von Aktualisierungen der Stack-Ressourcen	1368
Beispiel-Stack-Richtlinie	1369
Definieren einer Stack-Richtlinie	1370
Festlegen einer Stack-Richtlinie	1375
Aktualisieren geschützter Ressourcen	1376
Ändern einer Stack-Richtlinie	1379
Weitere Beispiel-Stack-Richtlinien	1380
Datenschutz	1384
Verschlüsselung im Ruhezustand	1385
Verschlüsselung während der Übertragung	1385
Richtlinie für den Datenverkehr zwischen Netzwerken	1386
Zugriffskontrolle mit IAM	1386
Definition identitätsbasierter IAM-Richtlinien für CloudFormation	1386
Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation	1396
Verwaltung von Anmeldeinformationen für Anwendungen, die auf Amazon EC2-Instances laufen	1397
Gewährung von temporärem Zugang (Verbundzugang)	1398
Beispiele für identitätsbasierte Richtlinien	1399
AWS verwaltete Richtlinien	1407
CloudFormation Servicerolle	1410
Serviceübergreifende Confused-Deputy-Prävention	1411

FAS-Anträge und Genehmigungsprüfung	1414
Protokollieren von API-Aufrufen	1416
CloudFormation Informationen in CloudTrail	1417
CloudFormation Logdateieinträge verstehen	1418
Sicherheit der Infrastruktur	1422
Ausfallsicherheit	1422
Compliance-Validierung	1423
Konfigurations- und Schwachstellenanalyse	1423
Bewährte Methoden für die Gewährleistung der Sicherheit	1423
Verwendung von IAM für die Zugriffskontrolle	1424
Keine Anmeldeinformationen in Vorlagen einbetten	1424
Wird verwendet AWS CloudTrail , um CloudFormation Anrufe zu protokollieren	1425
AWS PrivateLink	1425
Überlegungen zu CloudFormation VPC-Endpunkten	1425
Erstellen eines Schnittstellen-VPC-Endpunkts für CloudFormation	1427
Erstellen einer VPC-Endpunktrichtlinie für CloudFormation	1427
Überwachung mit EventBridge	1429
CloudFormation und Übersicht über Git-Sync-Ereignisse	1430
Berechtigungen	1432
Erstellen eines benutzerdefinierten Ereignismusters	1432
Detailreferenz zu Ereignissen	1433
Ressource Statusänderung	1435
Stapel-Statusänderung	1437
Drift Detection Statusänderung	1439
StackSet Änderung des Status	1442
StackSet Änderung des Status der Stack-Instance	1443
StackSetÄnderung des Betriebsstatus	1446
Repository-Synchronisierungsstatus ändern	1448
Ressource Synchronisationsstatus ändern	1451
Kontingente	1454
Verfügbarkeit von Features	1463
StackSets und Makros	1463
Fehlerbehebung	1464
Anleitung zur Fehlerbehebung	1464
Behebung von Fehlern	1465
Stack-Fehler löschen	1466

Abhängigkeitsfehler	1467
AWS Config und AWS Systems Manager Konflikte	1467
Fehler bei der Analyse der Parameter bei der Übergabe einer Liste	1468
IAM Unzureichende Berechtigungen	1468
Ungültiger Wert oder nicht unterstützte Ressourceneigenschaft	1468
Kontingent überschritten	1469
Rollback-Fehler bei verschachtelten Stapeln	1469
Keine Aktualisierungen zur Ausführung vorhanden	1470
Ressource konnte während eines Erstellens, Aktualisierens oder Löschens des Stack- Vorgangs nicht stabilisiert werden.	1470
Sicherheitsgruppe ist in VPC nicht vorhanden.	1471
Rollback-Aktualisierung fehlgeschlagen	1471
Warte-Bedingung erhielt nicht die erforderliche Anzahl an Signalen von einer Amazon EC2- Instance	1473
Ressource wurde vom Stack entfernt, aber nicht gelöscht.	1474
Den -Support kontaktieren	1475
Fehlerbehebung mit Amazon Q	1476
Features	1476
Funktionsweise	1477
Preisgestaltung	1477
Regionale Verfügbarkeit	1477
Einschränkungen	1477
Dokumentverlauf	1478
Archivierte Updates	1508
.....	mdcxxxix

Was ist CloudFormation?

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 Amazon EC2 EC2-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, sondern 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 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 CLI](#) und [AWS SDKs](#), und über verschiedene Integrationen verfügbar.

Eine Einführung zu finden Sie CloudFormation unter [Wie CloudFormation funktioniert](#).

Informationen zum Einstieg in die Nutzung CloudFormation finden Sie unter [Erstellen Sie Ihren ersten Stapel](#).

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-Managementkonsole 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 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](#)
- [Erstellen Sie Ihren ersten Stapel](#)

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 mit CloudFormation](#)

Die wichtigsten Konzepte

Bei der Verwendung 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

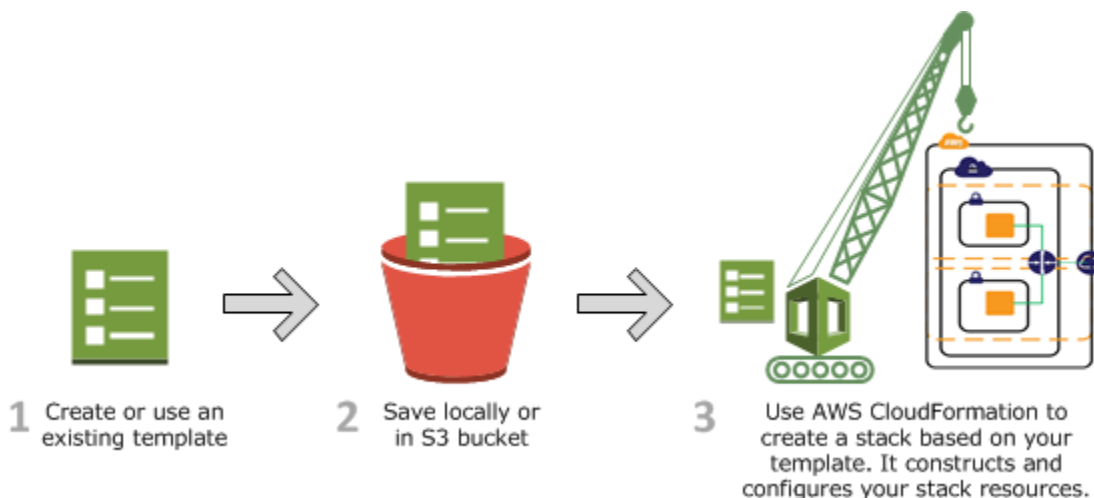
Wenn Sie Änderungen an den ausgeführten Ressourcen in einem Stack vornehmen müssen, aktualisieren Sie den Stack. Bevor Sie Änderungen an Ihren Ressourcen vornehmen, können Sie einen Änderungssatz generieren, eine Zusammenfassung der von Ihnen vorgeschlagenen Änderungen. Anhand von Änderungssätzen können 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 CloudFormation getätigt werden, werden alle von Ihrer Vorlage deklariert. Nehmen wir beispielsweise 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 die Amazon EC2 Create Instance API auf und gibt den Instance-Typ als `ant2.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 [Infrastructure Composer](#), um Ihre Vorlage zu visualisieren und zu validieren. So können Sie sicherstellen, dass Ihre Vorlage richtig 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-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

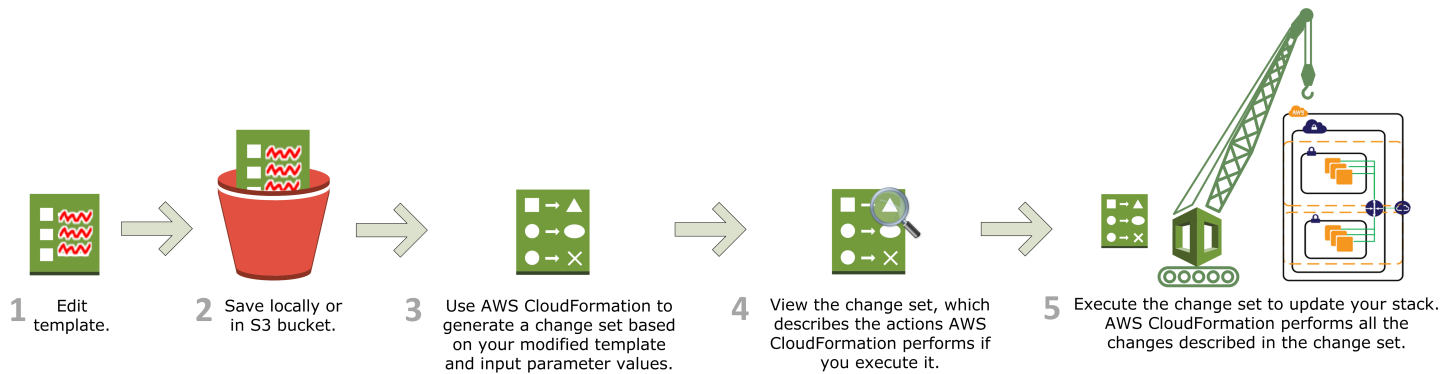
Wenn Sie eine lokal gespeicherte Vorlagendatei angeben, CloudFormation lädt sie in einen S3-Bucket in Ihrem AWS Konto hoch. CloudFormation erstellt einen Bucket für jede Region, in die Sie eine Vorlagendatei hochladen. Auf die Buckets kann jeder zugreifen, der über Amazon Simple Storage Service (Amazon 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.

Aktualisieren eines Stapels 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.



So aktualisieren Sie einen Stapel mit einem Änderungssatz

1. Sie können eine CloudFormation Stapelvorlage mithilfe [Infrastructure Composer](#) eines Texteditors ändern. Weitere Informationen finden Sie unter [Aktualisieren Sie Ihre Stapelvorlage](#).

Wenn Sie Ihre Vorlage aktualisieren, denken Sie 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 des Aktualisierungsverhaltens 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 zum Beispiel nicht, ob Sie ein Konto [Kontingent](#) überschreiten, ob Sie eine Ressource aktualisieren, die keine Aktualisierungen unterstützt, oder ob Sie nicht über ausreichende [Berechtigungen](#) zum Ändern einer Ressource verfügen, was dazu führen kann, dass eine Stapelaktualisierung 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 mit CloudFormation

Informationen zum Erstellen eines Hello CloudFormation World-Stacks mit der Konsole finden Sie unter [Erstellen Sie Ihren ersten Stapel](#).

Wenn Sie eine Anleitung benötigen, sollten Sie den Workshop [Erste Schritte mit CloudFormation](#) besuchen, in dem Sie praktische Erfahrungen mit der Entwicklung von Vorlagen sammeln können.

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 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-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer 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-Managementkonsole](#) als 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 über die Verwaltung der Zugriffsrechte finden Sie unter [Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management](#).

Erstellen Sie Ihren ersten Stapel

Dieses Thema führt Sie durch die Erstellung Ihres ersten CloudFormation Stacks mit dem AWS-Managementkonsole. 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 von 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](#)
- [Stack-Erstellung überwachen](#)
- [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 Verwendung von Amazon EC2, Amazon S3 und/oder verfügt CloudFormation, oder Administratorzugriff.
- Sie müssen über eine Virtual Private Cloud (VPC) verfügen, die Zugang zum Internet hat. Diese exemplarische Vorlage erfordert eine Standard-VPC, die automatisch mit neuer AWS-Konten

geliefert wird. 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 Stack erstellen aus.
3. Wählen Sie auf der Seite Stack erstellen die Option Von Infrastructure Composer erstellen und dann In Infrastructure Composer erstellen. Dadurch gelangen Sie im CloudFormation Konsolenmodus zu Infrastructure Composer, wo Sie die Beispielvorlage hochladen und validieren können.
4. Um die Beispielvorlage hochzuladen und zu validieren, gehen Sie wie folgt vor:
 - a. Choose Template. 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.

- Der Abschnitt **Parameters** deklariert Werte, die bei der Erstellung des Stapels an die Vorlage übergeben werden können. Später in der Vorlage angegebene Ressourcen verweisen auf diese Werte und verwenden die Daten. Parameter sind eine effektive Möglichkeit, 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, einzigartig sein könnten.
- 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-Instancetyps (Standard: `t2.micro` zulässig: `t3.micro` `t2.micro`).
 - **MyIP**- Gibt den IP-Adressbereich für den HTTP-Zugriff an (Standard: `0.0.0.0/0`, was den Zugriff von jeder IP erlaubt).
- Der **Resources**Abschnitt enthält die Definitionen der AWS Ressourcen, die Sie mit der Vorlage erstellen möchten. Ressourcendeklarationen 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 zur Erstellung eines Stapels verwenden. Sie können auch neue Stapel mit der gleichen Vorlage erstellen, um identische Konfigurationen von Ressourcen 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-Instance mit der folgenden Konfiguration:
 - Verwendet das neueste Amazon Linux 2 AMI
 - Wendet den ausgewählten Instancetyp an
 - Fügt die Eigenschaft `WebServerSecurityGroup` zur Eigenschaft `SecurityGroupIds` hinzu
 - Enthält ein Skript mit Benutzerdaten zur Installation von Apache HTTP Server
- Am Anfang jeder Ressourcen- und Parameterdeklaration wird ein logischer Name angegeben. Zum Beispiel ist `WebServerSecurityGroup` der logische Name, der der EC2-Sicherheitsgruppenressource zugewiesen ist. Die Funktion `Ref` wird dann verwendet, um in anderen Teilen der Vorlage auf Ressourcen und Parameter über ihre

logischen Namen zu verweisen. Wenn eine Ressource auf eine andere Ressource verweist, entsteht eine Abhängigkeit zwischen ihnen.

- Der Abschnitt **Outputs** definiert benutzerdefinierte Werte, die nach der Erstellung des Stacks zurückgegeben werden. Sie können die Ausgabewerte verwenden, um Informationen aus den Ressourcen im Stapel zurückzugeben, z. B. Ressourcen-IDs oder URLs.
 - Die Vorlage definiert eine Ausgabe:
 - **WebsiteURL**- Die URL des bereitgestellten Webservers, die unter Verwendung des öffentlichen DNS-Namens der EC2-Instance erstellt wird. Mit der Funktion Join können Sie die feste Variable `http://` und die Variable `PublicDnsName` zu einer einzigen Zeichenkette zusammenfassen, was die Ausgabe der vollständigen URL des Webservers erleichtert.
- b. Wählen Sie Validieren um sicherzustellen, dass der YAML-Code gültig ist, bevor Sie die Vorlage hochladen.
 - c. Wählen Sie anschließend Vorlage erstellen um die Vorlage zu erstellen und zu einem S3-Bucket hinzuzufügen.
 - d. Notieren Sie sich in dem sich öffnenden Dialogfenster 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. Auf der Seite Stapel erstellen wählen Sie Weiter.
 6. Geben Sie auf der Seite Stapeldetails angeben einen Namen in das Feld Stapelname ein. Der Stack-Name darf keine Leerzeichen enthalten. Wählen Sie für dieses Beispiel **MyTestStack**.
 7. Unter Parametergeben Sie die Parameterwerte wie folgt an:
 - LatestAmild: Dies ist standardmäßig auf das neueste Amazon Linux 2-AMI eingestellt.
 - InstanceType: Wählen Sie entweder t2.micro oder t3.micro als EC2-Instance-Typ.

 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).

- MeinIP: Geben Sie Ihre aktuelle öffentliche IP-Adresse mit dem Suffix /32 an. Das Suffix /32 wird in der CIDR-Notation verwendet, um anzugeben, dass eine einzelne IP-Adresse erlaubt ist. Das bedeutet im Wesentlichen, dass Sie den Datenverkehr zu und von dieser bestimmten IP-Adresse zulassen, und keine anderen.
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 so belassen, wie sie sind.
 9. Überprüfen Sie die Informationen für den Stack. Wenn Sie mit den Einstellungen zufrieden sind, klicken Sie auf Submit (Absenden).

Stack-Erstellung überwachen

Nachdem Sie „Senden“ 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 CloudFormation-Konsole angezeigt. Sein Status sollte `CREATE_IN_PROGRESS` sein. 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-Instance führt z. B. 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 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 auf die Registerkarte Ressourcen gehen, um die von Ihnen erstellte EC2-Instance und Sicherheitsgruppe anzuzeigen.

Testen Sie den Webserver

Nachdem der Stack erfolgreich erstellt wurde, navigieren Sie in der CloudFormation Konsole zur Registerkarte Ausgaben. Suchen Sie das Feld WebsiteURL . Diese enthält die öffentliche URL Ihrer EC2-Instance.

Öffnen Sie einen Browser und gehen Sie zu der unter WebsiteURLaufgeführten URL. Sie sollten eine einfache "Hello World!"-Meldung im Browser sehen.

Dies bestätigt, dass Ihre EC2-Instance den Apache HTTP Server ausführt und eine einfache Webseite bereitstellt.

Fehlerbehebung

Wenn bei der Stack-Erstellung ein Rollback auftritt, kann dies an einem fehlenden VPC liegen. Hier erfahren Sie, wie Sie dieses Problem beheben können.

Keine Standard-VPC verfügbar

Die Vorlage in dieser Anleitung erfordert eine Standard-VPC. Wenn Ihre Stack-Erstellung aufgrund von Fehlern bei der VPC- oder Subnetz-Verfügbarkeit fehlschlägt, haben Sie möglicherweise keine Standard-VPC in Ihrem Konto. Ihnen stehen folgende Optionen zur Verfügung:

- Erstellen einer neuen Standard-VPC- Sie können eine neue Standard-VPC über die Amazon VPC-Konsole erstellen. Eine Anleitung 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 Ressource `WebServer`, um die Subnetz-ID aufzunehmen:

```
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
```

Wenn Sie den Stack erstellen, müssen Sie ein Subnetz angeben, das über einen Internetzugang verfügt, 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 Stapel die Option neben dem Namen des von Ihnen erstellten Stapels (**MyTestStack**) und wählen Sie dann Löschen.
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).
4. Überwachen Sie den Fortschritt des Stapel-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 die Arbeit mit der Beispielvorlage beendet haben und Ihren Amazon S3-Bucket nicht mehr benötigen, löschen Sie ihn. Bevor Sie einen Bucket löschen können, müssen Sie ihn zunächst leeren. Wenn Sie einen Bucket leeren, werden alle darin enthaltenen Objekte gelöscht.

So leeren und löschen Sie den Amazon 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 Liste Buckets die Option neben dem Namen des Buckets, den Sie für dieses Tutorial erstellt haben, und wählen Sie dann Leer.
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 und wählen Sie dann Löschen.
8. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den Namen des Buckets ein und wählen Sie Bucket löschen.
9. Überwachen Sie in der Liste Buckets den Fortschritt des Löschvorgangs für den Bucket. Wenn Amazon S3 die Löschung 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 Workshop [Erste Schritte mit CloudFormation](#) für weitere praktische Übungen zur Erstellung von Vorlagen.
- Eine verkürzte Version von [Getting Started with CloudFormation](#) finden Sie unter [Bereitstellen von Anwendungen auf Amazon EC2](#). In diesem Thema wird dasselbe Szenario beschrieben, in dem ein CloudFormation Hilfsskript verwendet wird `cfn-init`, um eine Amazon EC2 EC2-Instance zu booten.

CloudFormation bewährte Verfahren

Bewährte Verfahren sind Empfehlungen, die Ihnen helfen können, den gesamten Arbeitsablauf CloudFormation effektiver zu nutzen und sichere Verfahren einzuführen. 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 Sie die Feedbackschleife, um die Entwicklungsgeschwindigkeit zu erhöhen.](#)
- [Stacks nach Lebenszyklus und Besitz organisieren](#)
- [Verwenden Sie stapelübergreifende Referenzen, um den Wert einer Ausgabe zurückzugeben, die von einem anderen Stapel exportiert wurde](#)
- [Wird CloudFormation StackSets für Bereitstellungen mit mehreren Konten und mehreren Regionen verwendet](#)
- [Verwenden von Vorlagen zum Replizieren von Stacks in mehreren Umgebungen](#)
- [Prüfen der Kontingente für alle Ressourcentypen](#)
- [Verwenden von Modulen, um Ressourcenkonfigurationen wiederzuverwenden](#)
- [Einführung von Infrastruktur als Code](#)

Erstellung von Vorlagen

- [Fügen Sie keine Anmeldedaten in Ihre Vorlagen ein](#)
- [Verwenden Sie AWS-spezifische Parametertypen](#)
- [Verwenden von Parameter-Einschränkungen](#)
- [Verwenden Sie Pseudoparameter, um die Portabilität zu fördern.](#)
- [Verwenden von `AWS::CloudFormation::Init` zur Bereitstellung von Software-Anwendungen auf Amazon-EC2-Instances](#)
- [Verwenden der neuesten Hilfsskripts](#)
- [Validieren von Vorlagen vor ihrer Verwendung](#)
- [Verwendung von YAML oder JSON für die Erstellung von Vorlagen](#)
- [Implementieren Sie eine umfassende Tagging-Strategie](#)
- [Nutzen Sie Vorlagenmakros für erweiterte Transformationen](#)

Verwaltung von Stacks

- [Verwalten Sie alle Stack-Ressourcen über CloudFormation](#)
- [Erstellen von Änderungssätzen vor der Aktualisierung von Stacks](#)
- [Verwenden Sie Stapelrichtlinien zum Schutz von Ressourcen](#)
- [Wird verwendet AWS CloudTrail , um CloudFormation Anrufe zu protokollieren](#)
- [Verwenden von Code-Bewertungen und Revisionskontrollen für die Verwaltung Ihrer Vorlagen](#)
- [Aktualisieren Sie Ihre Amazon-EC2-Instances regelmäßig](#)
- [Verwenden Sie die Drifterkennung regelmäßig](#)
- [Konfigurieren Sie Rollback-Auslöser für die automatische Wiederherstellung](#)
- [Implementieren Sie effektive Stack-Refactoring-Strategien](#)
- [Verwenden Sie CloudFormation Hooks für das Lebenszyklusmanagement](#)

Autorentools

- [Verwenden Sie den IaC-Generator, um Vorlagen aus vorhandenen Ressourcen zu erstellen.](#)
- [AWS Infrastructure Composer Für die visuelle Vorlagengestaltung verwenden](#)
- [Erwägen Sie AWS Cloud Development Kit \(AWS CDK\) die Verwendung für komplexe Infrastrukturen](#)

Sicherheit und Compliance

- [Verwendung von IAM für die Zugriffskontrolle](#)
- [Wenden Sie das Prinzip des geringsten Privilegs an](#)
- [Sichern Sie sensible Parameter](#)
- [Implementieren Sie Richtlinien als Code mit AWS CloudFormation Guard](#)

Verkürzen Sie die Feedbackschleife, um die Entwicklungsgeschwindigkeit zu erhöhen.

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 [CloudFormation Linter](#) (cfn-lint) und [TaskCat](#) Befehlszeilentools. [Das cfn-lint Tool bietet Ihnen die Möglichkeit, Ihre CloudFormation Vorlagen anhand der Ressourcenspezifikation zu validieren.](#) CloudFormation Dazu gehört das Überprüfen gültiger Werte für Ressourceneigenschaften, sowie bewährte Methoden. Plugins für cfn-lint sind [für eine Reihe von Code-Editoren verfügbar](#) Sie können sich auch dafür entscheiden, cfn-lint in die Konfiguration Ihres Quellcode-Repositorys zu integrieren, so dass Sie eine Vorlagenüberprüfung durchführen können, wenn Sie Ihre Beiträge übertragen. Weitere Informationen finden Sie unter [Git Pre-Commit-Validierung von 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.](#) 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 Ingenieuren vor, das eine Website besitzt, die auf Amazon EC2 Auto 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 Sie stapelübergreifende Referenzen, um den Wert einer Ausgabe zurückzugeben, die von einem anderen Stapel exportiert wurde

Wenn Sie Ihre AWS Ressourcen auf der Grundlage von Lebenszyklus und Besitz organisieren, möchten Sie vielleicht einen Stapel 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 stapelübergreifende Referenzen, um den Wert einer von einem anderen Stapel exportierten Ausgabe zurückzugeben, damit andere Stapel sie verwenden können. 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](#).

Wird CloudFormation StackSets für Bereitstellungen mit mehreren Konten und mehreren Regionen verwendet

CloudFormation StackSets erweitern Sie die Funktionalität von Stacks, indem Sie es Ihnen ermöglichen, Stacks für mehrere Konten und Regionen mit einem einzigen Vorgang zu erstellen, zu aktualisieren oder zu löschen. Verwenden Sie es StackSets für die Bereitstellung allgemeiner Infrastrukturkomponenten, Compliance-Kontrollen oder gemeinsam genutzter Dienste in Ihrem Unternehmen.

Implementieren Sie bei der Verwendung StackSets serviceverwaltete Berechtigungen AWS Organizations für eine vereinfachte Berechtigungsverwaltung. Dieser Ansatz ermöglicht die Bereitstellung StackSets für Konten innerhalb Ihrer Organisation, ohne dass Sie die IAM-Rollen in jedem Konto manuell konfigurieren müssen.

Weitere Informationen finden StackSets Sie unter [StackSets Konzepte](#).

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 sein, wie Best Practices zum Definieren 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 in 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 können. Weitere Informationen finden Sie unter [Erstellen Sie wiederverwendbare Ressourcenkonfigurationen, die in allen Vorlagen mit CloudFormation Modulen enthalten sein können](#).

Einführung von Infrastruktur als Code

Behandeln Sie Ihre CloudFormation Vorlagen wie Code, indem Sie Infrastructure-as-Code-Praktiken (IaC) implementieren. Speichern Sie Ihre Vorlagen in Versionskontrollsystemen, führen Sie Code-Reviews durch und verwenden Sie automatisierte Tests, um Änderungen zu validieren. Dieser Ansatz gewährleistet Konsistenz, verbessert die Zusammenarbeit und bietet einen Audit Trail für Infrastrukturänderungen.

Erwägen Sie die Implementierung von CI/CD Pipelines für Ihren Infrastrukturcode, um das Testen und Bereitstellen Ihrer CloudFormation Vorlagen zu automatisieren. Tools wie AWS CodePipeline

AWS CodeBuild, und AWS CodeDeploy können verwendet werden, um automatisierte Workflows für Ihre Infrastrukturbereitstellungen zu erstellen.

Weitere Informationen zur Implementierung von Best Practices für IaC finden Sie unter [AWS CloudFormation Als IaC-Tool verwenden](#).

Weitere Informationen zur Verwendung von Continuous Delivery mit CloudFormation finden Sie unter [Continuous Delivery with](#). CodePipeline

Fügen Sie keine Anmeldedaten in Ihre Vorlagen ein

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, externe Werte zu referenzieren, die in anderen Diensten wie dem AWS Systems Manager Parameter Store oder AWS Secrets Manager gespeichert und verwaltet werden. 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 [Verwenden von dynamischen Verweisen zum Angeben von Vorlagenwerten](#).

[AWS Secrets Manager](#) hilft Ihnen dabei, Anmeldeinformationen für Ihre Datenbanken und andere Dienste sicher zu verschlüsseln, zu speichern und 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](#).

Verwenden Sie AWS-spezifische Parametertypen

Wenn Ihre Vorlage Eingaben für bestehende AWS-spezifische Werte erfordert, wie z. B. bestehende Amazon Virtual Private Cloud Cloud-IDs oder einen Amazon EC2 EC2-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

Dropdownliste 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 Pseudoparametern in intrinsischen 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 Ziel AWS Partition, 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 verketteten mit: CloudFormation !Sub 'arn:\${AWS::Partition}:ssm:\${AWS::Region}:\${AWS::AccountId}:parameter/MySampleParameter

Ein weiteres Beispiel: Nehmen Sie an, Sie möchten Ressourcen oder Konfigurationen für mehrere Stacks freigeben. 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 referenzieren Sie den exportierten Wert der Subnetz-ID,

wenn Sie eine Amazon-EC2-Instance beschreiben. 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 Stapelnamen auch pro Konto und Region eindeutig sein müssen, erhöht die Verwendung dieses Pseudoparameters als Präfix die Möglichkeit, einen eindeutigen Exportnamen zu haben und fördert gleichzeitig einen wiederverwendbaren Ansatz für alle Stapel, aus denen Sie Werte exportieren. Alternativ können Sie ein Präfix Ihrer Wahl verwenden.

Verwenden von **AWS::CloudFormation::Init** zur Bereitstellung von Software-Anwendungen auf Amazon-EC2-Instances

Wenn Sie Stacks starten, können Sie mit dem Hilfsskript `cfn-init` und der Ressource `AWS::CloudFormation::Init` Softwareanwendungen auf Amazon EC2-Instances 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 Anleitung zur Verwendung von `cfn-init` und `AWS::CloudFormation::Init` finden Sie unter [Bereitstellen von Anwendungen auf Amazon 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 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-template](#) CLI-Befehl oder die [ValidateTemplate](#) API-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 `rules`, mit der Sie Regeln aus vorhandenen kompatiblen CloudFormation Vorlagen extrahieren können.

Weitere Informationen finden Sie im [CFN-Guard-Repository](#) unter. GitHub

Verwendung von YAML oder JSON für die Erstellung von Vorlagen

CloudFormation unterstützt sowohl das YAML- als auch das JSON-Format für Vorlagen. Jedes hat seine Vorteile, und die Wahl hängt von Ihren spezifischen Bedürfnissen ab:

Verwenden Sie YAML, wenn

- Sie legen Wert auf Lesbarkeit und Wartungsfreundlichkeit
- Sie möchten Kommentare einfügen, um Ihre Vorlage zu dokumentieren
- Sie arbeiten an komplexen Vorlagen mit verschachtelten Strukturen
- Sie möchten YAML-spezifische Features wie Anker und Aliase verwenden, um Wiederholungen zu vermeiden

Verwenden Sie JSON, wenn:

- Sie müssen sich mit Tools oder Systemen integrieren, die JSON bevorzugen
- Sie arbeiten mit programmgesteuerter Vorlagenerstellung oder-manipulation
- Sie benötigen eine strenge Datenvalidierung

YAML wird im Allgemeinen für die manuelle Erstellung von Vorlagen empfohlen, da es gut lesbar ist und Kommentare unterstützt. Sie ist besonders nützlich für komplexe Vorlagen, bei denen die auf Einrückungen basierende Struktur die Visualisierung von Ressourcenhierarchien erleichtert. JSON kann in automatisierten Workflows oder bei der Arbeit mit APIs dieser erwarteten JSON-Eingabe von Vorteil sein. Es ist auch von Vorteil, wenn Sie die strikte Einhaltung einer bestimmten Struktur sicherstellen müssen. Unabhängig davon, welches Format Sie wählen, sollten Sie sich auf die Erstellung gut strukturierter, dokumentierter und wartbarer Vorlagen konzentrieren. Wenn Sie YAML verwenden, nutzen Sie dessen Features wie Anker und Aliase, um Wiederholungen zu vermeiden und die Wartbarkeit zu verbessern.

Implementieren Sie eine umfassende Tagging-Strategie

Implementieren Sie eine konsistente Tagging-Strategie für alle Ressourcen, die mit Ihren CloudFormation Vorlagen erstellt wurden. Tags helfen bei der Organisation von Ressourcen, der Kostenzuweisung, der Zugriffskontrolle und der Automatisierung. Erwägen Sie die Aufnahme von Tags für Umgebung, Eigentümer, Kostenstelle, Anwendung und Zweck.

Verwenden Sie die Eigenschaft `Tags` der Ressource `AWS::CloudFormation::Stack`, um Tags auf alle unterstützten Ressourcen in einem Stack anzuwenden. Sie können auch die Eigenschaft `TagSpecifications` verwenden, die bei vielen Ressourcentypen verfügbar ist, um Tags bei der Erstellung der Ressource anzuwenden.

Weitere Informationen zum Taggen finden Sie unter [Ressourcentag](#).

Nutzen Sie Vorlagenmakros für erweiterte Transformationen

CloudFormation Makros ermöglichen Ihnen die benutzerdefinierte Verarbeitung von Vorlagen, von einfachen Aktionen wie find-and-replace Operationen bis hin zu komplexen Transformationen, die zusätzliche Ressourcen generieren. Verwenden Sie Makros, um die Funktionen von CloudFormation Vorlagen zu erweitern und wiederverwendbare Muster in Ihrer gesamten Organisation zu implementieren.

Dies AWS Serverless Application Model ist ein Beispiel für ein Makro, das die Entwicklung serverloser Anwendungen vereinfacht. Erwägen Sie die Erstellung benutzerdefinierter Makros für unternehmensspezifische Muster und Anforderungen.

Weitere Informationen zur Verwendung von Makros in Ihren Vorlagen finden Sie unter [Überblick über Makros](#). CloudFormation

Verwalten Sie 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 zur Drift finden Sie unter [Was ist Abweichung?](#).

Weitere Informationen zum Aktualisieren von Stapeln finden Sie unter [Einen CloudFormation Stack aktualisieren](#).

Erstellen von Änderungssätzen vor der Aktualisierung von Stacks

Mithilfe von Änderungssätzen können Sie sehen, wie sich vorgeschlagene Änderungen an einem Stack auf Ihre laufenden Ressourcen auswirken könnten, bevor Sie sie implementieren. 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 Sie Stapelrichtlinien zum Schutz von Ressourcen

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](#).

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 Sie bei Bedarf überprüfen, wer welchen CloudFormation Anruf in Ihrem Konto getätigt hat.

Weitere Informationen finden Sie unter [CloudFormation API-Aufrufe protokollieren mit AWS CloudTrail](#).

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 Amazon-EC2-Instances regelmäßig

Führen Sie auf all Ihren Amazon EC2 EC2-Windows-Instances und Amazon EC2 EC2-Linux-Instances, die mit erstellt wurden CloudFormation, 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.

Verwenden Sie die Drifterkennung regelmäßig

Verwenden Sie regelmäßig die CloudFormation Drift-Erkennungsfunktion, um Ressourcen zu identifizieren, die außerhalb der CloudFormation Verwaltung geändert wurden. Das Aufspüren und Beheben von Drifts trägt dazu bei, die Integrität Ihres Infrastructure as Code-Ansatzes aufrechtzuerhalten und stellt sicher, dass Ihre Vorlagen den Zustand Ihrer bereitgestellten Ressourcen genau wiedergeben.

Erwägen Sie die Einführung einer automatischen Drift-Erkennung als Teil Ihrer Betriebsabläufe. Sie können AWS Lambda Funktionen verwenden, die durch die EventBridge Amazon-Regeln ausgelöst werden, um regelmäßig nach Abweichungen zu suchen und Ihr Team zu benachrichtigen, wenn Unstimmigkeiten festgestellt werden.

Weitere Informationen zu Drift finden Sie unter [Erkennen von nicht verwalteten Konfigurationsänderungen an Stacks und Ressourcen mit der Drifterkennung](#).

Konfigurieren Sie Rollback-Auslöser für die automatische Wiederherstellung

Verwenden Sie Rollback-Trigger, um CloudWatch Amazon-Alarme festzulegen, die während der Stack-Erstellung und Aktualisierung überwacht werden CloudFormation sollen. Wenn einer der angegebenen Alarme in den ALARM Status wechselt, CloudFormation wird der gesamte Stack-Vorgang automatisch zurückgesetzt, wodurch sichergestellt wird, dass Ihre Infrastruktur in einem stabilen Zustand bleibt.

Konfigurieren Sie Rollback-Auslöser für kritische Metriken wie Anwendungsfehlerraten, Systemressourcenauslastung oder benutzerdefinierte Geschäftsmetriken, die den Zustand Ihrer Anwendung und Infrastruktur anzeigen.

Weitere Informationen über Rollback-Auslöser finden Sie unter [Rollback Ihres Stacks bei Alarmverletzung](#).

Implementieren Sie effektive Stack-Refactoring-Strategien

Im Zuge der Weiterentwicklung Ihrer Infrastruktur müssen Sie möglicherweise Ihre CloudFormation Stacks umgestalten, um die Wartbarkeit zu verbessern, die Komplexität zu reduzieren oder sich an sich ändernde Anforderungen anzupassen. Das Stack-Refactoring umfasst die Umstrukturierung Ihrer Vorlagen und Ressourcen unter Beibehaltung ihres externen Verhaltens und ihrer Funktionalität. Die Verwendung von Stack-Refactoring ist auf folgende Weise von Vorteil: CloudFormation

- Aufspaltung monolithischer Stapel: Aufteilung großer, komplexer Stapel in kleinere, besser verwaltbare Stapel, die nach Lebenszyklus oder Eigentümerschaft organisiert sind
- Konsolidierung verwandter Ressourcen: Zusammengehörige Ressourcen aus mehreren Stapeln in einem einzigen, zusammenhängenden Stapel zusammenfassen, um die Verwaltung zu vereinfachen
- Extrahieren wiederverwendbarer Komponenten: Verschieben gemeinsamer Muster in Module oder verschachtelte Stapel, um Wiederverwendung und Konsistenz zu fördern
- Verbesserung der Ressourcenorganisation: Umstrukturierung von Ressourcen innerhalb eines Stapels, um ihre Beziehungen und Abhängigkeiten besser widerzuspiegeln

Weitere Informationen zum Refactoring Ihrer CloudFormation Stacks finden Sie unter [Stapel-Refactoring](#)

Verwenden Sie CloudFormation Hooks für das Lebenszyklusmanagement

CloudFormation Hooks stellen Code bereit, der die Konfiguration Ihrer AWS Ressourcen vor der Bereitstellung proaktiv überprüft und komplexe Validierungsprüfungen durchführt. Hooks überprüfen, ob Ihre Ressourcen, Stacks und Änderungssätze mit den Anforderungen Ihres Unternehmens an Sicherheit, Betrieb und Kostenoptimierung übereinstimmen. Sie geben Warnungen aus, bevor eine Ressource bereitgestellt wird, oder schlagen den Vorgang fehl und stoppen ihn ganz, je nachdem, wie er konfiguriert wurde. Verstöße und Warnungen werden in Amazon protokolliert CloudWatch , um Einblick in nicht konforme Bereitstellungen zu geben.

Weitere Informationen zu diesen bewährten Verfahren für Hooks finden Sie unter, [AWS CloudFormation Hooks-Konzepte](#).

Weitere Informationen darüber, was Hooks für Ihre CloudFormation Ressourcen tun können, finden Sie unter [Was sind AWS CloudFormation Hooks?](#)

Verwenden Sie den IaC-Generator, um Vorlagen aus vorhandenen Ressourcen zu erstellen.

Der CloudFormation IaC (Infrastructure as Code) Generator hilft Ihnen dabei, CloudFormation Vorlagen aus Ihren vorhandenen AWS Ressourcen zu erstellen. Diese Funktion ist besonders nützlich, wenn Sie eine bestehende Infrastruktur replizieren, manuell erstellte Ressourcen dokumentieren oder zuvor nicht verwaltete Ressourcen verwalten müssen. CloudFormation Der IaC-Generator ist nützlich, um Ihre CloudFormation Vorlagen auf folgende Weise zu erstellen:

- Beschleunigte Vorlagenerstellung: Generieren Sie Vorlagen aus vorhandenen Ressourcen, anstatt sie von Grund auf neu zu schreiben
- Konsistente Infrastruktur: Stellen Sie sicher, dass neue Umgebungen mit bestehenden Umgebungen übereinstimmen, indem Sie generierte Vorlagen als Ausgangspunkt verwenden
- Migration zur Infrastruktur als Code: Schrittweise werden manuell erstellte Ressourcen verwaltet CloudFormation
- Documentation: Erstellen Sie eine Aufzeichnung Ihrer bestehenden Infrastruktur in Form einer Vorlage

Weitere Informationen über den IaC-Generator finden Sie unter [Vorlagen aus vorhandenen Ressourcen mit dem IaC-Generator generieren](#).

AWS Infrastructure Composer Für die visuelle Vorlagengestaltung verwenden

AWS Infrastructure Composer ist ein Tool für visuelles Design, mit dem Sie CloudFormation Vorlagen mithilfe einer drag-and-drop Benutzeroberfläche erstellen, visualisieren und ändern können. Es kann besonders nützlich sein, wenn es CloudFormation auf folgende Weise verwendet wird:

- Architekturplanung: Entwurf und Validierung von Infrastruktur-Architekturen vor der Implementierung
- Modernisierung von Vorlagen: Visualisieren Sie vorhandene Vorlagen, um ihre Struktur zu verstehen und Verbesserungsmöglichkeiten zu erkennen
- Schulung und Einarbeitung: Helfen Sie neuen Teammitgliedern durch visuelles Lernen, CloudFormation Konzepte und AWS Servicebeziehungen zu verstehen
- Kommunikation mit den Interessengruppen: Präsentieren Sie Infrastrukturentwürfe für nicht-technische Interessengruppen mit Hilfe klarer visueller Darstellungen
- Compliance-Überprüfungen: Verwenden Sie visuelle Diagramme, um Sicherheits- und Compliance-Prüfungen Ihrer Infrastrukturentwürfe zu erleichtern.
- Compliance-Überprüfungen: Verwenden Sie visuelle Diagramme, um Sicherheits- und Compliance-Prüfungen Ihrer Infrastrukturentwürfe zu erleichtern.

Weitere Informationen zu Infrastructure Composer finden Sie unter [Was ist AWS Infrastructure Composer?](#) .

Erwägen Sie AWS Cloud Development Kit (AWS CDK) die Verwendung für komplexe Infrastrukturen

Bei komplexen Infrastrukturanforderungen sollten Sie erwägen, das CDK zu verwenden, um Ihre Cloud-Ressourcen mithilfe vertrauter Programmiersprachen wie Python TypeScript, Java und .NET zu definieren. AWS CDK generiert CloudFormation Vorlagen aus Ihrem Code, sodass Sie alle Funktionen von nutzen und CloudFormation gleichzeitig die Abstraktionen und Programmierkonstrukte Ihrer bevorzugten Sprache verwenden können.

Das AWS CDK bietet Konstrukte auf hoher Ebene, die bewährte Verfahren zusammenfassen und die Definition gängiger Infrastrukturmuster vereinfachen. Dies kann die Menge an Code, die für die Definition Ihrer Infrastruktur benötigt wird, erheblich reduzieren und gleichzeitig die Einhaltung von Best Practices sicherstellen.

Weitere Informationen über die CDK finden Sie unter [AWS Cloud Development Kit \(AWS CDK\)](#).

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. Zudem benötigt jeder, der CloudFormation-Stacks verwaltet, Berechtigungen für Ressourcen innerhalb dieser Stacks. Wenn Benutzer beispielsweise Amazon EC2 EC2-Instances starten, aktualisieren oder beenden möchten, müssen sie über die Berechtigung verfügen, die entsprechenden Amazon EC2 EC2-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 Service-Rolle, 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 [CloudFormation Servicerolle](#).

Wenden Sie das Prinzip des geringsten Privilegs an

Wenden Sie bei der Konfiguration von IAM-Rollen für CloudFormation Servicerollen oder für Ressourcen, die mit Ihren Vorlagen erstellt wurden, immer das Prinzip der geringsten Rechte an. Gewähren Sie nur die Berechtigungen, die für die vorgesehene Funktionalität erforderlich sind, und vermeiden Sie nach Möglichkeit die Verwendung von Platzhalterberechtigungen.

Verwenden Sie IAM Access Analyzer, um die Ihren CloudFormation Servicerollen gewährten Berechtigungen zu überprüfen und ungenutzte Berechtigungen zu identifizieren, die entfernt werden können. Überprüfen und aktualisieren Sie die IAM-Richtlinien regelmäßig, um sicherzustellen, dass sie weiterhin mit Ihren Sicherheitsanforderungen übereinstimmen.

Sichern Sie sensible Parameter

Verwenden Sie für vertrauliche Informationen wie Passwörter, API-Schlüssel und andere geheime Daten den AWS Systems Manager Parameter Store oder betten Sie sie AWS Secrets Manager nicht direkt in Ihre Vorlagen ein. Verwenden Sie dynamische Referenzen in Ihren Vorlagen, um diese Werte bei Stapeloperationen sicher abzurufen.

Wenn Sie Parameter in Ihren Vorlagen verwenden, setzen Sie die Eigenschaft `NoEcho` für sensible Parameter auf `true`, um zu verhindern, dass ihre Werte in der Konsole, in API-Antworten oder in der CLI-Ausgabe angezeigt werden. Beachten Sie, dass `NoEcho` nicht verhindert, dass der Wert protokolliert wird, wenn er an andere Dienste oder Ressourcen übergeben wird, die den Wert möglicherweise protokollieren.

Weitere Informationen zur Verwendung von AWS Systems Manager Parameter Store CloudFormation finden [Sie unter Abrufen eines Klartextwerts aus dem AWS Systems Manager Parameterspeicher](#).

Weitere Informationen zur Verwendung der `NoEcho` Eigenschaft finden Sie unter [Syntax der CloudFormation Vorlagenparameter](#).

Weitere Informationen zur Verwendung von AWS Secrets Manager mit CloudFormation finden Sie unter [Create AWS Secrets Manager secrets in AWS CloudFormation](#).

Implementieren Sie Richtlinien als Code mit AWS CloudFormation Guard

AWS CloudFormation Guard (`cfn-guard`) ist ein policy-as-code Open-Source-Tool, mit dem Sie Regeln für Ihre CloudFormation Vorlagen definieren und durchsetzen können. Verwenden Sie `cfn-guard`, um sicherzustellen, dass Ihre Vorlagen den Unternehmensrichtlinien, bewährten Sicherheitsverfahren und Governance-Anforderungen entsprechen.

Integrieren Sie es `cfn-guard` in Ihre CI/CD Pipelines, um Vorlagen vor der Bereitstellung automatisch anhand Ihrer Richtlinienregeln zu validieren. Auf diese Weise können Sie verhindern, dass nicht konforme Ressourcen in Ihrer Umgebung bereitgestellt werden, und den Entwicklern frühzeitig eine Rückmeldung über Richtlinienverstöße geben.

Weitere Informationen zu Guard finden Sie unter [Was ist AWS CloudFormation Guard?](#)

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 Abschnitt [Resources](#), in dem mindestens eine Ressource deklariert werden muss.

Sie können Vorlagen mit den folgenden Methoden erstellen:

- **AWS Infrastructure Composer**- Eine visuelle Schnittstelle für die Gestaltung von Vorlagen.
- **Text Editor**- 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 arbeitet mit einer breiten Palette 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 validieren](#)
- [Erste Schritte mit Vorlagen](#)
- [Mustervorlagen](#)
- [CloudFormation Vorlagenformat](#)
- [CloudFormation Vorlagenabschnitte](#)
- [Visuelle Erstellung von Vorlagen mit Infrastructure Composer](#)
- [AWS CloudFormation Sprachserver](#)
- [Erzeugen Sie Vorlagen aus vorhandenen Ressourcen mit dem IaC-Generator](#)
- [Abrufen von in anderen Diensten gespeicherten Werten mit Hilfe von dynamischen Referenzen](#)
- [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. Wenn Sie einen Stapel erstellen oder aktualisieren, können Sie die S3-URL der Vorlage angeben, anstatt sie direkt hochzuladen.

Wenn Sie Vorlagen direkt über das AWS-Managementkonsole 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 einen Stack aus dem Repository-Quellcode mit Git-Synchronisierung](#).

Vorlagen validieren

Überprüfung der Syntax

Sie können die JSON- oder YAML-Syntax Ihrer Vorlage überprüfen, indem Sie den CLI-Befehl [validate-template](#) verwenden oder Ihre Vorlage in 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 Validierungswerkzeuge

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

- [CloudFormation Linter \(cfn-lint\)](#) — Validiert Vorlagen anhand der Schemas des [CloudFormation Ressourcenanbieters](#). Umfasst die Überprüfung gültiger Werte für Ressourceneigenschaften und bewährte Verfahren.
- [CloudFormation Rain \(rain fmt\)](#) — Formatieren Sie Ihre CloudFormation Vorlagen nach einem konsistenten Standard oder formatieren Sie eine Vorlage von JSON nach YAML (oder YAML nach JSON) neu. Bei der Verwendung von YAML werden Kommentare beibehalten und die Verwendung von intrinsischen Funktionen wird nach Möglichkeit auf die kurze Syntax umgestellt.

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. Validieren Sie die Vorlage- Validieren Sie die Vorlage mit den im Abschnitt [Vorlagen validieren](#) beschriebenen Methoden.
5. Einen Stack erstellen- Erstellen Sie einen Stack unter Verwendung der validierten Vorlage.

Planen Sie, die CloudFormation Vorlagenreferenz zu verwenden

Wenn Sie Ihre Vorlagen schreiben, finden Sie die Dokumentation für die detaillierte Syntax für die verschiedenen Ressourcentypen in der [AWS Ressourcen- und Eigenschaftstypen-Referenz](#).

Häufig benötigen Ihre Stack-Vorlagen intrinsische Funktionen, um Eigenschaftswerte zuzuweisen, die erst zur Laufzeit verfügbar sind, sowie spezielle Attribute, um das Verhalten von Ressourcen

zu steuern. Wenn Sie Ihre Vorlage schreiben, können Sie sich an den folgenden Ressourcen orientieren:

- [Referenz auf eine intrinsische Funktion](#)- Einige häufig verwendete intrinsische Funktionen sind:
 - 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 Zeichenkette.
- [Ressourcenattribut-Referenz](#)- Einige häufig verwendete spezielle Attribute sind:
 - DependsOn- Verwenden Sie dieses Attribut, um anzugeben, dass eine Ressource nach einer 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 [CloudFormation Beispielvorlagen](#) auf der GitHub Website.

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

Jede Vorlage in diesem Repository besteht die [CloudFormation Linter-Prüfungen](#) (cfn-lint) sowie 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 dem gleichen Zweck, bieten aber deutliche Vorteile in Bezug auf Lesbarkeit und Komplexität.

- JSON- JSON ist ein leichtgewichtiges Datenaustauschformat, das von Maschinen leicht geparkt und generiert werden kann. Allerdings kann es für Menschen mühsam werden, sie zu lesen

und zu schreiben, insbesondere bei komplexen Konfigurationen. In JSON wird die Vorlage durch verschachtelte geschweifte Klammern `{}` und Klammern `[]` strukturiert, um Ressourcen, Parameter und andere Komponenten zu definieren. Ihre Syntax erfordert die explizite Deklaration jedes Elements, was die Vorlage zwar langatmig machen kann, aber die strikte Einhaltung eines strukturierten Formats gewährleistet.

- **YAML**- YAML ist besser lesbar und weniger ausführlich als JSON. Es verwendet Einrückungen anstelle von geschweiften Klammern und Klammern, um Verschachtelungen zu kennzeichnen, was die Hierarchie der Ressourcen und Parameter leichter sichtbar macht. YAML wird oft wegen seiner Klarheit und Benutzerfreundlichkeit bevorzugt, insbesondere wenn es um komplexere Vorlagen geht. Da sich YAML jedoch auf die Einrückung verlässt, kann es zu Fehlern kommen, wenn die Abstände nicht konsistent sind, was sorgfältige Aufmerksamkeit erfordert, um die Genauigkeit zu wahren.

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 angegeben werden, bei anderen spielt die Reihenfolge keine Rolle. Bei der Erstellung Ihrer Vorlage kann es jedoch hilfreich sein, die in den folgenden Beispielen gezeigte logische Reihenfolge zu verwenden, da sich die Werte in einem Abschnitt auf Werte aus einem früheren Abschnitt beziehen können.

Verwenden Sie beim Erstellen von Vorlagen keine doppelten Hauptabschnitte, wie beispielsweise den Abschnitt `Resources`. Sie akzeptiert CloudFormation zwar die Vorlage, verhält sich aber bei der Verarbeitung der Vorlage undefiniert und stellt möglicherweise Ressourcen falsch bereit oder gibt unerklärliche Fehler zurück.

JSON

Das folgende Beispiel zeigt die Struktur einer JSON-formatierten Vorlage 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 YAML-formatierten Vorlage 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 JSON-formatierten Vorlagen werden Kommentare nicht unterstützt. JSON enthält von Haus aus keine Syntax für Kommentare, also Sie können keine Kommentare direkt in die JSON-Struktur einfügen. Wenn Sie jedoch Erklärungen oder Dokumentationen einfügen müssen, können Sie Metadaten hinzufügen. Weitere Informationen finden Sie unter [Metadata-Attribut](#).

In YAML-formatierte Vorlagen können Sie Inline-Kommentare einfügen, indem Sie das Symbol # verwenden.

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`
- Aliase
- 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 deren Eigenschaften und Werte anhand 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. Der Grund dafür ist, dass JSON Backslashes als Escape-Zeichen interpretiert und Sie sie als Escape-Zeichen verwenden müssen, damit sie im regulären Ausdruck als buchstäbliche 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 Eigenschaft AllowedPattern einen regulären Ausdruck an, der auf vier aufeinanderfolgende Ziffern (\d{4}) passt. Da der reguläre Ausdruck jedoch in einer JSON-Vorlage definiert ist, muss das Backslash-Zeichen mit einem zusätzlichen Backslash (\\d) escaped werden.

```
{
  "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 wird 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. Es ermöglicht Benutzern, bei der Erstellung oder Aktualisierung eines Stapels Werte zur Laufzeit zu übergeben. Auf diese Parameter kann in den Abschnitten `Resources` und `Outputs` verwiesen werden, so dass eine Anpassung möglich ist, ohne die Vorlage selbst zu ändern. Sie können zum Beispiel Parameter verwenden, um Instancetypen oder Umgebungseinstellungen festzulegen, die sich zwischen den Einsätzen unterscheiden.

Der ebenfalls optionale Abschnitt `Outputs` definiert die Werte, die bei der Anzeige der Eigenschaften eines Stacks zurückgegeben 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 Details zu den von der Vorlage erstellten Ressourcen abrufen und verwenden.

Weitere optionale Abschnitte sind `Mappings`, die wie Nachschlagetabellen funktionieren, um bedingte Werte zu verwalten. Mit Mappings definieren Sie Schlüssel-Wert-Paare und verwenden diese mit der intrinsischen Funktion `Fn::FindInMap` in den Abschnitten `Resources` und `Outputs`. Dies ist nützlich für Szenarien, in denen Sie Konfigurationen auf der Grundlage von Bedingungen wie AWS-Region der Umgebung anpassen müssen.

Die Abschnitte `Metadaten` und `Regeln` werden zwar weniger häufig verwendet, bieten jedoch zusätzliche Funktionen. `Metadata` kann zusätzliche Informationen über die Vorlage enthalten, während `Rules` einen Parameter oder eine Kombination von Parametern bei der Erstellung oder Aktualisierung von Stapeln validiert und sicherstellt, dass sie bestimmte Kriterien erfüllen. Der Abschnitt `Bedingungen` erhöht die Flexibilität weiter, indem er steuert, ob bestimmte Ressourcen

erstellt werden oder Eigenschaften einen Wert zugewiesen bekommen, der auf Bedingungen wie dem Umgebungstyp basiert.

Der Abschnitt Transform schließlich wird verwendet, um Makros während der Verarbeitung 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::IncludeTransformation` 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 Abschnitt `Resources` 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 sein (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 ein Type-Attribut haben, das die Art der AWS -Ressource definiert. Das Attribut Type hat das Format `AWS::ServiceName::ResourceType`. Das Attribut Type für einen Amazon S3-Bucket lautet zum Beispiel `AWS::S3::Bucket`.

Eine vollständige Liste der unterstützten Ressourcentypen finden Sie in der Referenz [AWS Ressourcen- und Eigenschaftstypen](#).

ResourceProperties

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

Einzelheiten zu den Eigenschaften, die für jeden Ressourcentyp unterstützt werden, finden Sie in den Themen unter [AWS Ressourcen- und Eigenschaftstypen Referenz](#).

Eigenschaftswerte können literale Strings, Listen von Strings, Boolesche Werte, dynamische Referenzen, Parameterreferenzen, Pseudoreferenzen oder der von einer Funktion zurückgegebene Wert sein. Die folgenden Beispiele zeigen Ihnen, wie Sie verschiedene Arten von Eigenschaftswerten deklarieren können:

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

Neben einer logischen ID haben bestimmte Ressourcen auch eine physische ID. Sie ist der tatsächliche Name, der dieser Ressource zugewiesen ist, zum Beispiel die ID einer EC2-Instance oder der Name eines S3-Buckets. Verwenden Sie die physischen IDs, um Ressourcen außerhalb von CloudFormation Vorlagen zu identifizieren, jedoch erst, nachdem die Ressourcen erstellt wurden. Angenommen, Sie geben einer EC2-Instance-Ressource eine logische ID von `MyEC2Instance`. Beim CloudFormation Erstellen der Instanz CloudFormation wird automatisch eine physische ID (z. B. `i-1234567890abcdef0`) generiert und der Instanz zugewiesen. Diese physische ID können Sie verwenden, um die Instance zu referenzieren und ihre Eigenschaften (beispielsweise den DNS-Namen) über die Amazon-EC2-Konsole abzurufen.

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 `habenMyStack`, CloudFormation könnten Sie den Bucket mit dem folgenden physischen Namen benennen `MyStack-MyBucket-abcdefghijkl1`.

Für Ressourcen, die benutzerdefinierte Namen unterstützen, können Sie Ihre eigenen physischen Namen vergeben, damit Sie die Ressourcen schnell identifizieren 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 die Eigenschaften einer Ressource auf der Grundlage des Namens oder der Eigenschaft einer anderen Ressource festlegen. Sie könnten beispielsweise eine EC2-Instance 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 er zurückgibt, hängt von der Art der Ressource ab. Für die meisten Ressourcen gibt es den physischen Namen der Ressource zurück. Für einige Ressourcentypen wird jedoch möglicherweise ein anderer Wert zurückgegeben, z. B. eine IP-Adresse für eine `AWS::EC2::EIP`-Ressource oder ein Amazon Resource Name (ARN) für ein Amazon SNS-Thema.

Die folgenden Beispiele zeigen, wie Sie die Funktion Ref in Eigenschaften verwenden können. In jedem dieser Beispiele gibt die Funktion Ref den tatsächlichen Namen der an anderer Stelle in der Vorlage deklarierten Ressource `LogicalResourceName` zurück. Das `!Ref`-Syntaxbeispiel im YAML-Beispiel ist nur eine kürzere Schreibweise für die Ref-Funktion.

JSON

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

YAML

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

Ausführlichere Informationen über die Funktion Ref finden Sie unter [Ref](#).

Die Funktion **Fn::GetAtt**

Die Funktion Ref ist hilfreich, wenn der Parameter oder der für eine Ressource zurückgegebene Wert genau das ist, was Sie wollen. Es kann jedoch sein, dass Sie andere Attribute einer Ressource benötigen. 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 Funktion `Fn::GetAtt`.

Die folgenden Beispiele zeigen, wie Sie die Funktion `GetAtt` in Eigenschaften verwenden können. Die Funktion `Fn::GetAtt` benötigt zwei Parameter, den logischen Namen der Ressource und den

Namen des Attributs, das abgerufen werden soll. Das `!GetAtt`-Syntaxbeispiel im YAML-Beispiel ist nur eine kürzere Schreibweise für die `GetAtt`-Funktion.

JSON

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

YAML

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

Ausführlichere Informationen über die Funktion `GetAtt` 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, und auf vorhandene AWS Ressourcen.

Themen

- [Deklarieren einer einzelnen Ressource mit einem benutzerdefinierten Namen](#)
- [Verweis auf andere Ressourcen mit der Funktion Ref](#)
- [Referenzieren von Ressourcenattributen mit der Funktion Fn::GetAtt](#)

Deklarieren einer einzelnen Ressource mit einem benutzerdefinierten Namen

Die folgenden Beispiele deklarieren eine einzelne Ressource vom Typ `AWS::S3::Bucket` mit dem logischen Namen `MyBucket`. Die Eigenschaft `BucketName` ist auf `amzn-s3-demo-bucket` eingestellt, die Sie durch den gewünschten Namen für Ihren S3-Bucket ersetzen sollten.

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 Amazon EC2 EC2-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
```

Verweis auf andere Ressourcen mit der Funktion **Ref** .

Die folgenden Beispiele zeigen eine Ressourcendeklaration, die eine EC2-Instance und eine Sicherheitsgruppe definiert. Die Ec2Instance-Ressource referenziert die InstanceSecurityGroup-Ressource als Teil ihrer SecurityGroupIds-Eigenschaft unter Verwendung der Ref-Funktion. Sie enthält auch eine bestehende 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 mit 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 und verwendet die Fn::GetAtt-Funktion, um das DomainName-Attribut des Buckets zu erhalten. Sie werden feststellen, dass die Funktion Fn::GetAtt 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"
            }
          }
        }
      }
    }
  }
}
```


verwenden. Weitere Informationen finden Sie unter [Geben Sie vorhandene Ressourcen zur Laufzeit mit von CloudFormation -bereitgestellten Parametertypen an](#).

Parameter sind eine beliebte Methode, um Eigenschaftswerte von Stack-Ressourcen anzugeben. Es kann jedoch Einstellungen geben, die von der Region abhängen oder die für den Benutzer aufgrund anderer Bedingungen oder Abhängigkeiten etwas kompliziert zu verstehen sind. In diesen Fällen möchten Sie vielleicht eine Logik in die Vorlage selbst einbauen, damit die Benutzer einfachere Werte (oder gar keine) angeben können, um die gewünschten Ergebnisse zu erhalten, z. B. durch Verwendung einer Zuordnung. Weitere Informationen finden Sie unter [CloudFormation MappingsVorlagensyntax](#).

Syntax

Sie deklarieren Parameter im Abschnitt Parameters 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 erscheinen auf der Seite Parameter angeben wenn Sie die Vorlage im Assistenten Stapel erstellen verwenden.

Note

Standardmäßig listet die CloudFormation Konsole Eingabeparameter alphabetisch nach ihrer logischen ID auf. Um diese Standardreihenfolge außer Kraft zu setzen und verwandte Parameter zu gruppieren, können Sie den Metadatenschlüssel `AWS::CloudFormation::Interface` 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 die Benutzer einen Wert für diesen Parameter angeben. Die Aufforderung an den Benutzer, einen Wert einzugeben, stellt jedoch nicht sicher, 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 die Benutzer bei der Stack-Erstellung einen Wert für den Schlüsselnamen angeben. Ist dies nicht der Fall, 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 `setzenttrue`, 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.

Um festzustellen, welche Ressourceneigenschaften den primären Bezeichner eines Ressourcentyps umfassen, lesen Sie die Dokumentation der Ressourcenreferenz für diese Ressource in der [AWS Ressourcen- und Eigenschaftstypenreferenz](#). Im

Abschnitt `Return values` (Rückgabewerte) 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 Constraints zu deklarieren: `MinLength`, `MaxLength`, `Default AllowedValues` und `AllowedPattern`.

Zum Beispiel könnten Benutzer `"MyUserName"` angeben.

Number

Eine Ganzzahl oder ein Fließkommawert. 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 Beschränkungen 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 Amazon EC2 EC2-Schlüsselpaarnamen und IDs VPC. Weitere Informationen finden Sie unter [Vorhandene Ressourcen zur Laufzeit spezifizieren](#).

Systems Manager Parameter-Typen

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 ruft den neuesten Wert aus dem Parameterspeicher ab, um ihn für den Stack zu verwenden. Weitere Informationen finden Sie unter [Vorhandene Ressourcen zur Laufzeit spezifizieren](#).

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) gegeben werden, der alphanumerisch und unter allen logischen Namen innerhalb der Vorlage eindeutig sein muss.
- Jedem Parameter muss ein Parametertyp zugewiesen werden, der von unterstützt wird CloudFormation. 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 String-Parameter](#)
- [Parameter Passwort](#)
- [Parameter referenzieren](#)
- [Komma-getrennte Listenparameter](#)
- [Zurückgeben eines Werts aus einem Parameter für Komma-getrennte Listen](#)

Einfacher String-Parameter

Das folgende Beispiel deklariert einen Parameter mit dem Namen `InstanceTypeParameter` vom Typ `String`. Mit diesem Parameter können Sie den Amazon EC2-Instancetyp für den Stack angeben. Wenn bei der Erstellung oder Aktualisierung des Stacks kein Wert angegeben wird, wird der Standardwert von CloudFormation verwendet `t2.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
```

Parameter Passwort

Das folgende Beispiel deklariert einen Parameter namens DBPwd vom Typ String ohne Standardwert. Die Eigenschaft NoEcho wird auf true gesetzt, um zu verhindern, dass der Parameterwert in Stapelbeschreibungen 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 Eigenschaft AllowedPattern .

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 referenziert die Eigenschaft InstanceType der EC2-Instance-Ressource den Parameterwert InstanceTypeParameter:

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
```

Komma-getrennte Listenparameter

Der Parametertyp `CommaDelimitedList` kann nützlich sein, wenn Sie mehrere Werte für eine einzelne Eigenschaft angeben müssen. Das folgende Beispiel deklariert einen Parameter namens `DbSubnetIpBlocks` mit einem Standardwert von drei durch Kommas getrennten CIDR-Blöcken.

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 kommasetrennten Liste eines Parameters zu verweisen, verwenden Sie die intrinsische Funktion `Fn::Select` im Abschnitt `Resources` Ihrer Vorlage. 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::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" }
            ]
          }
        }
      ]
    }
  }
}
}

```



```

    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. Sie müssen zum Beispiel auf sichere Strings verweisen, die im Systems Manager Parameter Store gespeichert sind. Weitere Informationen finden Sie unter [Abrufen von in anderen Diensten gespeicherten Werten mit Hilfe von dynamischen Referenzen](#).

Sie können auch Pseudoparameter innerhalb einer Ref- oder Sub-Funktion verwenden, um Werte dynamisch zu füllen. Weitere Informationen finden Sie unter [AWS Werte mithilfe von Pseudo-Parametern abrufen](#).

CloudFormation OutputsVorlagensyntax

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

- Erfassen Sie wichtige Details über Ihre 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-stacks](#) CLI-Befehls anzeigen.

- Stackübergreifende Referenzen- Sie können Ausgabewerte in andere Stack importieren, um [Referenzen zwischen Stackn herzustellen](#). Dies ist hilfreich, wenn Sie Ressourcen oder Konfigurationen über mehrere Stacks hinweg 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 IN_PROGRESS-[Status](#) befindet. 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)

Der Wert der Eigenschaft, die durch den Befehl [describe-stacks](#) zurückgegeben wird. Der Wert einer Ausgabe kann Literale, Parameterreferenzen, Pseudoparameter, einen Mapping-Wert oder intrinsische 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](#)
- [Exportname mit Fn::Subanpassen](#)
- [Exportname mit Fn::Joinanpassen](#)
- [Gibt eine URL zurück, die mit Fn::Joinerstellt wurde.](#)

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 Ausgabe mit dem Namen InstanceID gibt die ID der EC2-Instance mit der logischen ID EC2Instance zurück.

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

Exportname mit **Fn::Sub** anpassen

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](#).

Exportname mit **Fn::Join** anpassen

Sie können die Funktion **Fn::Join** auch verwenden, um Werte auf der Grundlage von Parametern, Ressourcenattributen und anderen Zeichenfolgen zu konstruieren.

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

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 mit **Fn::Join** erstellt wurde.

Im folgenden Beispiel für eine Vorlage, die eine WordPress Site erstellt, `InstallURL` ist die Zeichenfolge, die von einem `Fn::Join` Funktionsaufruf zurückgegeben wird, der verketteth`http://`, den DNS-Namen der Ressource und `ElasticLoadBalancer/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 Abschnitt `Mappings` hilft Ihnen bei der Erstellung von Schlüssel-Wert-Paaren, mit denen Sie Werte basierend auf bestimmten Bedingungen oder Abhängigkeiten angeben können.

Ein üblicher Anwendungsfall für den Abschnitt `Mappings` ist die Festlegung von Werten auf der Grundlage von AWS-Region, wo der Stack eingesetzt wird. Dies kann durch die Verwendung des Pseudoparameters `AWS::Region` erreicht werden. Der `AWS::Region` Pseudo-Parameter ist ein Wert, der in die Region CloudFormation aufgelöst wird, in der der Stack erstellt wurde. Pseudo-Parameter werden aufgelöst, CloudFormation wenn Sie den Stack erstellen.

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

Syntax

Der Abschnitt `Mappings` 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 die Zuordnung.
- Innerhalb der Zuordnung ist jede Karte ein Schlüssel, gefolgt von einer anderen Zuordnung.
- 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 `List` sein.

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 allein aktualisieren. Sie müssen Änderungen einbeziehen, die Ressourcen hinzufügen, ändern oder löschen.

Beispiele

Themen

- [Grundlegende Zuweisungen](#)
- [Zuweisung mit mehreren Werten](#)
- [Zurückgeben eines Werts aus einer Zuweisung](#)
- [Eingabeparameter und `Fn::FindInMap`](#)

Grundlegende Zuweisungen

Das folgende Beispiel zeigt einen `Mappings`-Abschnitt mit einer Map `RegionToInstanceType`, 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 Instancetyp aus der T-Familie, der in der Region verfügbar ist, die durch den Schlüssel repräsentiert wird. Die Name-Wert-Paare haben einen Namen (`InstanceType` in diesem Beispiel) und einen Wert.

JSON

```
"Mappings" : {
  "RegionToInstanceType" : {
    "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:  
  RegionToInstanceType:  
    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 hat Regionsschlüssel, die zwei Gruppen von Werten zugeordnet sind: eine mit dem Namen MyAMI1 und die andere mit MyAMI2.

Note

Die in diesen Beispielen IDs gezeigten AMI sind Platzhalter zu Demonstrationszwecken. Wann immer möglich, sollten Sie dynamische Verweise auf AWS Systems Manager - Parameter als Alternative zum Mappings-Abschnitt verwenden. Um zu vermeiden, dass alle Ihre Vorlagen bei jeder Änderung des AMI, das Sie verwenden möchten, mit einer neuen ID aktualisiert werden, verwenden Sie einen AWS Systems Manager Parameter, um die neueste AMI-ID abzurufen, wenn der Stack erstellt oder aktualisiert wird. Die neuesten Versionen von häufig verwendeten AMIs sind auch als öffentliche Parameter in Systems Manager verfügbar. Weitere Informationen finden Sie unter [Abrufen von in anderen Diensten gespeicherten Werten unter Verwendung dynamischer Referenzen](#).

JSON

```
"Mappings" : {
```

```

"RegionToAMI" : {
  "us-east-1"      : { "MyAMI1" : "ami-12345678901234567", "MyAMI2" :
"ami-23456789012345678" },
  "us-west-1"      : { "MyAMI1" : "ami-34567890123456789", "MyAMI2" :
"ami-45678901234567890" },
  "eu-west-1"      : { "MyAMI1" : "ami-56789012345678901", "MyAMI2" :
"ami-67890123456789012" },
  "ap-southeast-1" : { "MyAMI1" : "ami-78901234567890123", "MyAMI2" :
"ami-89012345678901234" },
  "ap-northeast-1" : { "MyAMI1" : "ami-90123456789012345", "MyAMI2" :
"ami-01234567890123456" }
}
}

```

YAML

```

Mappings:
  RegionToAMI:
    us-east-1:
      MyAMI1: ami-12345678901234567
      MyAMI2: ami-23456789012345678
    us-west-1:
      MyAMI1: ami-34567890123456789
      MyAMI2: ami-45678901234567890
    eu-west-1:
      MyAMI1: ami-56789012345678901
      MyAMI2: ami-67890123456789012
    ap-southeast-1:
      MyAMI1: ami-78901234567890123
      MyAMI2: ami-89012345678901234
    ap-northeast-1:
      MyAMI1: ami-90123456789012345
      MyAMI2: ami-01234567890123456

```

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 Amazon EC2-Ressource, deren Eigenschaft `InstanceType` durch die Funktion `FindInMap` festgelegt wird. Die `FindInMap` Funktion gibt Schlüssel als den AWS-Region Ort an, an dem der Stack erstellt wird (unter Verwendung des `AWS::Region` Pseudo-Parameters) und `InstanceType` als Namen des Werts, dem die Zuordnung erfolgen soll, an. Der `ImageId` verwendet einen Systems Manager-

Parameter, um dynamisch das neueste Amazon Linux 2 AMI abzurufen. Weitere Informationen zu Pseudoparametern finden Sie unter [AWS Werte mithilfe von Pseudo-Parametern abrufen](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
    "RegionToInstanceType" : {
      "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" }
    }
  },
  "Resources" : {
    "myEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
        "InstanceType" : { "Fn::FindInMap" : [ "RegionToInstanceType", { "Ref" :
"AWS::Region" }, "InstanceType" ]}
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Mappings:
  RegionToInstanceType:
    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
Resources:
  myEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
      InstanceType: !FindInMap [RegionToInstanceType, !Ref 'AWS::Region', InstanceType]

```

Eingabeparameter und **Fn::FindInMap**

Die folgende Beispielvorlage zeigt, wie Sie eine EC2-Instance mit mehreren Mappings erstellen. Die Vorlage verwendet verschachtelte Zuordnungen, um automatisch den geeigneten Instanztyp und die entsprechende Sicherheitsgruppe basierend auf dem Ziel- AWS-Region und Umgebungstyp (oder) auszuwählen. Dev Prod Außerdem verwendet es einen Systems Manager-Parameter, um dynamisch das neueste Amazon Linux 2 AMI abzurufen.

JSON

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "EnvironmentType" : {
      "Description" : "The environment type (Dev or Prod)",
      "Type" : "String",
      "Default" : "Dev",
      "AllowedValues" : [ "Dev", "Prod" ]
    }
  },
  "Mappings" : {
    "RegionAndEnvironmentToInstanceType" : {
      "us-east-1"      : { "Dev" : "t3.micro", "Prod" : "c5.large" },
      "us-west-1"      : { "Dev" : "t2.micro", "Prod" : "m5.large" }
    },
    "RegionAndEnvironmentToSecurityGroup" : {
      "us-east-1"      : { "Dev" : "sg-12345678", "Prod" : "sg-abcdef01" },
      "us-west-1"      : { "Dev" : "sg-ghijkl23", "Prod" : "sg-45678abc" }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {

```

```

    "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}",
    "InstanceType" : { "Fn::FindInMap": [ "RegionAndEnvironmentToInstanceType",
{ "Ref": "AWS::Region" }, { "Ref": "EnvironmentType" } ]},
    "SecurityGroupIds" : [{ "Fn::FindInMap" :
[ "RegionAndEnvironmentToSecurityGroup", { "Ref" : "AWS::Region" }, { "Ref" :
"EnvironmentType" } ]}]
  }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvironmentType:
    Description: The environment type (Dev or Prod)
    Type: String
    Default: Dev
    AllowedValues:
      - Dev
      - Prod
Mappings:
  RegionAndEnvironmentToInstanceType:
    us-east-1:
      Dev: t3.micro
      Prod: c5.large
    us-west-1:
      Dev: t2.micro
      Prod: m5.large
  RegionAndEnvironmentToSecurityGroup:
    us-east-1:
      Dev: sg-12345678
      Prod: sg-abcdef01
    us-west-1:
      Dev: sg-ghijkl23
      Prod: sg-45678abc
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:

```

```
ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
InstanceType: !FindInMap [RegionAndEnvironmentToInstanceType, !Ref
'AWS::Region', !Ref EnvironmentType]
SecurityGroupIds:
  - !FindInMap [RegionAndEnvironmentToSecurityGroup, !Ref 'AWS::Region', !Ref
EnvironmentType]
```

Zugehörige Ressourcen

Diese verwandten Themen können hilfreich sein, wenn Sie Vorlagen entwickeln, die die Funktion `Fn::FindInMap` verwenden.

- [Fn::FindInMap](#)
- [Fn::FindInMap-Verbesserungen](#)
- [Fn::Sub](#)

CloudFormation MetadataVorlagensyntax

Metadata speichert zusätzliche Informationen in Form 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-Werte-Paare. Sie können zum Beispiel zusätzliche Informationen bereitstellen, die sich nicht auf die Erstellung von Ressourcen auswirken, aber zusätzlichen Kontext über die Infrastruktur, das Team oder die Besonderheiten der Bereitstellung 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`

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"
```

Für die Syntax für `AWS::CloudFormation::Interfaces` siehe [Organisieren 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 ihren logischen IDs auf, wenn Sie Stapel 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 können beispielsweise alle EC2-bezogenen Parameter in einer Gruppe zusammenfassen und alle VPC-Parameter in einer anderen Gruppe.

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 Anzeigenamen 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. Die Parameterbezeichnung muss die logische ID eines gültigen Parameters sein, der im Abschnitt Parameters der Vorlage deklariert wurde.

Required: No

default

Das Standardlabel, das 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

Parameters

Network Configuration

Which VPC should this be deployed to? Search by ID, or Name tag value
VpcId of your existing Virtual Private Cloud (VPC)

SubnetId Search by ID, or Name tag value
The list of SubnetIds in your Virtual Private Cloud (VPC)

SecurityGroupID Search by ID, name or Name tag value
VpcId of your existing Virtual Private Cloud (VPC)

Amazon EC2 Configuration

InstanceType m1.small Web

KeyName Search
Name of an existing EC2 KeyPair to enable SSH access to the instance

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 die Standard-Parameterbeschränkungen nicht ausreichen. Wenn zum Beispiel SSL aktiviert ist, müssen sowohl ein Zertifikat als auch ein Domainname angegeben werden. Eine Regel kann sicherstellen, dass diese Abhängigkeiten erfüllt werden.

Syntax

Der Abschnitt `Rules` verwendet die folgende Syntax:

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 diese durch Kommas voneinander 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 in Kraft tritt. 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 zu `true` ausgewertet werden muss.

AssertDescription

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

Regelspezifische intrinsische Funktionen

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

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 den Bedingungen oder Behauptungen einer Regel verwendet. Die Eigenschaft `condition` bestimmt, ob die Assertionen CloudFormation angewendet werden. Wenn die Bedingung zu `true` ausgewertet wird, 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 `false` ausgewertet wird, 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 Überprüfungen. Sie erstellen eine Beispiel-Website, die auf einer Auto 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-Gruppe können so konfiguriert werden, dass sie auf einem beliebigen Port lauschen (8888 ist die Standardeinstellung).

Die Regeln in dieser Vorlage validieren die Eingabeparameter vor der Erstellung des Stapels. Sie überprüfen, ob alle Subnetze zur angegebenen VPC gehören und stellen sicher, dass, wenn der Parameter UseSSL auf Yesgesetzt wird, sowohl ein SSL-Zertifikat-ARN als auch ein Name der gehosteten Zone angegeben werden.

Note

Wenn Sie einen Stack anhand dieser Vorlage 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

```

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 zum Beispiel in Ihrer Vorlage einen Eingabeparameter `EnvironmentType` hinzufügen, der entweder `prod` oder `test` als Eingabe akzeptiert. Für die `prod`-Umgebung können Sie EC2-Instances mit bestimmten Funktionen einbeziehen, während Sie für die `test`-Umgebung vielleicht reduzierte Funktionen verwenden, um Geld zu sparen. Mit dieser Bedingungsdefinition können Sie festlegen, welche Ressourcen erstellt werden und wie sie für jeden Umgebungstyp konfiguriert werden.

Syntax

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

JSON

```
"Conditions": {  
  "LogicalConditionName1": {  
    "Intrinsic function": ...  
  },  
  
  "LogicalConditionName2": {  
    "Intrinsic function": ...  
  }  
}
```

YAML

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

  LogicalConditionName2:
    Intrinsic function:
      ...
```

Wie Bedingungen funktionieren

Um die Bedingungen zu verwenden, gehen Sie folgendermaßen vor:

1. Fügen Sie eine Parameterdefinition hinzu- Definieren Sie die Eingaben, die Ihre Bedingungen auswerten werden, im Abschnitt `Parameters` Ihrer Vorlage. Die Bedingungen werden auf der Grundlage der Werte dieser Eingabeparameter als wahr oder falsch ausgewertet. Beachten Sie, dass Pseudo-Parameter automatisch verfügbar sind und keine explizite Definition im Abschnitt `Parameters` 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 Abschnitt `Conditions` mit Hilfe von intrinsischen Funktionen wie `Fn::If` oder `Fn::Equals`. Diese Bedingungen bestimmen, wann die zugehörigen Ressourcen CloudFormation erstellt werden. Die Bedingungen können sich auf Folgendes beziehen:
 - Eingabe oder Pseudo-Parameterwerte
 - Andere Bedingungen
 - Werte mappen

Sie können jedoch in Bedingungen nicht auf logische Ressourcen IDs oder deren Attribute verweisen.
3. Bedingungen mit Ressourcen oder Ausgängen verknüpfen- Bedingungen in Ressourcen oder Ausgängen mit dem Schlüssel `Condition` und der logischen ID einer Bedingung referenzieren. Verwenden Sie optional `Fn::If` in anderen Teilen der Vorlage (z. B. bei Eigenschaftswerten), um Werte auf der Grundlage einer Bedingung festzulegen. Weitere Informationen finden Sie unter [Verwendung der Taste `Condition`](#).

CloudFormation wertet Bedingungen aus, wenn ein Stack 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 mit einer wahren Bedingung verbunden bleiben, werden aktualisiert, während diejenigen, die mit einer falschen Bedingung verbunden werden, 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.

Verwendung der Taste **Condition** .

Sobald eine Bedingung definiert ist, können Sie sie an mehreren Stellen in der Vorlage anwenden, beispielsweise `Resources` und `Outputs`, indem Sie den Schlüssel `Condition` verwenden.

Der Schlüssel `Condition` verweist auf den logischen Namen einer Bedingung und gibt das ausgewertete Ergebnis der angegebenen Bedingung zurück.

Themen

- [Bedingungen mit Ressourcen verknüpfen](#)
- [Bedingungen mit Ausgaben verknüpfen](#)
- [Referenzbedingungen in anderen Bedingungen](#)
- [Bedingte Rückgabe von Eigenschaftswerten mit `Fn::If`](#)

Bedingungen mit Ressourcen verknüpfen

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
```

Bedingungen mit Ausgaben verknüpfen

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

Wenn Sie Bedingungen im Abschnitt `Conditions` definieren, können Sie mit dem Schlüssel `Condition` auf andere Bedingungen verweisen. Damit können Sie eine komplexere bedingte Logik erstellen, indem Sie mehrere Bedingungen kombinieren.

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

Bedingte Rückgabe von Eigenschaftswerten mit **Fn::If**

Für eine genauere Kontrolle können Sie die intrinsische Funktion `Fn::If` verwenden, um bedingt einen von zwei Eigenschaftswerten innerhalb von Ressourcen oder Ausgaben 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 Sie einen EC2-Instance-Typ auf der Grundlage einer Umgebungsbedingung festlegen. Wenn die Bedingung `IsProduction` als wahr ausgewertet wird, wird der Instancetyp auf `c5.xlarge` gesetzt. Andernfalls wird er auf `t3.small` gesetzt.

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 auch den Pseudoparameter `AWS::NoValue` 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

- [Umweltbasierte Ressourcenerstellung](#)
- [Ressourcenbereitstellung unter mehreren Bedingungen](#)

Umweltbasierte Ressourcenerstellung

In den folgenden Beispielen wird eine EC2-Instance bereitgestellt und ein neues EBS-Volume nur dann erstellt und angehängt, wenn der Umgebungstyp `prod` ist. Wenn die Umgebung `test` ist, wird die EC2-Instance 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
```

Ressourcenbereitstellung unter mehreren Bedingungen

Die folgenden Beispiele erstellen einen S3-Bucket, wenn ein Bucket-Name angegeben wird, und fügen nur dann eine Bucket-Richtlinie hinzu, wenn die Umgebung auf `prod` eingestellt ist. Wenn kein Bucket-Name angegeben wird oder die Umgebung `test` lautet, 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 Bedingung `CreateBucketPolicy`, wie Sie mit dem Schlüssel `Condition` auf andere Bedingungen verweisen können. Die Richtlinie wird nur erstellt, wenn die Bedingungen `IsProduction` und `CreateBucket` als wahr bewertet werden.

 Note

Komplexere Beispiele für die Verwendung von Bedingungen finden Sie im Thema [Condition Attribut](#) im CloudFormation Template Reference Guide.

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 über die Funktionsweise von Makros finden Sie unter [Führen Sie eine benutzerdefinierte Verarbeitung von CloudFormation Vorlagen mit Vorlagenmakros durch](#).

CloudFormation unterstützt auch Transformationen, bei denen es sich um Makros handelt, die von bereitgestellt werden. CloudFormation CloudFormation behandelt diese Transformationen in Bezug auf Ausführungsreihenfolge und Umfang genauso wie alle Makros, die Sie erstellen. Weitere Informationen finden Sie unter [Transform-Referenz](#).

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

Im folgenden Vorlagenbeispiel wird beispielsweise CloudFormation ausgewertet MyMacro und dannAWS::Serverless, beide können den Inhalt der gesamten Vorlage verarbeiten, da sie im Abschnitt enthalten sind. Transform

```
# 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 Abschnitt `AWSTemplateFormatVersion` (optional) gibt die Version des Vorlagenformats an, mit dem die Vorlage übereinstimmt. Die neueste Vorlagenformatversion ist die Version `2010-09-09`. Dies ist derzeit der einzige gültige Wert.

Die Version des Vorlagenformats ist nicht die gleiche wie die API-Version. Die Version des Vorlagenformats kann sich unabhängig von den API-Versionen ändern.

Der zur Deklaration 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

Der Abschnitt `Description` (optional) ermöglicht es Ihnen, eine Textzeichenfolge einzufügen, 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.
```

Visuelle Erstellung von Vorlagen mit Infrastructure Composer

AWS Infrastructure 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 & 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 CloudFormation Designer.

Mit Infrastructure Composer im CloudFormation Konsolenmodus können Sie eine Vielzahl von Ressourcen, so genannten Karten, 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 unter [Wie man in AWS Infrastructure Composer](#) komponiert, im AWS Infrastructure Composer Entwicklerhandbuch.

Warum Infrastructure Composer im CloudFormation Konsolenmodus verwenden?

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, einsatzfähigen Code erstellen und Ihre Entwickler-Workflows mit dem visuellen Designer im Infrastructure Composer einbinden.

Wie unterscheidet sich dieser Modus von dem 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 Auch die lokale Synchronisierung ist in diesem Modus nicht verfügbar.

Sie können jedoch diese Lambda-bezogenen Karten und das lokale Synchronisierungsfeature in der [Infrastructure Composer-Konsole](#) oder in der AWS Toolkit for Visual Studio Code verwenden. Weitere Informationen finden Sie im [AWS Infrastructure Composer Entwicklerhandbuch](#) und [Infrastructure Composer](#) im AWS Toolkit for Visual Studio Code -Benutzerhandbuch.

AWS CloudFormation Sprachserver

Der AWS CloudFormation Sprachserver bietet Funktionen zur Beschleunigung der Inhaltserstellung infrastructure-as-code (IaC) und zur sicheren und zuverlässigen Bereitstellung von AWS Ressourcen. Es folgt dem [Language Server Protocol](#) (LSP) und bietet Dokumentation zu Hover, Autovervollständigung, Diagnose über statische Validierung, Go-to-Definitions- und Code-Aktionen. Zusätzlich zu diesen herkömmlichen Sprachserverfunktionen bietet der Server Online-Funktionen, mit denen Sie Ressourcen erkunden und bereitstellen können AWS . CloudFormation Dazu gehören die Möglichkeit, Vorlagen mithilfe von Änderungssätzen zu validieren und bereitzustellen, Stack-Diffs, Ereignisse, Ressourcen und Ausgaben anzuzeigen, Stapel aufzulisten und Ressourcen nach Typ zu

durchsuchen sowie den aktuellen Ressourcenstatus direkt in CloudFormation Vorlagen einzufügen. Entwicklertools und integrierte Entwicklungsumgebungen (IDE) können die Sprachserverfunktionen nutzen, um die Erfahrung von IaC-Entwicklern zu verbessern. CloudFormation

Themen

- [IDEs Integration mit dem Sprachserver AWS CloudFormation](#)
- [Erste Schritte](#)
- [Initialisieren eines CloudFormation Projekts in der IDE](#)
- [Open-Source](#)
- [Benötigen Sie Hilfe?](#)

IDEs Integration mit dem Sprachserver AWS CloudFormation

Die [AWS Toolkit-Erweiterung](#) integriert den CloudFormation Sprachserver, um das Authoring-Erlebnis zu verbessern. Diese Integration wird derzeit in VS Code Based unterstützt IDEs, einschließlich:

- [Visual Studio Code](#)
- [Kiro](#)
- [Mauszeiger](#)

Erste Schritte

Themen

- [Voraussetzungen](#)
- [Schritt 1: Installieren oder aktualisieren Sie das Toolkit AWS](#)
- [Schritt 2: Zugriff CloudFormation im Toolkit-Panel AWS](#)
- [Schritt 3: Validieren, testen und verfeinern Sie Ihre Vorlage](#)
- [Schritt 4: Navigieren Sie durch die Vorlage](#)
- [Schritt 5: Validieren und Bereitstellen](#)

Voraussetzungen

Bevor Sie beginnen, stellen Sie Folgendes sicher:

- Sie verwenden eine VS Code-basierte IDE auf einem unterstützten Betriebssystem (macOS, Windows, oder Linux).
- Sie haben die neueste Version des [AWS Toolkits](#) für Ihre IDE installiert oder auf diese aktualisiert.

Für einige Funktionen des AWS CloudFormation Sprachservers sind aktive AWS-Konto und konfigurierte Anmeldeinformationen erforderlich. Sie müssen AWS-Konto über das AWS Toolkit mit gültigen Anmeldeinformationen bei Ihrem angemeldet sein.

Schritt 1: Installieren oder aktualisieren Sie das Toolkit AWS

- Öffnen Sie den Extensions oder den Plugin-Manager Ihrer IDE.
- Suchen Sie nach AWS Toolkit.
- Installieren oder aktualisieren Sie das AWS Toolkit für Ihre IDE auf Version 3.85.0 oder höher.
- Starten Sie Ihre IDE nach der Installation neu.

Hinweis: Nach der Installation aktiviert das AWS Toolkit automatisch die CloudFormation IDE-Unterstützung. Wenn Sie das AWS Toolkit zum ersten Mal mit der AWS CloudFormation Sprachserver-Erweiterung installieren oder aktualisieren, werden Sie aufgefordert, die Erlaubnis zur Erfassung anonymer Nutzungsdaten AWS zu erteilen. Diese Daten tragen zur AWS Verbesserung des CloudFormation Sprachservers bei und verbessern das Authoring-Erlebnis. Es werden keine vertraulichen Informationen gesammelt und AWS es werden keine Vorlageninhalte, Ressourcenkonfigurationen oder identifizierbare Kundendaten aufgezeichnet oder gespeichert. Sie können Ihre Telemetrie-Einstellungen jederzeit in den IDE-Einstellungen ändern. Die gesammelten Nutzungsdaten konzentrieren sich nur auf Funktionsinteraktionen und Leistungskennzahlen. Diese Erkenntnisse helfen dabei, Verbesserungen wie schnellere Validierung, verbesserte automatische Vervollständigung und bessere Fehlerdiagnose zu AWS identifizieren und zu priorisieren.

Schritt 2: Zugriff CloudFormation im Toolkit-Panel AWS

Öffnen Sie in Ihrer IDE das AWS Toolkit-Bedienfeld über die Aktivitätsleiste. Wählen Sie unter AWS Toolkit die Option CLOUDFORMATION aus.

Panel-Abschnitte:

- **Region:** Zeigt die aktuelle Region an AWS-Region. Sie können ihn ändern, indem Sie den Namen der Region auswählen oder den Befehl AWS CloudFormation: Region auswählen in der Befehlspalette verwenden.

- **Umgebung:** Zeigt die Umgebung an, die Sie während dieser Zeit ausgewählt haben CFN init (z. B. dev, test oder prod). Dieser Wert wird erst angezeigt, nachdem der Schritt zur Umgebungsauswahl abgeschlossen wurde.
- **Stacks:** Zeigt eine paginierte Liste der CloudFormation Stacks in Ihrem Konto an.
 - Klicken Sie auf das Aktualisierungssymbol, um die Liste der Stacks zu aktualisieren.
 - Verwenden Sie das Pluszeichen, um eine neue Vorlage als Stapel bereitzustellen.
 - Jeder Stack-Eintrag beinhaltet:
 - Überblick: Zeigt die Zusammenfassung und den Status des Stacks an
 - Stack-ID
 - Beschreibung
 - Erstellungszeit und Aktualisierungszeit
 - Status und Statusgrund
 - Ereignisse, Ergebnisse und Ressourcen
 - Set ändern
- **Ressourcen:** Nachdem Sie einen Ressourcentyp ausgewählt haben, werden im Bereich die AWS Ressourcen dieses Typs in Ihrem Konto angezeigt. Sie können sie anzeigen, aktualisieren, kopieren oder in Ihre Vorlage importieren.

Schritt 3: Validieren, testen und verfeinern Sie Ihre Vorlage

Während Sie Ihre CloudFormation Vorlage schreiben, bietet die IDE intelligente Authoring-Unterstützung, damit Sie schneller eine genaue und konforme Infrastruktur erstellen können. Der CloudFormation Sprachserver läuft im Hintergrund und bietet die folgenden Authoring-Funktionen:

- **Codevervollständigung:** Schlägt Ressourcentypen, Parameter und Eigenschaften auf der Grundlage von CloudFormation Schemas vor.
- **Bestehende AWS Ressourcen hinzufügen:** Ermöglicht es Ihnen, vorhandene Ressourcen aus Ihrem AWS-Konto in Ihre Vorlage zu importieren. Die IDE verwendet die [AWS -Cloud-Control- API \(CCAPI\)](#), um die Live-Konfiguration und die Eigenschaften der Ressource abzurufen, sodass Sie die bestehende Infrastruktur innerhalb Ihrer Vorlage klonen oder wiederverwenden können.

Um Ressourcen zu Ihrer Vorlage hinzuzufügen

- Erweitern Sie den Bereich CloudFormation Ressourcen: Erweitern Sie im Seitenbereich des AWS Toolkits den Bereich und klicken Sie dann unter Ressourcen auf das Symbol „+ hinzufügen“.
- Nach einem Ressourcentyp suchen: Um einen bestimmten AWS Ressourcentyp zu finden, geben Sie in der Suchleiste den spezifischen AWS Ressourcentyp ein, den Sie hinzufügen möchten. Beispiel:
 - `AWS::S3::Bucket`
 - `AWS::Lambda::Function`

Sie können den Ressourcentyp mithilfe von AWS CloudFormation: Ressourcentypen hinzufügen in der Befehlspalette auswählen.

- Ressourcen zu Ihrer Vorlage hinzufügen: Im Bereich Ressourcen wird eine paginierte Liste der erkannten AWS Ressourcen in Ihrem Konto angezeigt. Wenn Sie über viele Ressourcen verfügen, wird nur die erste Seite angezeigt. Verwenden Sie die Navigationssteuerelemente am unteren Rand des Panels, um durch weitere Seiten zu blättern und alle verfügbaren Ressourcen anzuzeigen.
- Wählen Sie die Ressource aus, die Sie in Ihre Vorlage aufnehmen möchten.
- Sie können eine Ressource auf zwei Arten in Ihre Vorlage einfügen, abhängig von Ihrem Ziel:
 - Klonen Sie eine vorhandene Ressource: Erstellen Sie eine neue Ressource in Ihrer Vorlage, indem Sie die Live-Konfiguration und die Eigenschaften einer vorhandenen AWS Ressource verwenden.
 - Importieren Sie eine vorhandene Ressource: Fügen Sie die tatsächliche Ressource in Ihren Stapel ein, indem Sie sie unter Verwendung ihres Live-Status zu Ihrer Vorlage hinzufügen.

Tipps

- Sie können das Ressourcen-Panel jederzeit aktualisieren, um die aktuelle Liste der in Ihrem Konto oder Ihrer Region verfügbaren Ressourcen einzusehen.
- Wenn Sie Ressourcen importieren, fügen Sie keine Ressource hinzu, die bereits zu einem vorhandenen CloudFormation Stack im selben Konto gehört.
- Um zu überprüfen, ob eine Ressource bereits von verwaltet wird CloudFormation, klicken Sie auf das i-Symbol (Information) neben dem Ressourcennamen.

- Alternativ können Sie den Befehl AWS CloudFormation Get Stack Management Info verwenden, um zu überprüfen, ob eine Ressource zu einem Stack gehört.

Fügen Sie verwandte Ressourcen hinzu

Sie können der ausgewählten Ressource verwandte Ressourcen hinzufügen, indem Sie den folgenden Befehl verwenden AWS CloudFormation: Verwandte Ressourcen nach Typ hinzufügen. Sobald Sie einen Ressourcentyp aus den bereits in Ihrer Vorlage definierten ausgewählt haben, zeigt die IDE eine Liste von Ressourcen an, die normalerweise mit diesem Typ verknüpft sind oder von diesem abhängig sind. Wenn Sie beispielsweise einen auswählen `AWS::EC2::Instance`, schlägt die IDE möglicherweise vor, verwandte Ressourcen wie `AWS::EC2::SecurityGroup` oder hinzuzufügen `AWS::EC2::Subnet`. Mit dieser Funktion können Sie schnell verbundene Infrastrukturkomponenten erstellen, ohne manuell nach kompatiblen Ressourcentypen suchen zu müssen.

Statische Validierung

Die CloudFormation IDE bietet eine integrierte statische Validierung, die von [AWS CloudFormation Linter \(cfn-lint\)](#) und unterstützt wird. [AWS CloudFormation Guard](#) Diese Validierungen laufen im Hintergrund ab, während Sie Vorlagen erstellen, und helfen Ihnen dabei, Syntaxfehler, Compliance-Lücken und Probleme mit bewährten Verfahren vor der Implementierung zu identifizieren.

Überblick über die statische Validierung

In der IDE werden Sie zwei Arten von statischen Echtzeitvalidierungen sehen:

- CloudFormation Linter (`cfn-lint`): Validiert Ihre Vorlage anhand von CloudFormation Ressourcenspezifikationen und Schemaregeln.
- Guard (`cfn-guard`): Überprüft Ihre Vorlage anhand von Compliance-Regeln und organisatorischen Policy-Paketen.

CloudFormation Linter (cfn-lint)

Der CloudFormation Linter ist in die IDE integriert, um die Syntax und Struktur Ihrer Vorlage während der Eingabe automatisch zu überprüfen.

- Schemavalidierung: Erkennt Syntax- und Schemafehler, um sicherzustellen, dass Ihre Vorlagen dem CloudFormation Ressourcenschema entsprechen.

- Fehlerhervorhebung: Zeigt rote oder gelbe, verschnörkelte Linien unter Problemen an, die für Blockierungen oder Warnungen bei der Bereitstellung stehen.
- Zeigen Sie mit der Maus auf die Hilfe: Wenn Sie mit der Maus auf einen Fehler zeigen, zeigt die IDE die zu diesem Problem gehörende Diagnosemeldung an. Wenn eine schnelle Lösung verfügbar ist, wird sie auch im Hover-Panel angeboten.

Integration von Guard

Guard validiert Ihre Vorlagen anhand von Regelsätzen, die Compliance- und Sicherheitsrichtlinien definieren. Die IDE führt Guard-Validierungen in Echtzeit über den CloudFormation Sprachserver durch, sodass Sie bei der Erstellung von Vorlagen sofort Feedback erhalten.

- Standardregelpakete: Die IDE enthält einen vorregistrierten Satz von Guard-Regeln, die sich auf grundlegende bewährte Methoden für Ressourcensicherheit und Konfigurationshygiene konzentrieren. Weitere Informationen finden Sie unter [dem Link zur Guard-Rule-Registrierung](#).
- Regelpakete hinzufügen: Um Regelsätze hinzuzufügen oder zu ändern, öffnen Sie Einstellungen, Guard und wählen Sie weitere Guard-Regelpakete aus oder laden Sie sie hoch.

Tipps: Verstehen von Indikatoren mit verschnörkelten Linien

- Blaue Schnörkel: Tipps zu bewährten Methoden oder Optimierungsempfehlungen.
- Gelbe Kringel: Warnungen vor Problemen, die nicht blockieren (z. B. fehlende Tags oder Parameter).
- Rote Kringel: Blockieren der Bereitstellung, z. B. ungültige Eigenschaftsnamen, fehlende Pflichtfelder oder Schemakonflikte.

Schritt 4: Navigieren Sie durch die Vorlage

Wenn Sie im Explorer auf eine Vorlagendatei klicken, zeigt das Übersichtsfenster automatisch eine strukturierte, hierarchische Ansicht Ihrer CloudFormation Vorlage an. Es organisiert die Vorlage in Abschnitte wie `Parameters`, `Resources` `Outputs` `Mappings`, und zeigt jeden Ressourcentyp und jede logische ID an. Dies macht es einfach, innerhalb umfangreicher Vorlagen schnell nach bestimmten Ressourcen oder Parametern zu suchen und zu ihnen zu navigieren.

Sie können `Go to Definition` für systeminterne Funktionen wie `GetAtt` und `verwendenRef`, sodass Sie direkt zu der Ressource oder dem Parameter springen können, auf die in Ihrer Vorlage verwiesen

wird. Auf diese Weise können Sie Abhängigkeiten nachverfolgen, Ressourcenbeziehungen verstehen und Änderungen effizienter vornehmen.

Schritt 5: Validieren und Bereitstellen

Wenn Sie bereit sind, Ihre CloudFormation Vorlage bereitzustellen, öffnen Sie die Befehlspalette und führen Sie Folgendes aus AWS CloudFormation: Validieren und Bereitstellen. Der Befehl validiert Ihre Vorlage, und wenn keine Blockierungsfehler gefunden werden, wird mit der Bereitstellungsphase fortgefahren. Vor der Bestätigung zeigt die IDE eine Zusammenfassung des [Änderungssatzes](#) und eine Vergleichsansicht an, sodass Sie alle vorgeschlagenen Änderungen überprüfen können.

Wie funktioniert die Validierung

Die IDE führt [vor der Bereitstellung automatisch eine Validierungsprüfung](#) durch und validiert Ihre Vorlage anhand gängiger Fehlerursachen, darunter:

- Ungültige Eigenschaftssyntax oder falsche Schemaübereinstimmungen: Diese Probleme treten normalerweise `cfn-lint` beim Authoring auf. Wenn ein Benutzer jedoch mit der Bereitstellung fortfährt, ohne sie zu adressieren, werden bei CloudFormation der Validierung während der Bereitstellung dieselben Fehler angezeigt, bevor der Stack erstellt oder aktualisiert wird.
- Der Ressourcenname steht in Konflikt mit vorhandenen Ressourcen in Ihrem Konto.
- Dienstspezifische Einschränkungen, wie z. B. Konflikte mit S3-Bucket-Namen oder fehlende Verschlüsselung.

Wenn bei der Überprüfung Fehler erkannt werden, hebt die IDE die Probleme direkt in Ihrer Vorlage anhand roter oder gelber Schnörkellinien hervor und listet die Fehler im Bereich PROBLEME auf. Jedes Problem enthält die spezifische Eigenschaft oder Ressource, die den Fehler verursacht hat, sowie einen Lösungsvorschlag. Wenn keine Blockierungsfehler vorliegen, können Sie mit der Bereitstellungsphase fortfahren.

Wenn Warnungen gefunden werden (nicht blockierende Probleme), wird ein Dialogfeld angezeigt, in dem Sie entweder mit der Bereitstellung fortfahren oder den Vorgang abbrechen und Korrekturen vornehmen können.

Die IDE öffnet einen [Änderungssatz, der](#) Änderungen berücksichtigt und alle Unterschiede zwischen Ihrer aktuellen Vorlage und der bereitgestellten Stack-Konfiguration anzeigt. Auf diese Weise können Sie den Änderungssatz vor der Ausführung überprüfen, bestätigen oder stornieren.

Hinweis: Änderungssätze, die Drift berücksichtigen, verbessern den CloudFormation Bereitstellungsprozess, da sie es Ihnen ermöglichen, Stack-Drift sicher zu handhaben. Stack-Drift tritt auf, wenn der tatsächliche Status Ihrer Ressourcen von dem abweicht, was in Ihrer CloudFormation Vorlage definiert ist, was häufig auf manuelle Änderungen zurückzuführen ist AWS-Managementkonsole, die über die CLI oder das SDK vorgenommen wurden. CloudFormationIm [Änderungssatz mit Drift-Erkennung](#) wird Ihre verarbeitete Stack-Konfiguration mit dem aktuellen Ressourcenstatus verglichen, und die IDE deckt diese Unterschiede auf, sodass Sie die Ressourcen vor der Bereitstellung wieder an die Richtlinien anpassen können.

Stack-Ereignisse ansehen

Wenn die Bereitstellung beginnt, können Sie zur CloudFormationRegisterkarte im Fenster wechseln, um den Fortschritt in Echtzeit zu überwachen. Unter Stack-Ereignisse sehen Sie eine Liste der während der Bereitstellung ausgeführten Operationen. Jedes Ereignis enthält Details wie:

- Zeitstempel: Der Zeitpunkt, zu dem das Ereignis eingetreten ist
- Ressource: Die spezifische AWS Ressource, die erstellt, aktualisiert oder gelöscht wird
- Status: Der aktuelle Status des Vorgangs (z. B. CREATE_IN_PROGRESS, UPDATE_COMPLETE, oder ROLLBACK_IN_PROGRESS)
- Grund: Zusätzliche Kontext- oder Fehlermeldungen, falls zutreffend

In diesem Bereich können Sie sich auch die Ressourcen und Ausgaben des Stacks ansehen. Wenn Sie den Stapel im öffnen möchten AWS-Managementkonsole, verwenden Sie das externe Linksymbol neben dem Stacknamen. In der Ansicht Stack-Ereignisse können Sie den Bereitstellungsfortschritt verfolgen, potenzielle Probleme identifizieren und überprüfen, wann Ihr Stack erfolgreich abgeschlossen wurde.

Initialisieren eines CloudFormation Projekts in der IDE

Durch die Initialisierung eines CloudFormation Projekts in der IDE können Sie einen strukturierten Arbeitsbereich mit den richtigen Ordnern, der richtigen Umgebungskonfiguration und den richtigen AWS Anmeldeinformationen einrichten, sodass Sie Ihre Vorlagen zuverlässig validieren und bereitstellen können. Sie können ein neues CloudFormation Projekt direkt von der IDE aus initialisieren, um dieses empfohlene Setup zu erstellen.

Um ein Projekt zu initialisieren: CloudFormation

- Öffnen Sie die Befehlspalette

- Öffnen Sie in Ihrer IDE die Befehlspalette (Ctrl+Shift+P oder Cmd+Shift+P aktiviert macOS).
- Wählen Sie AWS CloudFormation: CFN Init: Projekt initialisieren.
- Wählen Sie ein Projektverzeichnis
 - Standardmäßig verwendet die IDE Ihr aktuelles Arbeitsverzeichnis.
 - Sie können diesen Pfad in einen beliebigen Ordner ändern, in dem Sie Ihre CloudFormation Vorlagen speichern möchten.
- Wählen Sie Ihr AWS Anmeldeinformationsprofil
 - Sie werden aufgefordert, ein AWS Anmeldeinformationsprofil auszuwählen. Das ausgewählte Profil wird für die Erkennung, Validierung und Bereitstellung von Umgebungen verwendet.
- Richten Sie Ihre Umgebung ein
 - Sie werden aufgefordert, eine Umgebung zu erstellen oder auszuwählen.
 - Umgebungen definieren, wo und wie Ihre Vorlagen bereitgestellt oder validiert werden (z. B. Entwicklung, Beta oder Produktion). Sie können AWS CloudFormation: CFN Init: Add Environment verwenden, um Ihre Umgebung auszuwählen oder zu ändern.
 - Sie können AWS CloudFormation: CFN Init: Remove Environment verwenden, um die von Ihnen ausgewählte Umgebung zu entfernen.
- (Optional) Parameterdateien importieren
 - Wenn Sie bereits über vorhandene Parameterdateien verfügen, ermöglicht Ihnen die IDE, diese während der Initialisierung zu importieren.
 - Die IDE erkennt automatisch kompatible Dateien und verknüpft sie mit Ihrem Projekt, sodass sie bei der Validierung und Bereitstellung von Vorlagen verwendet werden können.
- Benennen Sie das Projekt und schließen Sie es ab
 - Geben Sie einen Projektnamen ein, z. B. Beta-Umgebung, und schließen Sie das Setup ab.
 - Die IDE erstellt die anfängliche Projektstruktur und die Konfigurationsdatei für Sie.

Sie können direkt von der IDE aus Validierungen ausführen, Bereitstellungen in der Vorschau anzeigen oder zwischen Umgebungen wechseln.

Open-Source

Der AWS CloudFormation Sprachserver ist im Rahmen der Apache-2.0 Lizenz als Open-Source-Lösung verfügbar, sodass Kunden volle Transparenz darüber haben, wie Vorlagendiagnosen, Schemavalidierung und statische Analysen durchgeführt werden. Dies reduziert die Probleme

bei Sicherheit und Compliance für Kunden, die vor der Einführung von Tools Transparenz auf Quellenebene benötigen.

Die Codebasis ist öffentlich verfügbar unter: [cloudformation-languageserver/ GitHub. https://github.com/aws-cloudformation/](https://cloudformation-languageserver.github.com/aws-cloudformation/)

Benötigen Sie Hilfe?

[CloudFormation AWS re:Post Probiere](#) die Community an.

Erzeugen 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

Im Folgenden finden Sie die Vorteile des IaC-Generators:

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

Der Prozess der IaC-Generierung besteht aus den folgenden Schritten:

1. Ressourcen scannen- Der erste Schritt besteht darin, einen Scan Ihrer Ressourcen zu starten. Dieser Scan ist regionsweit und läuft nach 30 Tagen ab. Während dieser Zeit können Sie mehrere Vorlagen aus demselben Scan erstellen.
2. Erstellen Sie Ihre Vorlage- Um die Vorlage zu erstellen, haben Sie zwei Möglichkeiten:
 - Erstellen Sie eine neue Vorlage von Grund auf und fügen Sie die gescannten und 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.

Das IaC-Generatorfeature ist in allen kommerziellen Regionen verfügbar und unterstützt viele gängige AWS -Ressourcentypen. Eine vollständige Liste der unterstützten Ressourcen finden Sie unter [Unterstützung für Ressourcentypen](#).

Themen

- [Überlegungen](#)
- [Für das Scannen von Ressourcen erforderliche IAM-Berechtigungen](#)
- [Häufig verwendete Befehle zur Erstellung, 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](#)
- [Erstellen Sie einen CloudFormation Stapel aus gescannten Ressourcen](#)
- [Auflösen von schreibgeschützten Eigenschaften](#)

Überlegungen

Sie können JSON- oder YAML-Vorlagen für AWS Ressourcen generieren, auf die Sie Lesezugriff haben. Die Vorlagen für die IaC-Generator-Fähigkeit modellieren Cloud-Ressourcen zuverlässig und schnell, ohne eine Ressource Eigenschaft für Eigenschaft beschreiben zu müssen.

In der folgenden Tabelle sind die für das IaC-Generierungsfeature verfügbaren Quoten aufgeführt.

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 erstellt werden	5	5
Gleichzeitige Anzahl von Ressourcen, die für eine Vorlagenerstellung modelliert werden	5	5
Gesamtzahl der Ressourcen, die in einer Vorlage modelliert werden können	500	500
Maximale Anzahl von generierten 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 erforderliche IAM-Berechtigungen

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

- CloudFormation Berechtigungen zum Scannen
- Leseberechtigungen für AWS Zieldienste

Der Scanbereich ist auf Ressourcen beschränkt, auf die Sie Lesezugriff haben. Fehlende Berechtigungen führen nicht zum Scheitern des Scans, aber zum Ausschluss dieser Ressourcen.

Ein Beispiel für eine IAM-Richtlinie, die Berechtigungen zum Scannen und zur Vorlagenverwaltung erteilt, finden Sie unter [Erlauben Sie alle IaC-Generatoroperationen](#).

Häufig verwendete Befehle zur Erstellung, Verwaltung und Löschung von Vorlagen

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

- [start-resource-scan](#) um einen Scan der Ressourcen im Konto in einem zu starten AWS-Region.
- [describe-resource-scan](#) um den Fortschritt eines Ressourcenscans zu überwachen.
- [list-resource-scans](#) um die Ressourcenscans in einem aufzulisten AWS-Region.
- [list-resource-scan-resources](#) um die beim Ressourcenscan gefundenen Ressourcen aufzulisten.
- [list-resource-scan-related-resources](#), um die Ressourcen aufzulisten, die sich auf Ihre gescannten Ressourcen beziehen.
- [create-generated-template](#) um eine CloudFormation Vorlage aus einer Reihe gescannter Ressourcen zu generieren.
- [update-generated-template](#) um die generierte Vorlage zu aktualisieren.
- [describe-generated-template](#) um Informationen über eine generierte Vorlage zurückzugeben.
- [list-generated-templates](#) um alle generierten Vorlagen in Ihrem Konto und Ihrer aktuellen Region aufzulisten.
- [delete-generated-template](#) um 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 eine Ressourcenprüfung durchführen, um Ihre aktuellen Ressourcen und deren Beziehungen zu ermitteln.

Sie können eine Ressourcenprüfung mit einer der folgenden Optionen starten. Wenn Sie den IaC-Generator zum ersten Mal verwenden, empfehlen wir Ihnen 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 (Teilscan)**- Wählen Sie manuell aus, welche Ressourcentypen im aktuellen Konto und in der Region gescannt werden sollen. Diese Option ermöglicht einen schnelleren und gezielteren Scanprozess und ist daher ideal für die iterative Entwicklung von Vorlagen.

Nach Abschluss des Scans können Sie auswählen, welche Ressourcen und die damit verbundenen Ressourcen bei der Erstellung Ihrer Vorlage berücksichtigt werden sollen. Bei der Verwendung der partiellen Überprüfung sind verwandte Ressourcen bei der Vorlagengenerierung nur dann verfügbar, wenn entweder:

- Sie haben sie vor dem Start des Scans speziell ausgewählt, oder
- Sie wurden benötigt, um Ihre ausgewählten Ressourcentypen zu entdecken.

Wenn Sie beispielsweise `AWS::EKS::Nodegroup` auswählen, ohne `AWS::EKS::Cluster` auszuwählen, 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 wird der Scan nur die von Ihnen ausgewählten Ressourcen berücksichtigen.

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 erforderliche IAM-Berechtigungen](#).

Themen

- [Starten Sie eine Ressourcenprüfung \(Konsole\)](#)
- [Starten Sie eine Ressourcenprüfung \(AWS CLI\)](#)

Starten Sie eine Ressourcenprüfung (Konsole)

So starten Sie einen Ressourcenscan für alle Ressourcentypen (vollständiger Scan)

1. Öffnen Sie die [laC-Generator-Seite](#) der CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms das AWS-Region , das die zu scannenden Ressourcen enthält.
3. Wählen Sie im Bereich Scans die Option Einen neuen Scan starten und dann Alle Ressourcen scannen.

So starten Sie einen Ressourcenscan für bestimmte Ressourcentypen (Teilscan)

1. Öffnen Sie die [laC-Generatorseite](#) der CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms das AWS-Region , das die zu scannenden Ressourcen enthält.
3. Wählen Sie im Bereich Scans die Option Einen neuen Scan starten und dann Bestimmte Ressourcen scannen.
4. Wählen Sie im Dialogfeld Teilweise Überprüfung starten bis zu 100 Ressourcentypen aus und wählen Sie dann Überprüfung starten.

Starten Sie eine Ressourcenprüfung (AWS CLI)

So starten Sie einen Ressourcenscan für alle Ressourcentypen (vollständiger Scan)

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

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

Bei Erfolg gibt dieser Befehl die ARN des Scans zurück. Beachten Sie die ARN in der Eigenschaft ResourceScanId . Sie benötigen sie, um Ihre Vorlage zu erstellen.


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

So starten Sie einen Ressourcenscan für bestimmte Ressourcentypen (Teilscan)

1. Verwenden Sie den folgenden Befehl [cat](#) um die Ressourcentypen, die Sie scannen möchten, in einer JSON-Datei mit dem Namen `config.json` in Ihrem Home-Verzeichnis zu speichern. Nachfolgend sehen Sie eine Beispielkonfiguration, die nach Amazon EC2-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-scan](#) Befehl mit der `--scan-filters` Option zusammen mit der `config.json` Datei, die Sie erstellt haben, 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 die ARN des Scans zurück. Beachten Sie die ARN in der Eigenschaft `ResourceScanId` . Sie benötigen sie, um Ihre Vorlage zu erstellen.

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

So überwachen Sie den Fortschritt einer Ressourcenprüfung

Verwenden Sie den Befehl [describe-resource-scan](#). Ersetzen Sie bei der Option `--resource-scan-id` die Beispiel-ARN durch die tatsächliche 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 Teil-Scan sieht die Ausgabe ähnlich aus wie die folgende:

```
{
```

```
{
  "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 [DescribeResourceScan](#) in der AWS CloudFormation API-Referenz.

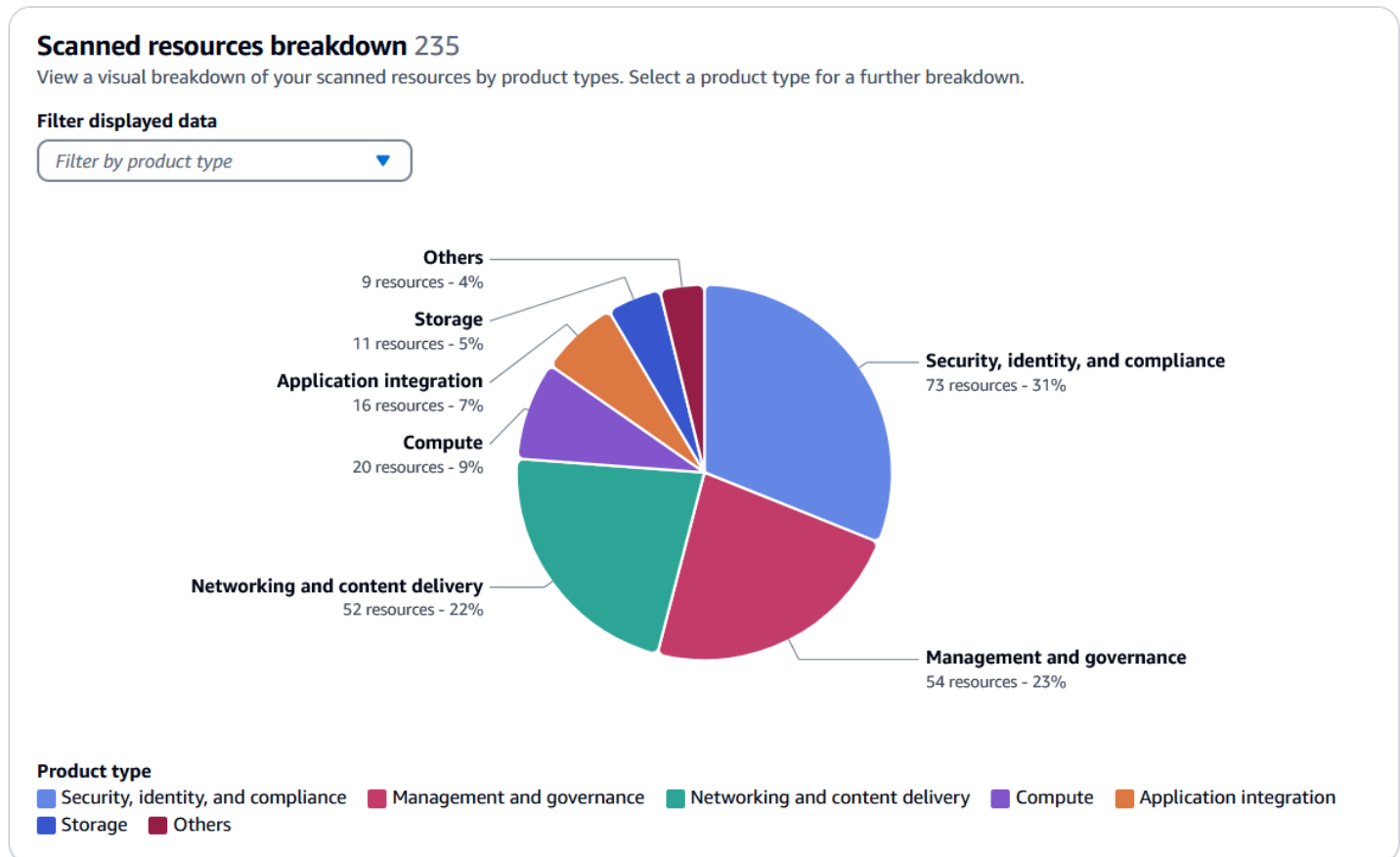
Die Scan-Zusammenfassung in der CloudFormation Konsole anzeigen

Nach Abschluss der Prüfung können Sie eine Visualisierung der bei der Prüfung gefundenen Ressourcen anzeigen, um die Konzentration der Ressourcen in den verschiedenen Produkttypen zu ermitteln.

Um Informationen über die bei der Suche gefundenen Ressourcen anzuzeigen

1. Öffnen Sie die [laC-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 laC-Generator.
4. Unter Aufschlüsselung der gescannten Ressourcen finden Sie eine visuelle Aufschlüsselung der gescannten Ressourcen nach Produkttyp, zum Beispiel Compute und Storage.

- Um die Anzahl der angezeigten Produkttypen anzupassen, wählen Sie Angezeigte Daten filtern. So können Sie die Visualisierung auf die Produkttypen zuschneiden, an denen Sie am meisten interessiert sind.
- Auf der rechten Seite der Seite befindet sich das Feld Details der Scan-Zusammenfassung. Um das Panel zu öffnen, wählen Sie das Symbol Panel 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 dem IaC-Generatorfeature gescannt wurden.

Erstellen Sie eine Vorlage aus gescannten Ressourcen (Konsole)

So erstellen Sie eine Stapelvorlage aus gescannten Ressourcen

- Öffnen Sie die [IaC-Generatorseite](#) der CloudFormation Konsole.

2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms das AWS-Region , das die gescannten Ressourcen enthält.
3. Wählen Sie im Bereich Vorlagen die Option Vorlage erstellen.
4. Wählen Sie Von einer neuen Vorlage beginnen.
 - a. Geben Sie für Vorlagenname einen Namen für Ihre Vorlage ein.
 - b. (Optional) Konfigurieren Sie Ihre Löschrichtlinie und Update-Ersatzrichtlinie.
 - c. Wählen Sie Weiter um gescannte Ressourcen zur Vorlage hinzuzufügen.
5. Unter Gescannte Ressourcen hinzufügen durchsuchen Sie die Liste der gescannten Ressourcen und wählen die Ressourcen aus, die Sie zu Ihrer Vorlage hinzufügen möchten. Sie können die Ressourcen nach Ressourcenbezeichner, Ressourcentyp oder Tags filtern. Die Filter schließen sich gegenseitig ein.
6. Wenn Sie alle benötigten Ressourcen zu Ihrer Vorlage hinzugefügt haben, wählen Sie Weiter um die Seite Gescannte Ressourcen hinzufügen zu verlassen und zur Seite Verwandte Ressourcen hinzufügen zu gelangen.
7. Sehen Sie sich eine empfohlene Liste mit verwandten Ressourcen an. Zusammenhängende Ressourcen, wie Amazon EC2-Instances und Sicherheitsgruppen, sind voneinander abhängig und gehören in der Regel zur gleichen Arbeitslast. Wählen Sie die zugehörigen Ressourcen aus, die Sie in die generierte Vorlage aufnehmen möchten.

 Note

Wir schlagen vor, dass Sie alle zugehörigen Ressourcen zu dieser Vorlage hinzufügen.

8. Sehen Sie sich die Details der Vorlage, die gescannten Ressourcen und die verwandten Ressourcen an.
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)

So erstellen Sie eine Stapelvorlage aus gescannten Ressourcen

1. Verwenden Sie den [list-resource-scan-resources](#) Befehl, um die beim Scan gefundenen Ressourcen aufzulisten, und geben Sie optional die `--resource-identifier` Option zur

Begrenzung der Ausgabe an. Ersetzen Sie bei der Option `--resource-scan-id` die Beispiel-ARN durch die tatsächliche 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 ein Antwortbeispiel, das `ManagedByStack` angibt, 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 [ScannedResource](#) in der AWS CloudFormation API-Referenz.

2. Verwenden Sie den Befehl `cat`, um die Ressourcentypen und Bezeichner in einer JSON-Datei namens `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 zu der JSON-Datei hinzu, die Sie im vorherigen Schritt erstellt haben. Sie benötigen sie, 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 [ScannedResource](#) in 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 Befehl `list-resource-scan-related-resources` in Stapeln von 100 Ressourcen aus und konsolidieren die Ergebnisse.

Beachten Sie, dass die Ausgabe möglicherweise doppelte Ressourcen in der Liste enthält.

4. Verwenden Sie den [create-generated-template](#) Befehl, 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.
 - Ersetzen Sie `MyTemplate` 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"
}
```

Erstellen Sie einen CloudFormation Stapel aus gescannten Ressourcen

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. So können Sie die gesamte Anwendungsarchitektur mit den Ressourcen und ihren Beziehungen visualisieren. Weitere Informationen über Infrastructure Composer finden Sie unter [Visuelle Erstellung von Vorlagen mit Infrastructure Composer](#).

So erstellen Sie den Stapel und importieren die gescannten Ressourcen

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 die Registerkarte Vorlagen und wählen Sie dann den Namen Ihrer Vorlage, um weitere Informationen anzuzeigen.
4. Auf der Registerkarte Schablonendefinition oben im Abschnitt Schablone können Sie die Schablone je nach Vorliebe von der YAML- zur JSON-Syntax umschalten.
5. Überprüfen Sie die Details Ihrer Vorlage, um sicherzustellen, dass alles richtig eingestellt ist. Um das Überprüfen und Verstehen der Vorlage zu erleichtern, können Sie mit dem Infrastructure Composer von der Standard-Codeansicht zu einer grafischen Ansicht der in der Vorlage beschriebenen Infrastruktur wechseln. Wählen Sie dazu unter Vorlagedie Option Leinwand anstelle von Vorlage.

Leinwand-Aktionen

- Wenn Sie sich auf die Details einer bestimmten Ressource innerhalb Ihrer Vorlage konzentrieren möchten, doppelklicken Sie auf eine Karte, um das Panel Ressourceneigenschaften aufzurufen.
 - Um die Karten auf der Leinwand visuell anzuordnen und zu organisieren, wählen Sie Anordnen oben links auf der Leinwand.
 - Um Ihre Leinwand zu vergrößern oder zu verkleinern, verwenden Sie die Zoomsteuerung unten rechts auf der Leinwand.
6. Um eine bestimmte Ressource in der Konsole anzuzeigen, wählen Sie die Registerkarte Vorlagenressourcen und dann die physische ID der Ressource, die Sie sich ansehen möchten. Dadurch gelangen Sie zur Konsole für diese spezielle Ressource. Sie können auch Ressourcen in der Vorlagendefinition auf der Registerkarte Vorlagenressourcen hinzufügen, entfernen und neu synchronisieren.
 7. Auf der Registerkarte Vorlagendefinition gibt der IaC-Generator möglicherweise Warnungen zu Ressourcen aus, die schreibgeschützte Eigenschaften enthalten. Nachdem Sie die Warnungen überprüft haben, können Sie die generierte Vorlage herunterladen und die erforderlichen Änderungen vornehmen. Weitere Informationen finden Sie unter [Auflösen von schreibgeschützten Eigenschaften](#).
 8. Wenn Sie mit Ihrer Vorlagendefinition zufrieden sind, wählen Sie auf der Registerkarte Vorlagendefinition die Option In Stapel importieren und dann Weiter.
 9. Geben Sie auf der Seite Stack-Details an im Bereich Stack angeben den Namen Ihres Stacks ein und wählen Sie anschließend Weiter.

10. Überprüfen Sie die Parameter für den Stapel und geben Sie sie ein. Wählen Sie Weiter aus.
11. Überprüfen Sie Ihre Optionen auf der Seite Änderungen überprüfen und wählen Sie Weiter.
12. Überprüfen Sie Ihre Angaben auf der Seite Überprüfen und importieren und wählen Sie Ressourcen importieren.

Auflösen von schreibgeschützten Eigenschaften

Mit dem CloudFormation IaC-Generator können Sie eine Vorlage mithilfe von Ressourcen generieren, die in Ihrem Konto bereitgestellt wurden und die noch nicht verwaltet werden. 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. Damit können Sie die Eigenschaften als Parameterwerte bei Importvorgängen eingeben. Es gibt jedoch Szenarien, in denen diese Konvertierung nicht möglich ist, und CloudFormation behandelt diese Fälle unterschiedlich.

Sich gegenseitig ausschließende Eigenschaften

Einige Ressourcen haben mehrere Sätze von sich gegenseitig ausschließenden Eigenschaften, von denen zumindest einige schreibgeschützt sind. In diesen Fällen kann der IaC-Generator nicht feststellen, welcher Satz von exklusiven Eigenschaften bei der Erstellung auf die Ressource angewendet wurde. Sie können zum Beispiel den Code für [AWS::Lambda::Function](#) mit einer dieser Eigenschaftsgruppen angeben.

- Code/S3Bucket,,Code/S3Key und optional Code/S3ObjectVersion
- Code/ImageUri
- Code/ZipFile

Alle diese Eigenschaften sind schreibgeschützt. Der IaC-Generator wählt eine der exklusiven Eigenschaftsgruppen aus und fügt sie der generierten Vorlage hinzu. Für jede der schreibgeschützten Eigenschaften werden Parameter hinzugefügt. Die Parameternamen enthalten OneOf und die Parameterbeschreibungen geben an, dass die entsprechende Eigenschaft durch andere

exklusive Eigenschaften ersetzt werden kann. Der IaC-Generator setzt den Warnungstyp `MUTUALLY_EXCLUSIVE_PROPERTIES` für die enthaltenen Eigenschaften.

Sich gegenseitig ausschließende Typen

In manchen Fällen kann eine schreibgeschützte Eigenschaft mehrere Datentypen haben. Zum Beispiel kann die Eigenschaft `Body` von [AWS::ApiGateway::RestApi](#) entweder ein `object` oder ein `string` sein. Wenn dies der Fall ist, fügt der IaC-Generator die Eigenschaft in die generierte Vorlage mit dem Typ `string` ein und setzt einen Warnungstyp `MUTUALLY_EXCLUSIVE_TYPES`.

Array-Eigenschaften

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

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 der Warnungstyp `UNSUPPORTED_PROPERTIES` wird festgelegt.

Warnungen und nächste Schritte

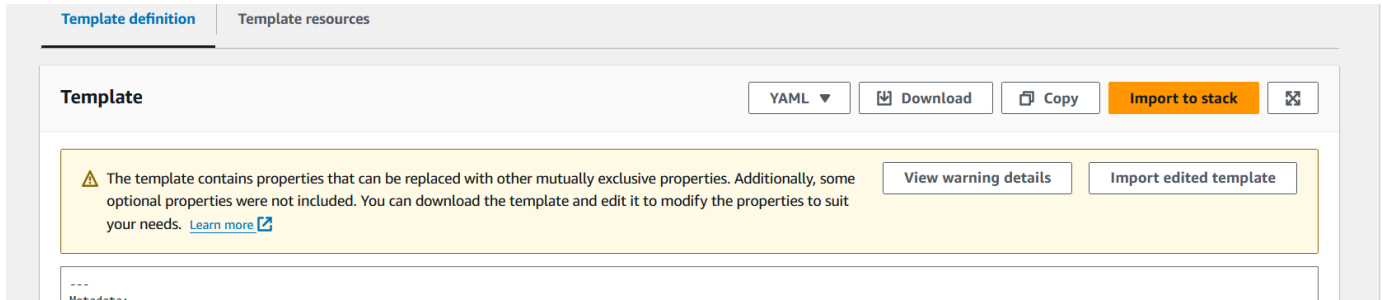
Um festzustellen, welche Eigenschaften schreibgeschützt sind, müssen Sie sich die Warnungen ansehen, die die Konsole des IaC-Generators ausgibt. Die Referenz [AWS Ressourcen- und Eigenschaftstypen](#) zeigt nicht an, ob eine Eigenschaft schreibgeschützt ist oder ob sie mehrere Typen unterstützt.

Alternativ können Sie in den Schemata der Ressourcenanbieter sehen, welche Eigenschaften schreibgeschützt sind. Informationen zum Herunterladen der Ressourcenanbieter-Schemas finden Sie in den [CloudFormation Ressourcenanbieter-Schemas](#).

So lösen Sie Probleme mit schreibgeschützten Eigenschaften

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 die Registerkarte Vorlagen und wählen Sie dann den Namen der von Ihnen erstellten Vorlage.
4. Wenn die generierte Vorlage Ressourcen mit schreibgeschützten Eigenschaften enthält, zeigt die Konsole des IaC-Generators auf der Registerkarte Vorlagendefinition eine Warnung mit einer Zusammenfassung der Art der Probleme an. Beispiel:



5. Wählen Sie Details zur Warnung anzeigen für weitere Details. Die Ressourcen mit schreibgeschützten Eigenschaften werden durch die in der generierten Vorlage verwendete logische ID und den Ressourcentyp identifiziert.

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

Resource warnings (1)
×

▼ LambdaFunction00laClambda002fD6K (AWS::Lambda::Function) [Learn more](#)

Unsupported properties	Mutually exclusive properties
1. SnapStart/ApplyOn Applying SnapStart setting on function resource type. Possible values: [PublishedVersions, None] This property can be replaced with other types	1. Code/S3Bucket An Amazon S3 bucket in the same AWS Region as your function. The bucket can be in a different AWS account. This property can be replaced with other exclusive properties
2. Code/S3ObjectVersion For versioned objects, the version of the deployment package object to use. This property can be replaced with other exclusive properties	2. Code/S3Key The Amazon S3 key of the deployment package. This property can be replaced with other exclusive properties

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

Wie Sie Probleme mit schreibgeschützten Eigenschaften in AWS::ApiGateway::RestAPI-Ressourcen beheben

In diesem Thema wird erklärt, wie Sie Probleme mit schreibgeschützten Eigenschaften in [AWS::ApiGateway::RestApi](#)-Ressourcen beheben, wenn Sie den IaC-Generator verwenden.

Problem

Wenn eine generierte Vorlage `AWS::ApiGateway::RestApi`-Ressourcen enthält, werden Warnungen erzeugt, die besagen, dass die Eigenschaften `Body`, `BodyS3Location` und `CloneFrom` als `UNSUPPORTED_PROPERTIES` identifiziert werden. Dies liegt daran, dass es sich um optionale schreibgeschützte Eigenschaften handelt. Der IaC-Generator weiß nicht, ob diese Eigenschaften jemals auf die Ressource angewendet wurden. Daher lässt es diese Eigenschaften in der generierten Vorlage weg.

Auflösung

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

1. Verwenden Sie die Amazon API Gateway [GetExport](#) API-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-Entwicklerleitfaden.
2. Rufen Sie die Eigenschaft `Body` aus der Antwort der API-Aktion `GetExport` ab. Laden Sie sie in einen Amazon S3-Bucket hoch.
3. Laden Sie die generierte Vorlage herunter.
4. Fügen Sie die Eigenschaften `BodyS3Location/Bucket` und `BodyS3Location/Key` zur Vorlage hinzu und geben Sie den Bucket-Namen und den Schlüssel an, in dem `Body` gespeichert ist.
5. Öffnen Sie die generierte Vorlage in der Konsole des IaC-Generators und wählen Sie `Bearbeitete Vorlage importieren`.

Wie Sie Probleme mit schreibgeschützten Eigenschaften in AWS::Lambda::Function-Ressourcen beheben

In diesem Thema wird erklärt, wie Sie Probleme mit schreibgeschützten Eigenschaften in [AWS::Lambda::Function](#)-Ressourcen beheben, wenn Sie den IaC-Generator verwenden.

Problem

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

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

Für eine bestimmte `AWS::Lambda::Function`-Ressource kann nur eines dieser Sets verwendet werden.

Der IaC-Generator kann nicht feststellen, welcher Satz von exklusiven schreibgeschützten Eigenschaften zum Erstellen oder Aktualisieren der Ressource verwendet wurde. Daher wird nur der erste Satz von Eigenschaften in die generierte Vorlage aufgenommen. Die Eigenschaften `Code/ImageUri` und `Code/ZipFile` werden weggelassen.

Außerdem gibt der IaC-Generator die folgenden Warnungen aus:

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

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

Auflösung

Wenn Sie Ihren Lambda-Code in einem Amazon S3-Bucket speichern und die **`S3ObjectVersion`** Eigenschaft nicht verwenden, können Sie die generierte Vorlage ohne Änderungen importieren. Der IaC-Generator wird Sie während des Importvorgangs nach dem Amazon S3-Bucket und dem Schlüssel als Vorlagenparameter fragen.

Wenn Sie Ihren Lambda-Code in einem Amazon ECR-Repository speichern, können Sie Ihre Vorlage anhand der folgenden Anweisungen aktualisieren:

1. Laden Sie die generierte Vorlage herunter.

2. Entfernen Sie die Eigenschaften und die entsprechenden Parameter für die Eigenschaften `Code/S3Bucket` und `Code/S3Key` aus der generierten Vorlage.
3. Ersetzen Sie die entfernten Eigenschaften in der generierten Vorlage durch die Eigenschaft `Code/ImageUri`, die die URL für das Amazon ECR-Repository angibt.
4. Öffnen Sie die generierte Vorlage in der IaC-Generator-Konsole und wählen Sie die Schaltfläche **Bearbeitete Vorlage importieren**.

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

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

Wenn Sie keine Kopie Ihres Lambda-Codes haben, können Sie Ihre Vorlage anhand 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 lautet der Parameter `RepositoryType` `S3`, wenn sich der Code in einem Amazon S3-Bucket befindet, oder `ECR`, wenn sich der Code in einem Amazon ECR-Repository befindet.
3. In der Antwort enthält der Parameter `Location` 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-Bucket hoch.
5. Führen Sie einen Importvorgang mit der generierten Vorlage aus und geben Sie den Bucketnamen und den Schlüssel als Parameterwerte an.

Abrufen von in anderen Diensten gespeicherten Werten mit Hilfe von dynamischen Referenzen

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 Referenzen können Sie:

- Sichere Zeichenketten 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 beschränken- Beschränken Sie den Zugriff auf die Parameter des Parameterspeichers oder die Secrets Manager-Geheimnisse auf autorisierte Principals und Rollen.
- Zugangsdaten rotieren- Rotieren Sie regelmäßig Ihre sensiblen Daten, die in Parameter Store oder Secrets Manager gespeichert sind, um ein hohes Maß an Sicherheit zu gewährleisten.
- Automatisierte Rotation- Nutzen Sie die automatischen Rotationsfeatures von Secrets Manager, um Ihre sensiblen Daten regelmäßig zu aktualisieren und über Ihre 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.

Um herauszufinden, welche Ressourceneigenschaften den primären Bezeichner eines Ressourcentyps enthalten, lesen Sie die Dokumentation der Ressourcenreferenz für diese Ressource in [AWS Ressource und Eigenschaftstypen referenzieren](#). Im Abschnitt Return values (Rückgabewerte) 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 Zeichenkette des dynamischen Verweises an die Transformation übergeben und die Verweise werden aufgelöst, wenn Sie den Änderungssatz mithilfe der Vorlage ausführen.
- Dynamische Referenzen können nicht für sichere Werte (wie die im Parameter Store oder Secrets Manager gespeicherten) in benutzerdefinierten Ressourcen verwendet werden.
- Dynamische Referenzen werden auch in `AWS::CloudFormation::Init`-Metadaten und Amazon EC2 UserData-Eigenschaften nicht unterstützt.
- Erstellen Sie keine dynamische Referenz, die mit einem umgekehrten Schrägstrich (\) endet. CloudFormation kann diese Verweise nicht auflösen, was dazu führen wird, dass Stack-Operationen fehlschlagen.

Die folgenden Themen enthalten Informationen und andere Überlegungen zur Verwendung dynamischer Referenzen.

Themen

- [Abrufen eines Klartextwertes aus dem Systems Manager Parameter Store](#)
- [Abrufen eines sicheren String-Wertes aus dem Systems Manager Parameter Store](#)
- [Holen Sie ein Geheimnis oder einen geheimen Wert aus dem Secrets Manager](#)

Abrufen eines Klartextwertes aus dem Systems Manager Parameter Store

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 Parameterspeicher](#) im AWS Systems Manager -Benutzerhandbuch.

Um einen Klartextwert aus dem Parameterspeicher in Ihrer Vorlage zu verwenden, verwenden Sie eine dynamische Referenz `ssm`. Diese Referenz ermöglicht Ihnen den Zugriff auf Werte von Parametern des Typs `String` oder `StringList` im Parameter Store.

Um zu überprüfen, welche Version einer dynamischen Referenz `ssm` in einer Stapeloperation verwendet wird, erstellen Sie ein Änderungsset für die Stapeloperation. 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 von dynamischen ssm-Referenzen gibt es einige wichtige Dinge zu beachten:

- CloudFormation unterstützt keine Drifterkennung bei dynamischen Referenzen. Bei dynamischen ssm-Referenzen, bei denen Sie die Parameterversion nicht angegeben haben, empfehlen wir, dass Sie, wenn Sie die Parameterversion in Systems Manager aktualisieren, auch einen Stackaktualisierungsvorgang für alle Stack durchführen, die die dynamische ssm-Referenz enthalten, um die neueste Parameterversion zu erhalten.
- Um eine ssm dynamische Referenz im `Parameters` Abschnitt Ihrer CloudFormation Vorlage zu verwenden, 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 auch als Systemmanager-Parametertyp in Ihrer Vorlage 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. Dies kann Ihre Vorlagen einfacher und leichter zu pflegen machen. 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 Anforderung 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 Parameter Store gespeicherten Parameter anzugeben, müssen Sie die Berechtigung haben, [GetParameters](#) für den angegebenen Parameter aufzurufen. Wie Sie IAM-Richtlinien erstellen, die den Zugriff auf bestimmte Systems Manager-Parameter ermöglichen, erfahren Sie unter [Einschränkung des Zugriffs auf Systems Manager-Parameter mit IAM-Richtlinien](#) im AWS Systems Manager -Benutzerhandbuch.

Referenzmuster

Verwenden Sie das folgende Referenzmuster, um auf einen Klartextwert zu ssm zu 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 unter [Arbeiten mit Parameterversionen](#) im AWS Systems Manager -Benutzerhandbuch.

Optional.

Beispiele

Themen

- [Öffentlicher AMI ID Parameter](#)
- [Benutzerdefinierter AMI-ID-Parameter](#)

Öffentlicher AMI ID Parameter

Das folgende Beispiel erstellt eine EC2-Instance, die auf einen öffentlichen AMI-Parameter verweist. Die dynamische Referenz ruft die neueste Amazon Linux 2023 AMI ID aus dem öffentlichen Parameter ab. Weitere Informationen über öffentliche Parameter finden Sie unter [Entdecken öffentlicher Parameter im Parameterspeicher](#) im AWS Systems Manager -Benutzerhandbuch.

JSON

```
{
```

```

"Resources": {
  "MyInstance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}",
      "InstanceType": "t2.micro"
    }
  }
}

```

YAML

```

Resources:
  MyInstance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}'
      InstanceType: t2.micro

```

Benutzerdefinierter AMI-ID-Parameter

Das folgende Beispiel erstellt eine EC2-Startvorlage, 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 Parameters *golden-ami* jedes Mal ab, wenn eine Instance aus der 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
```

Abrufen eines sicheren String-Wertes aus dem Systems Manager Parameter Store

In können Sie vertrauliche Daten wie Passwörter oder Lizenzschlüssel verwenden CloudFormation, ohne sie direkt in Ihren Vorlagen verfügbar zu machen, indem Sie die sensiblen Daten als „sichere Zeichenfolge“ im AWS Systems Manager Parameter Store speichern. Eine Einführung in den Parameterspeicher finden Sie unter [AWS Systems Manager Parameterspeicher](#) 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 speichert es nur die wörtliche dynamische Referenz, die den Klartext-Parameternamen der sicheren Zeichenfolge enthält.

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

CloudFormation gibt bei API-Aufrufen nicht den tatsächlichen Parameterwert für sichere Zeichenketten zurück. Sie gibt nur die wörtliche dynamische Referenz zurück. Beim Vergleich von Änderungen mithilfe von Änderungssätzen wird CloudFormation nur die wörtliche dynamische

Referenzzeichenfolge verglichen. Es löst die tatsächlichen sicheren String-Werte nicht auf und vergleicht sie nicht.

Bei der Verwendung von dynamischen `ssm-secure`-Referenzen gibt es einige wichtige Dinge zu beachten:

- CloudFormation kann nicht auf Parameter Store-Werte von anderen AWS-Konten zugreifen.
- CloudFormation unterstützt nicht die Verwendung von Systems Manager Manager-Parameterbeschriftungen oder öffentlichen Parametern in dynamischen Verweisen.
- In den Regionen `cn-north-1` und `cn-northwest-1` werden sichere Zeichenfolgen von Systems Manager nicht unterstützt.
- Dynamische Verweise für 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. Dann verwenden Sie `CONTINUE_UPDATE_ROLLBACK`, ohne die Ressource zu überspringen.

Ressourcen, die dynamische Parametermuster für sichere Zeichenfolgen unterstützen

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

Ressource	Eigenschaftstyp	Eigenschaften
AWS::DirectoryService::MicrosoftAD		Password
AWS::DirectoryService::SimpleAD		Password
AWS::ElastiCache::ReplicationGroup		AuthToken
AWS::IAM::User	LoginProfile	Password

Ressource	Eigenschaftstyp	Eigenschaften
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` zu 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 unter [Arbeiten mit Parameterversionen](#) im AWS Systems Manager -Benutzerhandbuch.

Optional.

Beispiel

Das folgende Beispiel verwendet eine dynamische Referenz `ssm-secure`, um das Passwort für einen IAM-Benutzer auf eine sichere Zeichenkette zu setzen, die im Parameterspeicher 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 ein Geheimnis oder einen geheimen Wert aus dem 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.

Best Practices

Folgen Sie diesen bewährten Methoden, wenn Sie dynamische Secrets Manager Manager-Referenzen in Ihren CloudFormation Vorlagen verwenden:

- Verwenden Sie versionslose Referenzen für Ihre CloudFormation Vorlagen — Speichern Sie Anmeldeinformationen in Secrets Manager und verwenden Sie dynamische Verweise ohne Angabe von `version-stage` `version-id` Parametern, um korrekte Workflows für die geheime Rotation zu unterstützen.
- Nutzen Sie die automatische Rotation- Verwenden Sie das automatische Rotationsfeature von Secrets Manager mit versionslosen dynamischen Referenzen für die Verwaltung von Zugangsdaten. Dadurch wird sichergestellt, dass Ihre Anmeldedaten regelmäßig aktualisiert werden, ohne dass Änderungen an der Vorlage erforderlich sind. Weitere Informationen finden Sie unter [Drehen von AWS Secrets Manager Geheimnissen](#).
- Verwenden Sie versionierte Referenzen sparsam- Geben Sie explizite `version-stage` oder `version-id` Parameter nur für bestimmte Szenarien wie Tests oder Rollback-Situationen an.

Überlegungen

Bei der Verwendung von dynamischen `secretsmanager`-Referenzen sind einige wichtige Punkte zu beachten:

- CloudFormation verfolgt nicht, welche Version eines Secrets in früheren Bereitstellungen verwendet wurde. Planen Sie Ihre Strategie zur Verwaltung von Geheimnissen sorgfältig, bevor Sie dynamische Referenzen einführen. Verwenden Sie nach Möglichkeit versionslose Referenzen, um die automatische Rotation von Geheimnissen zu nutzen. Überwachen und validieren Sie Ressourcenaktualisierungen, wenn Sie Änderungen an dynamischen Referenzkonfigurationen vornehmen, beispielsweise beim Übergang von nicht versionierten zu versionierten dynamischen Referenzen und umgekehrt.
- Wenn nur der geheime Wert in Secrets Manager aktualisiert wird CloudFormation , wird der neue Wert nicht automatisch abgerufen. CloudFormation ruft den geheimen Wert nur bei der

Ressourcenerstellung oder bei Aktualisierungen ab, bei denen die Ressource geändert wird, die die dynamische Referenz enthält.

Nehmen wir zum Beispiel an, Ihre Vorlage enthält eine [AWS::RDS::DBInstance](#)-Ressource, bei der die `MasterPassword`-Eigenschaft auf eine dynamische Referenz von Secrets Manager eingestellt ist. Nachdem Sie einen Stapel aus dieser Vorlage erstellt haben, aktualisieren Sie den Wert des Geheimnisses in Secrets Manager. Die Eigenschaft `MasterPassword` behält jedoch den alten Kennwortwert bei.

Um den neuen geheimen Wert anzuwenden, müssen Sie die `AWS::RDS::DBInstance` Ressource in Ihrer CloudFormation Vorlage ändern und ein Stack-Update durchführen.

Um diesen manuellen Prozess in Zukunft zu vermeiden, sollten Sie den Secrets Manager verwenden, um das Geheimnis automatisch zu drehen.

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

Berechtigungen

Um ein in Secrets Manager gespeichertes Geheimnis anzugeben, müssen Sie die Berechtigung haben, [GetSecretValue](#) für das Geheimnis aufzurufen.

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 `json-key` angeben, 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

- [Abrufen von Benutzernamen und Passwort aus einem Geheimnis](#)
- [Abrufen der gesamten SecretString](#)
- [Abrufen eines Wertes aus einer bestimmten Version eines Geheimnisses](#)
- [Abrufen von Geheimnissen von einem anderen AWS-Konto](#)

Abrufen von Benutzernamen und Passwort aus einem Geheimnis

Das folgende [AWS::RDS::DBInstance](#) Beispiel ruft die im *MySecret*-Geheimnis gespeicherten Werte für Benutzernamen und Kennwort ab. Dieses Beispiel zeigt das empfohlene Muster für versionslose dynamische Referenzen, das automatisch die AWSCURRENT-Version verwendet und Secrets Manager Rotationsworkflows unterstützt, ohne dass Vorlagenänderungen erforderlich sind.

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}}'
```

Abrufen der gesamten SecretString

Die folgende dynamische Referenz ruft die SecretString für *MySecret* ab.

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

Alternative Vorgehensweise:

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

Abrufen eines Wertes aus einer bestimmten Version eines Geheimnisses

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

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

Abrufen von Geheimnissen von einem anderen AWS-Konto

Der folgende dynamische Verweis ruft das SecretString für *MySecret* ab, das in einem anderen AWS-Konto enthalten ist. 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}}
```

Der folgende dynamische Verweis ruft den Wert *password* für *MySecret* ab, der in einem anderen AWS-Konto enthalten ist. 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 Zugriff auf wichtige AWS Umgebungsinformationen wie Konten IDs, Regionsnamen und Stack-Details bieten, 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 Pseudoparameter entweder mit der Funktion Ref intrinsisch oder mit der Funktion Fn::Sub intrinsisch referenzieren.

Punkt

Die Funktion Ref verwendet die folgende allgemeine Syntax. Weitere Informationen finden Sie unter [Ref](#).

JSON

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

YAML

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

Fn::Sub

Die Fn::Sub-Funktion verwendet ein anderes Format, das die \${}-Syntax um den Pseudoparameter herum enthält. 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. Um auf eine bestimmte ARN in der Liste zuzugreifen, verwenden Sie die Funktion `Fn::Select` intrinsic. Weitere Informationen finden Sie unter [Fn::Select](#).

AWS::NoValue

Entfernt die entsprechende Ressourceneigenschaft, wenn als Rückgabewert in der intrinsic Funktion `Fn::If` angegeben. Weitere Informationen finden Sie unter [Fn::If](#).

Dieser Pseudoparameter ist besonders nützlich, um bedingte Ressourceneigenschaften zu erstellen, die nur unter bestimmten Bedingungen einbezogen werden sollen.

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, beispielsweise `arn:aws:cloudformation:us-west-2:123456789012:stack/teststack/51af3dc0-da77-11e4-872e-1234567db123`.

AWS::StackName

Gibt den Namen des Stacks zurück, beispielsweise `teststack`.

Der Stapelname wird üblicherweise verwendet, um eindeutige Ressourcennamen zu erstellen, die leicht als zu einem bestimmten Stapel gehörig identifiziert werden können.

AWS::URLSuffix

Gibt das Suffix für die AWS Domain zurück, AWS-Region in der der Stack bereitgestellt wird. Das Suffix ist normalerweise, `amazonaws.com` aber für die Region China (Beijing) lautet 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 verwenden `AWS::StackName`, und `AWS::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-Gruppe so konfiguriert, dass sie Benachrichtigungen für Instancestart-Ereignisse 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 Funktion `Fn::Select` 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

Die folgenden Beispiele erstellen eine Amazon RDS DB-Instance, die nur dann einen Snapshot verwendet, wenn eine Snapshot-ID angegeben wird. Wenn die UseDBSnapshot Bedingung als wahr ausgewertet wird, wird der DBSnapshotName Parameterwert für die Eigenschaft CloudFormation verwendet. DBSnapshotIdentifier 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 Stack die von einem anderen Stack erstellten Ressourcen verwenden muss.

Sie könnten zum Beispiel einen Stack haben, der Netzwerkressourcen wie Subnetze und Sicherheitsgruppen für Ihre Webserver erstellt. Andere Stacks, die die eigentlichen Webserver erstellen, können dann die vom ersten Stack erstellten Netzwerkressourcen nutzen. Sie müssen die Ressourcen nicht IDs in der Vorlage des Stacks fest codieren oder IDs als Eingabeparameter übergeben.

Um Informationen zwischen Stackn auszutauschen, exportieren Sie Ausgabewerte aus einem Stack und importieren sie in einen anderen Stack. Funktionsweise:

1. In der Vorlage des ersten Stacks (beispielsweise des NetzwerkStacks) definieren Sie bestimmte Werte für den Export, indem Sie das Feld `Export` im Abschnitt `Outputs` verwenden. Weitere Informationen finden Sie unter [CloudFormation OutputsVorlagensyntax](#).
2. Wenn Sie diesen Stapel 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 Funktion [Fn::ImportValue](#), um die exportierten Werte aus dem ersten Stack zu importieren.
4. Wenn Sie den zweiten Stack (z. B. den Webserver-Stack) erstellen oder aktualisieren, CloudFormation werden die exportierten Werte automatisch aus dem ersten Stapel abgerufen und verwendet.

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 anderen 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 Teilnetz verwenden, indem sie seine ID importieren. Jeder Stack muss nicht sein eigenes Subnetz erstellen. Solange Stacks die Subnetz-ID importieren, können sie nicht geändert oder gelöscht werden.

Weitere Informationen über verschachtelte Stapel finden Sie unter [Aufteilung einer Vorlage in wiederverwendbare Teile mit verschachtelten Stapeln](#).

Überlegungen

Die folgenden Einschränkungen gelten für Cross-Stack-Referenzen:

- Jeder `AWS-KontoExport` 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 Stapeln anzeigen möchten, verwenden Sie eine der folgenden Methoden:

So listen Sie exportierte Ausgabewerte auf (Konsole)

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie im linken Navigationsbereich Exporte.

So listen Sie exportierte Ausgabewerte auf (AWS CLI)

Verwenden Sie den folgenden [list-exports](#)-Befehl. Ersetze es durch `us-east-1` 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-075ed3xmplb2fb1"
    },
    {
        "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
        "Name": "private-vpc-vpcid",
        "Value": "vpc-011d7xmpl1100e9841"
    }
]
```

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 Funktion `Fn::ImportValue` referenzieren.

Auflistung von Stacks, die einen exportierten Ausgabewert importieren

Um exportierte Ausgabewerte zu löschen oder zu ändern, müssen Sie zunächst herausfinden, von welchen Stapeln sie importiert werden.

Um die Stapel anzuzeigen, die einen exportierten Ausgabewert importieren, verwenden Sie eine der folgenden Methoden:

Auflisten von Stacks, die einen exportierten Ausgabewert importieren (Konsole)

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie im linken Navigationsbereich Exporte.
3. Um zu sehen, welche Stacks einen bestimmten Exportwert importieren, wählen Sie den Exportnamen für diesen Exportwert. 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 Befehl `list-imports`. `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 Stack einen bestimmten exportierten Wert importieren, müssen Sie diese Stack 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 Parametertypen](#)
- [Nicht unterstützte Systems Manager Parametertypen](#)

-Übersicht

In können Sie Parameter verwenden CloudFormation, um Ihre Stapel anzupassen, indem Sie bei der Erstellung oder Aktualisierung des Stacks Eingabewerte angeben. Dieses Feature macht Ihre Vorlagen wiederverwendbar und flexibel für verschiedene Szenarien.

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 die Art des Eingabewerts, den der Parameter akzeptieren kann. Zum Beispiel akzeptiert `Number` nur numerische Werte, während `String` auch Texteingaben akzeptiert.

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 fallen in zwei Kategorien:

- **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-Managementkonsole, 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 Dropdown-Liste aus. In einigen Fällen können Sie sogar nach Werten über ID, Name oder Name-Tag-Wert 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 über den Parameterspeicher finden Sie unter [Systems Manager Parameterspeicher](#).

Sobald Ihre Parameter im `Parameters` Abschnitt definiert sind, können Sie mithilfe der `Ref` Funktion in Ihrer 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 vorhandene VPC-ID, Subnetz-ID und einen Schlüsselpaar-Namen aus Ihrem Konto angeben. Sie können auch einen vorhandenen Parameterspeicherschlüssel angeben, der auf die gewünschte AMI-ID verweist, oder den Standardwert `/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2` beibehalten. Dieser öffentliche Parameter ist ein Alias für die regionale AMI-ID für das neueste Amazon Linux 2 AMI. Weitere Informationen über öffentliche Parameter finden Sie unter [Entdecken öffentlicher Parameter im Parameterspeicher](#) 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 Befehl [create-stack](#) erstellt einen Stapel auf der Grundlage der Beispielvorlage.

```

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 Zeichenketten akzeptiert, wie beispielsweise `List<AWS::EC2::Subnet::Id>`, müssen Sie die Kommas innerhalb von `ParameterValue` mit einem doppelten Backslash entwerfen, 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 sensible Konfigurationsdefinitionen, wie beispielsweise Zugangsdaten von Drittanbietern, zu beschränken. Weitere Informationen finden Sie unter [Abrufen von in anderen Diensten gespeicherten Werten mit Hilfe von dynamischen Referenzen](#).

Wenn Sie Vorlagenbenutzern die Möglichkeit geben möchten, Werte aus verschiedenen Werten anzugeben AWS-Konten, verwenden Sie keine AWS-spezifischen Parametertypen. Definieren Sie stattdessen Parameter vom Typ `String` oder `CommaDelimitedList`.

Bei den Parametertypen von Systems Manager gibt es einige Dinge zu beachten:

- Sie können die aufgelösten Parameterwerte auf der Registerkarte Parameter des Stacks in der Konsole sehen, oder indem Sie [describe-stacks](#) oder [describe-change-set](#) ausführen. Denken Sie daran, dass diese Werte festgelegt werden, wenn der Stapel erstellt oder aktualisiert wird. Sie können sich also von den aktuellen Werten im Parameter Store unterscheiden.
- Wenn Sie bei Stack-Updates 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 Parameterspeicher selbst validieren.
- CloudFormation Verwendet beim Erstellen oder Aktualisieren von Stacks und beim Erstellen von Änderungssätzen den Wert, der zu diesem Zeitpunkt im Parameterspeicher 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 Parameterspeicher-Parameter, die von einem anderen AWS-Konto gemeinsam genutzt werden, müssen Sie den vollständigen ARN des Parameters 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`

Gibt die ID eines Amazon-EC2-Images an (zum Beispiel `ami-0ff8a91507f77f867`). Beachten Sie, dass die CloudFormation Konsole keine Dropdownliste mit Werten für diesen Parametertyp anzeigt.

`AWS::EC2::Instance::Id`

Gibt die ID einer Amazon EC2-Instance an (zum Beispiel `i-1e731a32`).

`AWS::EC2::KeyPair::KeyName`

Gibt den Namen eines Amazon EC2-Schlüsselpaars an.

`AWS::EC2::SecurityGroup::GroupName`

Ein Standardname der VPC-Sicherheitsgruppe, beispielsweise `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 `Z23YXV40VPL04A`).

`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 Amazon EC2 EC2-Images IDs, z. B. `ami-0ff8a91507f77f867`, `ami-0a584ac55a7631c0c` Beachten Sie, dass die CloudFormation Konsole keine Dropdownliste mit Werten für diesen Parametertyp anzeigt.

`List<AWS::EC2::Instance::Id>`

Ein Array von Amazon EC2 EC2-Instances IDs, z. B. `i-1e731a32`, `i-1e731a34`

`List<AWS::EC2::SecurityGroup::GroupName>`

Ein Array von Standard-VPC-Sicherheitsgruppennamen, z. B. `my-sg-abc`, `my-sg-def`.

`List<AWS::EC2::SecurityGroup::Id>`

Eine Reihe von Sicherheitsgruppen IDs, wie `sg-a123fd85`, `sg-b456fd85` z.

`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. `Z23YXV40VPL04A`, `Z23YXV40VPL04B` B.

Unterstützte Systems 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 Parametertypen

CloudFormation unterstützt den folgenden Systems Manager Manager-Parametertyp nicht:

- Listen von Systems 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 Zeichenketten als Parameter Werte für bestimmte Ressourcen angeben. Weitere Informationen finden Sie unter [Abrufen von in anderen Diensten gespeicherten Werten mit Hilfe von dynamischen Referenzen](#).

CloudFormation Komplettlösungen

Diese Dokumentation enthält eine Sammlung von Komplettlösungen, die Ihnen praktische Übungen für die Stack-Bereitstellung bieten.

- [Verweisen Sie auf Ressourcenausgaben in einem anderen CloudFormation Stack](#)— Diese exemplarische Vorgehensweise zeigt Ihnen, wie Sie auf Ausgaben von einem CloudFormation Stapel innerhalb eines anderen Stacks verweisen. Anstatt alle Ressourcen in einem einzigen Stapel zusammenzufassen, können Sie verwandte AWS Ressourcen in separaten Stacks erstellen, um modularere und wiederverwendbarere Vorlagen zu erstellen.
- [Bereitstellen von Anwendungen auf Amazon EC2](#)— Erfahren Sie, wie CloudFormation Sie Ihre Anwendung auf EC2 Amazon-Instances automatisch installieren, konfigurieren und starten können. Auf diese Weise können Sie problemlos Implementierungen duplizieren und bestehende Installationen aktualisieren, ohne sich direkt mit den Instances zu verbinden.
- [Einen CloudFormation Stack aktualisieren](#)— Gehen Sie eine einfache Abfolge von Updates für einen laufenden Stack mit durch CloudFormation.
- [Erstellen Sie eine skalierte Anwendung mit Load Balancing](#)— Erfahren Sie, wie Sie damit eine skalierbare Anwendung mit Lastenausgleich erstellen können. CloudFormation Diese Anleitung behandelt die Erstellung einer Auto Scaling-Gruppe, eines Load Balancers und anderer zugehöriger Ressourcen, um sicherzustellen, dass Ihre Anwendung unterschiedliche Verkehrslasten bewältigen und eine hohe Verfügbarkeit aufrechterhalten kann.
- [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.
- [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. Dann können Sie auf benötigte Ressourcenausgaben aus anderen Stapeln 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 Netzwerkstapel zu erstellen](#)
- [Verwenden Sie eine Beispielvorlage, um einen Webanwendungsstapel zu erstellen](#)
- [Überprüfen Sie, ob der Stack wie vorgesehen funktioniert](#)
- [Fehler bei der AMI-Zuordnung beheben](#)

- [Bereinigen Ihrer Ressourcen](#)

Verwenden Sie eine Beispielvorlage, um einen Netzwerkstapel 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](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. Im Abschnitt Outputs sehen Sie die Netzwerkressourcen, die die Beispielvorlage exportiert. 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-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie auf der Seite Stapel oben rechts Stapel erstellen und dann Mit neuen Ressourcen (Standard).
3. Wählen Sie Vorhandene Vorlage auswählen, und im Abschnitt Vorlage angeben wählen Sie Amazon S3 URL.
4. Für Amazon S3 URL, fügen Sie 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 [Fortschritt des Stapels überwachen](#).

Verwenden Sie eine Beispielvorlage, um einen Webanwendungsstapel zu erstellen

Der Webanwendungs-Stack erstellt eine EC2-Instance, die die Sicherheitsgruppe und das Subnetz aus dem Netzwerk-Stack 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. Im Abschnitt Resources sehen Sie die Eigenschaften der EC2-Instance. 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 Stack oben rechts Stack erstellen und dann Mit neuen Ressourcen (Standard).
2. Wählen Sie Vorhandene Vorlage auswählen, und im Abschnitt Vorlage angeben wählen Sie Amazon S3 URL.

3. Für Amazon S3 URL, fügen Sie 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 aus.

Die Beispielvorgabe 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.

Es kann mehrere Minuten dauern CloudFormation , bis Ihr Stack erstellt ist.

Überprüfen Sie, ob 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 [Amazon EC2-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](#).

Fehler bei der AMI-Zuordnung beheben

Wenn Sie den Fehler `Template error: Unable to get mapping for AWSRegionArch2AMI::[region]:HVM64` erhalten, enthält die Vorlage keine AMI-Zuordnung für Ihr 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 Abschnitt AWSRegionArch2AMI-Zuordnung.
3. Fügen Sie den folgenden Systems 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 vorhandene Referenz ImageId :

```
"ImageId": { "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" } ,
  "HVM64" ] },
```

mit:

```
"ImageId": { "Ref": "LatestAmiId" },
```

Mit diesem Parameter wird automatisch das neueste Amazon Linux 2 AMI für die Region ausgewählt, in der Sie den Stack bereitstellen.

Für andere Linux-Distributionen verwenden Sie den entsprechenden Parameterpfad. Weitere Informationen finden Sie unter [Ermitteln öffentlicher Parameter im Parameterspeicher](#) 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 bei der Erstellung des Stacks die URL Ihrer S3-Vorlage 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 Sie den Stapel gelöscht haben, wiederholen Sie die gleichen Schritte für den Stapel `SampleNetworkCrossStack`.

Note

Warten Sie, bis der `SampleWebAppCrossStack` Stapel CloudFormation vollständig gelöscht ist. Wenn die EC2-Instance immer noch in der VPC läuft, CloudFormation wird die VPC im Stack nicht gelöscht. `SampleNetworkCrossStack`

Bereitstellen von Anwendungen auf Amazon EC2

Sie können CloudFormation verwenden, um Anwendungen auf Amazon EC2 EC2-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-init`, `cfn-signal`, `cfn-get-metadata`, und `cfn-hup`), die auf `cloud-init` basieren. Sie rufen diese Hilfsskripte aus Ihren CloudFormation Vorlagen auf, um Anwendungen auf Amazon EC2 EC2-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 CloudFormation Template Reference Guide.

Im Tutorial [Erste Schritte](#) haben Sie einen einfachen Webserver mit `UserData` und einem einfachen Bash-Skript erstellt. Während dies für eine einfache "Hello World"-Seite funktionierte, benötigen reale Anwendungen oft eine komplexere Konfiguration, einschließlich:

- Mehrere Softwarepakete in der richtigen Reihenfolge installiert.
- Komplexe Konfigurationsdateien, die mit bestimmten Inhalten erstellt werden.
- Dienste gestartet und für die automatische Ausführung konfiguriert.
- Fehlerbehandlung und Validierung des Einrichtungsprozesses.

CloudFormation Die Hilfsskripte bieten eine robustere und wartungsfreundlichere Methode zur Konfiguration von EC2-Instances im Vergleich zu grundlegenden Bash-Skripten in. UserData Das Hilfsskript `cfn-init` liest Konfigurationsdaten aus den Metadaten Ihrer Vorlage und wendet sie systematisch auf Ihre Instance an.

In diesem Lernprogramm lernen Sie, wie Sie das Hilfsskript `cfn-init` verwenden und den Bootstrapping-Prozess überwachen.

Note

CloudFormation ist kostenlos, aber die von Ihnen erstellten Amazon EC2 EC2-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](#)
- [Verstehen von Bootstrap-Konzepten](#)
- [Beginnen Sie mit einem einfachen Bootstrap-Beispiel](#)
- [Hinzufügen von Dateien und Befehlen](#)
- [Hinzufügen von Netzwerksicherheit](#)
- [Die vollständige Bootstrap-Vorlage](#)
- [Erstellen Sie den Stapel über die Konsole](#)
- [Überwachen Sie den Bootstrap-Prozess](#)
- [Testen Sie den gebootstrapt Webserver](#)
- [Fehlerbehebung bei Bootstrap-Problemen](#)
- [Bereinigen von Ressourcen](#)
- [Nächste Schritte](#)

Voraussetzungen

- Sie müssen das [Erstellen Sie Ihren ersten Stapel](#) 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-Zugriff, der über Berechtigungen zur Verwendung von Amazon EC2 verfügt CloudFormation, und/oder Administratorzugriff.
- Sie benötigen eine Virtual Private Cloud (VPC) mit Zugriff auf das Internet. Diese Übungsvorlage erfordert eine Standard-VPC, die automatisch mit neuer AWS-Konten geliefert wird. Wenn Sie keine Standard-VPC haben oder diese gelöscht wurde, finden Sie im Abschnitt zur Fehlerbehebung im [Erstellen Sie Ihren ersten Stapel](#)-Tutorial alternative Lösungen.

Verstehen von Bootstrap-Konzepten

Bevor wir die Vorlage erstellen, sollten wir die wichtigsten Konzepte verstehen, die das Bootstrapping möglich machen.

Das **cfn-init**-Hilfsskript

CloudFormation stellt Python-Hilfsskripte bereit, mit denen Sie Software installieren und Dienste auf einer Amazon EC2 EC2-Instance starten können. Das Skript `cfn-init` liest die Ressourcen-Metadaten aus Ihrer Vorlage und wendet die Konfiguration auf Ihre Instance an.

Das Verfahren funktioniert folgendermaßen:

1. Sie definieren die Konfiguration im Abschnitt Metadata Ihrer EC2-Ressource.
2. Sie rufen `cfn-init` aus dem Skript UserData auf.
3. `cfn-init` liest die Metadaten und wendet die Konfiguration an.
4. Ihre Instance wird nach Ihren Vorgaben konfiguriert.

Struktur der Metadaten

Die Konfiguration wird in einer bestimmten Struktur innerhalb Ihrer EC2-Instance definiert.

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:  # Metadata section for the resource
        config:                  # Required key that cfn-init looks for
          packages:              # Configuration name (you can have multiple)
          files:                 # Install packages
                                # Create files
```

commands:	# Run commands
services:	# Start/stop services

Das Skript `cfn-init` 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 lediglich Apache 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
            systemd:
              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 veranschaulicht die wichtigsten Konzepte:

- Der Abschnitt `packages` installiert das Paket `httpd` mit `yum`. Dies funktioniert unter Amazon Linux und anderen Linux-Distributionen, die `yum` verwenden.
- Der Abschnitt `services` sorgt dafür, dass `httpd` automatisch gestartet und ausgeführt wird.
- `UserData` installiert die neuesten Bootstrap-Tools und ruft `cfn-init` auf.

Hinzufügen von Dateien und Befehlen

Lassen Sie uns nun unser Beispiel erweitern, indem wir eine benutzerdefinierte Webseite und eine Protokolldatei im Verzeichnis `/var/log` auf der EC2-Instance hinzufügen.

Dateien erstellen

Der Abschnitt `files` ermöglicht es Ihnen, Dateien mit bestimmten Inhalten auf der Instance zu erstellen. Die vertikale Pipe (`|`) ermöglicht es Ihnen, einen wörtlichen Textblock (HTML-Code) als Inhalt der Datei zu übergeben (`/var/www/html/index.html`).

```
files:
  /var/www/html/index.html:
    content: |
      <body>
        <h1>Congratulations, you have successfully launched the AWS CloudFormation
sample.</h1>
      </body>
```

Ausführen von Befehlen

Im Abschnitt `commands` können Sie während des Bootstrap-Prozesses Shell-Befehle ausführen. Dieser Befehl erstellt eine Protokolldatei unter `/var/log/welcome.txt` auf der EC2-Instance. Um sie anzuzeigen, benötigen Sie ein Amazon EC2-Schlüsselpaar für den SSH-Zugang und einen IP-Adressbereich, der für SSH zur Instance verwendet werden kann (hier nicht behandelt).

```
commands:
  createWelcomeLog:
    command: "echo 'cfn-init ran successfully!' > /var/log/welcome.txt"
```

Hinzufügen von Netzwerksicherheit

Da wir einen Webserver einrichten, müssen wir den Webverkehr (HTTP) zu unserer EC2-Instance zulassen. Dazu erstellen wir eine Sicherheitsgruppe, die eingehenden Datenverkehr auf Port 80 von Ihrer IP-Adresse zulässt. EC2-Instances müssen auch Datenverkehr ins Internet senden, zum Beispiel um Paketaktualisierungen zu installieren. Standardmäßig erlauben Sicherheitsgruppen den gesamten ausgehenden Datenverkehr. Anschließend verknüpfen wir diese Sicherheitsgruppe mit unserer EC2-Instance über die Eigenschaft `SecurityGroupIds`.

```
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 vollständige Bootstrap-Vorlage

Lassen Sie uns nun alle Teile zusammenfügen. Hier ist die vollständige Vorlage, die alle besprochenen Konzepte kombiniert.

AWS::TemplateFormatVersion: 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:
```

```
          systemd:
```

```
            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 Stapel über die Konsole

Das folgende Verfahren beinhaltet das Hochladen der Beispiel-Stack-Vorlage aus einer Datei. Öffnen Sie einen Texteditor auf Ihrem lokalen Rechner 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-Managementkonsole und öffnen Sie die CloudFormation Konsole.
2. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
3. Wählen Sie unter Vorlage angeben, Vorlagendatei hochladen, Datei auswählen um die `samplelinux2stack.template` Datei hochzuladen.
4. Wählen Sie Weiter aus.
5. Geben Sie auf der Seite Stack-Details angeben den Stack-Namen **BootstrapTutorialStack** ein.
6. Führen Sie unter Parameterfolgende Schritte aus.
 - LatestAmild: Behalten Sie den Standardwert bei.
 - InstanceType: Wählen Sie entweder t2.micro oder t3.micro für den EC2-Instance-Typ.
 - MeineIP: Geben Sie Ihre öffentliche IP-Adresse mit dem Suffix /32 ein.
7. Wählen Sie zweimal Weiter und dann Senden um den Stapel zu erstellen.

Überwachen Sie den Bootstrap-Prozess

Bootstrap-Prozesse dauern länger als einfache EC2-Starts, da zusätzliche Software installiert und konfiguriert werden muss.

So überwachen Sie den Bootstrap-Fortschritt

1. Wählen Sie in der CloudFormation Konsole Ihren Stack aus und öffnen Sie den Tab Ereignisse.
2. Achten Sie auf das Ereignis `WebServer CREATE_IN_PROGRESS`. Der Bootstrap-Prozess beginnt, nachdem die Instance gestartet wurde.
3. Der Bootstrap-Prozess dauert in der Regel nur wenige Minuten. Sie werden `WebServer CREATE_COMPLETE` sehen, wenn es fertig ist.

Wenn Sie sehen möchten, was während des Bootstrap-Prozesses passiert, können Sie die Instanceprotokolle überprüfen.

So zeigen Sie Bootstrap-Protokolle an (optional)

1. Öffnen Sie die [EC2-Konsole](#) und suchen Sie Ihre Instance.
2. Wählen Sie die Instance aus und wählen Sie dann Aktionen, Überwachen und Fehlerbehebung, Systemprotokoll abrufen um den Bootstrap-Fortschritt zu sehen.
3. Wenn Sie die Protokolle nicht sofort sehen, warten Sie und aktualisieren Sie die Seite.

Testen Sie den gebootstrapten Webserver

Wenn Ihr Stack `CREATE_COMPLETE` anzeigt, testen Sie Ihren Webserver.

So testen Sie den Webserver

1. Gehen Sie in der CloudFormation Konsole zum Tab Outputs für Ihren Stack.
2. Klicken Sie auf den Wert `WebsiteURL` um Ihren Webserver in einer neuen Registerkarte zu öffnen.
3. Sie sollten Ihre benutzerdefinierte Webseite mit der Meldung `Congratulations, you have successfully launched the AWS CloudFormation sample` sehen.

Note

Wenn die Seite nicht sofort geladen wird, warten Sie eine Minute und versuchen Sie es erneut. Der Bootstrap-Prozess wird möglicherweise noch abgeschlossen, auch wenn der Stack `CREATE_COMPLETE` anzeigt.

Fehlerbehebung bei Bootstrap-Problemen

Wenn Ihr Bootstrap-Prozess fehlschlägt oder Ihr Webserver nicht funktioniert, finden Sie hier die häufigsten Probleme und Lösungen.

Häufige Probleme

- Stackerstellung fehlgeschlagen- Überprüfen Sie die Registerkarte Ereignisse auf spezifische Fehlermeldungen.
- Webserver nicht erreichbar- Überprüfen Sie, ob Ihre IP-Adresse im Parameter MyIP korrekt ist. Denken Sie daran, am Ende /32 einzufügen.
- Bootstrap-Prozess schlägt fehl- Die Instance kann starten, aber `cfn-init` schlägt fehl. Überprüfen Sie die Systemprotokolle wie im Abschnitt Überwachung beschrieben.

Bereinigen von Ressourcen

Um laufende Kosten zu vermeiden, können Sie den Stapel und seine Ressourcen löschen.

Löschen des Stacks und der entsprechenden Ressourcen

1. Öffnen Sie die [CloudFormation-Konsole](#).
2. Wählen Sie auf der Seite Stapel die Option neben dem Namen des von Ihnen erstellten Stapels (**BootstrapTutorialStack**) und wählen Sie dann Löschen.
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).
4. Überwachen Sie den Fortschritt des Stapel-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, wie Sie EC2-Instances mit booten. CloudFormation Sie verstehen jetzt:

- So verwenden Sie `cfn-init`-Hilfsskripte
- So strukturieren Sie Metadaten für das Bootstrapping
- Wie Sie Pakete installieren, Dateien erstellen, Befehle ausführen und Dienste verwalten
- Wie Sie Bootstrap-Probleme überwachen

Um weiter zu lernen:

- Erfahren Sie, wie Sie einen laufenden Stack aktualisieren und das `cfn-hup` Hilfsskript verwenden. Weitere Informationen finden Sie unter [Einen CloudFormation Stack aktualisieren](#).
- Lernen Sie, wie man einen Windows-Stack bootet. Weitere Informationen finden Sie unter [Bootstrapping-basierte Stacks Windows CloudFormation](#).
- Erforschen Sie komplexere Bootstrap-Szenarien mit mehreren Konfigurationssätzen. Weitere Informationen finden Sie unter [cfn-init](#) und [AWS::CloudFormation::Init](#) im CloudFormation Template Reference Guide.
- Erfahren Sie mehr über `cfn-signal` für die Meldung des Bootstrap-Abschlussstatus. Weitere Informationen finden Sie unter [cfn-signal](#) im CloudFormation Template Reference Guide.

Einen CloudFormation Stack aktualisieren

Note

Dieses Tutorial baut auf Konzepten aus dem [Bereitstellen von Anwendungen auf Amazon EC2](#) Tutorial auf. Wenn Sie dieses Tutorial noch nicht abgeschlossen haben, empfehlen wir, dies zunächst zu tun, um das EC2 Bootstrapping mit zu verstehen. CloudFormation

In diesem Thema wird ein einfacher Ablauf der Aktualisierungen eines laufenden Stacks veranschaulicht. Wir führen Sie durch die folgenden Schritte:

1. Den ersten Stack erstellen — Erstellen Sie einen Stack mit einem Amazon Linux 2-AMI, installieren Sie das Apache Web Server und eine einfache PHP-Anwendung mithilfe der CloudFormation Hilfsskripte.
2. Anwendung aktualisieren — Aktualisieren Sie eine der Dateien in der Anwendung und stellen Sie die Software mithilfe von bereit CloudFormation.
3. key pair hinzufügen — Fügen Sie der Instance ein EC2 Amazon-Schlüsselpaar hinzu und aktualisieren Sie dann die Sicherheitsgruppe, um SSH-Zugriff auf die Instance zu ermöglichen.
4. Instance-Typ aktualisieren — Ändern Sie den Instance-Typ der zugrunde liegenden EC2 Amazon-Instance.
5. AMI aktualisieren — Ändern Sie das Amazon Machine Image (AMI) für die EC2 Amazon-Instance in Ihrem Stack.

 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

- [Schritt 1: Erstellen Sie den ersten Stack](#)
- [Schritt 2: Aktualisieren Sie die Anwendung](#)
- [Schritt 3: SSH-Zugriff mit einem key pair hinzufügen](#)
- [Schritt 4: Aktualisieren Sie den Instanztyp](#)
- [Schritt 5: AMI aktualisieren](#)
- [Verfügbarkeit und Auswirkungen](#)
- [Zugehörige Ressourcen](#)

Schritt 1: Erstellen Sie den ersten Stack

Wir beginnen mit der Erstellung eines Stacks, den wir im Rest dieses Themas verwenden können. Wir haben eine einfache Vorlage bereitgestellt, mit der eine einzelne Instanz einer PHP-Webanwendung gestartet wird, die auf einem Amazon Linux 2-AMI gehostet wird Apache Web Server und dort ausgeführt wird.

Die Apache Web Server, PHP und die einfache PHP-Anwendung werden alle von den CloudFormation Hilfsskripten installiert, die standardmäßig auf dem Amazon Linux 2-AMI installiert sind. Der folgende Vorlagenausschnitt zeigt die Metadaten, die die zu installierenden Pakete und Dateien beschreiben, in diesem Fall die Apache Web Server und die PHP-Infrastruktur aus dem Yum Repository für das Amazon Linux 2-AMI. Das Snippet zeigt auch den Services Abschnitt, der sicherstellt, dass der läuft. Apache Web Server

```
WebServerInstance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      config:
        packages:
```

```

yum:
  httpd: []
  php: []
files:
  /var/www/html/index.php:
    content: |
      <?php
      echo '<h1>Hello World!</h1>';
      ?>
    mode: '000644'
    owner: apache
    group: apache
services:
  systemd:
    httpd:
      enabled: true
      ensureRunning: true

```

Die Anwendung selbst ist ein Beispiel für „Hello World“, das vollständig in der Vorlage 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 EC2 Amazon-Instance definierten Konfiguration wartet. 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 aus aktualisieren. CloudFormation Der folgende Ausschnitt aus derselben EC2 Amazon-Ressource in der Vorlage zeigt die Teile, die konfiguriert cfn-hup werden müssen, um cfn-init alle zwei Minuten aufzurufen, um Aktualisierungen der Metadaten zu bemerken und anzuwenden. Andernfalls wird der cfn-init nur einmal beim Start ausgeführt.

```

files:
  /etc/cfn/cfn-hup.conf:
    content: !Sub |
      [main]
      stack=${AWS::StackId}
      region=${AWS::Region}
      # The interval used to check for changes to the resource metadata in minutes.
      Default is 15

```

```

    interval=2
    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.WebServerInstance.Metadata.AWS::CloudFormation::Init
    action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --region
    ${AWS::Region}
    runas=root
services:
  systemd:
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf

```

Um den Stack zu vervollständigen, enthält die UserData Eigenschaft im Properties Abschnitt der EC2 Amazon-Instance-Definition das cloud-init Skript, das cfn-init die Installation der Pakete und Dateien aufruft. Weitere Informationen finden Sie in der [Referenz zu CloudFormation Hilfsskripten](#) im CloudFormation Template Reference Guide. Die Vorlage erstellt auch eine EC2 Amazon-Sicherheitsgruppe.

```
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'
```

```
  InstanceType:
```

```
    Description: WebServer EC2 instance type
```

```
    Type: String
```

```
    Default: t3.micro
```

```
    AllowedValues:
```

- t3.nano
- t3.micro
- t3.small

- t3.medium
- t3a.nano
- t3a.micro
- t3a.small
- t3a.medium
- m5.large
- m5.xlarge
- m5.2xlarge
- m5a.large
- m5a.xlarge
- m5a.2xlarge
- c5.large
- c5.xlarge
- c5.2xlarge
- r5.large
- r5.xlarge
- r5.2xlarge
- r5a.large
- r5a.xlarge
- r5a.2xlarge

ConstraintDescription: must be a valid EC2 instance type.

Resources:

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

SecurityGroupIds:

- !Ref WebServerSecurityGroup

UserData:

Fn::Base64: !Sub |

#!/bin/bash -xe

Get the latest CloudFormation package

yum update -y aws-cfn-bootstrap

Run cfn-init

/opt/aws/bin/cfn-init -v --stack \${AWS::StackName} --resource

WebServerInstance --region \${AWS::Region} || 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'

Signal success or failure

```

/opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region}
Metadata:
  AWS::CloudFormation::Init:
    config:
      packages:
        yum:
          httpd: []
          php: []
      files:
        /var/www/html/index.php:
          content: |
            <?php
            echo "<h1>Hello World!</h1>";
            ?>
          mode: '000644'
          owner: apache
          group: apache
        /etc/cfn/cfn-hup.conf:
          content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
            # The interval used to check for changes to the resource metadata in
minutes. Default is 15
            interval=2
            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.WebServerInstance.Metadata.AWS::CloudFormation::Init
            action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}
            runas=root
      services:
        systemd:
          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
    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:
        Value: !Sub 'http://${WebServerInstance.PublicDnsName}'
        Description: URL of the web application
```

Um einen Stack von dieser Vorlage aus zu starten

1. Kopieren Sie die 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 an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
3. Wählen Sie Stack erstellen, Mit neuen Ressourcen (Standard) aus.
4. Wählen Sie „Bestehende Vorlage auswählen“.
5. Wählen Sie unter Vorlage angeben die Option Vorlagendatei hochladen aus, suchen Sie nach der Datei, die Sie im ersten Schritt erstellt haben, und klicken Sie dann auf Weiter.
6. Geben Sie auf der Seite „Stack-Details angeben“ den Namen des Stacks ein **UpdateTutorial**.
7. Behalten Sie unter Parameter alle Parameter bei und wählen Sie zweimal Weiter aus.
8. Wählen Sie auf dem Bildschirm Überprüfen und erstellen die Option Senden aus.

Sobald der Status Ihres Stacks lautet `CREATE_COMPLETE`, wird auf dem Tab „Ausgaben“ die URL Ihrer Website angezeigt. Wenn Sie den Wert der `WebsiteURL` Ausgabe wählen, werden Sie sehen, dass Ihre neue PHP-Anwendung funktioniert.

Schritt 2: Aktualisieren Sie die 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:

```
files:
  /var/www/html/index.php:
    content: |
      <?php
      echo "<h1>Hello World!</h1>";
      echo "<p>This is an updated version of our application.</p>";
      ?>
    mode: '000644'
    owner: apache
    group: apache
```

Verwenden Sie einen Text-Editor, um die Vorlagendatei, die Sie lokal gespeichert haben, zu bearbeiten.

Aktualisieren Sie jetzt den Stack.

Um den Stack mit Ihrer aktualisierten Vorlage zu aktualisieren

1. Wählen Sie in der CloudFormation Konsole Ihren **UpdateTutorial** Stack aus.
2. Wählen Sie Update, Direktes Update durchführen.
3. Wählen Sie Bestehende Vorlage ersetzen.
4. Wählen Sie unter Vorlage angeben die Option Eine Vorlagendatei hochladen und Ihre geänderte Vorlagendatei hochladen aus, und klicken Sie dann auf Weiter.
5. Lassen Sie auf der Seite „Stack-Details angeben“ alle Parameter unverändert und wählen Sie zweimal „Weiter“.
6. Überprüfen Sie auf der Seite Überprüfen die Änderungen. Unter Änderungen sollten Sie sehen, dass die `WebServerInstance` Ressource aktualisiert CloudFormation wird.
7. Wählen Sie Absenden aus.

Wenn sich Ihr Stack im `UPDATE_COMPLETE` Status befindet, können Sie den `WebsiteURL` Ausgabewert erneut auswählen, um zu überprüfen, ob die Änderungen an Ihrer Anwendung wirksam wurden. Der `cfn-hup` Daemon wird alle 2 Minuten ausgeführt, sodass es bis zu 2 Minuten dauern kann, bis sich die Anwendung ändert, sobald der Stack aktualisiert wurde.

Rufen Sie die CloudFormation Konsole auf, um zu sehen, welche Ressourcen aktualisiert wurden. 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 Stapelressourcen wurde modifiziert. CloudFormation aktualisiert nur die Ressourcen im Stapel, die von Änderungen am Stapel betroffen sind. Solche Änderungen können direkt sein, z. B. Änderungen an Eigenschaften oder Metadaten, oder sie können auf Abhängigkeiten oder Datenflüsse durch `RefGetAtt`, oder andere systeminterne Vorlagenfunktionen zurückzuführen sein. Weitere Informationen finden Sie unter [Intrinsische Funktionsreferenz](#).

Dieses einfache Update veranschaulicht den Prozess. 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.

```
packages:
  yum:
    httpd: []
    php: []
    mysql: []
    php-mysql: []
    mysql-server: []
    mysql-libs: []

    ...

services:
  systemd:
    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
```

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 für jedes Paket leer, was bedeutet, dass die neueste Version des Pakets installiert werden soll.

```
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"
```

Schritt 3: SSH-Zugriff mit einem key pair hinzufügen

Sie können auch eine Ressource in der Vorlage aktualisieren, um Eigenschaften hinzuzufügen, die ursprünglich nicht in der Vorlage angegeben wurden. 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.

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 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 that can be used to SSH to the EC2 instances 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.

2. Fügen Sie die KeyName Eigenschaft zur EC2 Amazon-Instance hinzu.

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

KeyName: !Ref KeyName

SecurityGroupIds:

- !Ref WebServerSecurityGroup

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 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
```

4. Aktualisieren Sie den Stack mit den gleichen Schritten wie unter beschrieben. [Schritt 2: Aktualisieren Sie die Anwendung](#)

Schritt 4: Aktualisieren Sie den Instanztyp

Lassen Sie uns nun demonstrieren, wie Sie die zugrunde liegende Infrastruktur aktualisieren, indem Sie den Instanztyp ändern.

Der Stack, den wir bisher erstellt haben, verwendet eine EC2 t3.micro-Amazon-Instance. Nehmen wir an, dass Ihre neu erstellte Website mehr Traffic erhält, als eine t3.micro-Instance verarbeiten kann, und Sie möchten jetzt zu einem Amazon-Instance-Typ m5.large wechseln. EC2 Wenn sich die Architektur des Instance-Typs ändert, muss die Instance mit einem anderen AMI erstellt werden. Sowohl t3.micro als auch m5.large verwenden jedoch dieselben CPU-Architekturen und laufen unter Amazon Linux 2 (x86_64). AMIs Weitere Informationen finden Sie unter [Kompatibilität bei der Änderung des Instance-Typs](#) im EC2 Amazon-Benutzerhandbuch.

Lassen Sie uns die Vorlage verwenden, 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 auf der Seite „Stack-Details angeben“ ändern.

Um den Stack mit einem neuen Parameterwert zu aktualisieren

1. Wählen Sie in der CloudFormation Konsole Ihren **UpdateTutorial** Stack aus.
2. Wählen Sie Update, Direktes Update durchführen.
3. Wählen Sie „Bestehende Vorlage verwenden“ und anschließend „Weiter“.
4. Ändern Sie auf der Seite „Stack-Details angeben“ den Wert des InstanceTypeTextfeldes von t3.micro zum5.large. Wählen Sie dann zweimal Weiter aus.

5. Überprüfen Sie auf der Seite Überprüfen die Änderungen. Unter Änderungen sollten Sie sehen, dass die `WebServerInstance` Ressource aktualisiert CloudFormation wird.
6. Wählen Sie Absenden 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-Managementkonsole, öffnen Sie die EC2 Amazon-Konsole und suchen Sie dort nach Ihrer Instance.

Schritt 5: AMI aktualisieren

Lassen Sie uns nun unseren Stack aktualisieren, um Amazon Linux 2023 zu verwenden, die nächste Generation von Amazon Linux.

Die Aktualisierung des AMI ist eine wichtige Änderung, die den Austausch der Instance erfordert. Wir können die Instance nicht einfach starten und stoppen, um das AMI zu ändern; CloudFormation betrachtet dies als Änderung 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.

Schauen wir uns an, wie wir unsere Stack-Vorlage aktualisieren könnten, um Amazon Linux 2023 zu verwenden. Zu den wichtigsten Änderungen gehören die Aktualisierung des AMI-Parameters und der Wechsel vom Paketmanager `yum` zum `dnf` Paketmanager.

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

```
Parameters:
```

```
  LatestAmiId:
```

```
    Description: The latest Amazon Linux 2023 AMI from the Parameter Store
```

```
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
```

```
    Default: '/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64'
```

```
  InstanceType:
```

```
    Description: WebServer EC2 instance type
```

Type: String
Default: t3.micro
AllowedValues:

- t3.nano
- t3.micro
- t3.small
- t3.medium
- t3a.nano
- t3a.micro
- t3a.small
- t3a.medium
- m5.large
- m5.xlarge
- m5.2xlarge
- m5a.large
- m5a.xlarge
- m5a.2xlarge
- c5.large
- c5.xlarge
- c5.2xlarge
- r5.large
- r5.xlarge
- r5.2xlarge
- r5a.large
- r5a.xlarge
- r5a.2xlarge

ConstraintDescription: must be a valid EC2 instance type.

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 that can be used to SSH to the EC2 instances 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:

```

WebServerInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref LatestAmiId
    InstanceType: !Ref InstanceType
    KeyName: !Ref KeyName
    SecurityGroupIds:
      - !Ref WebServerSecurityGroup
    UserData:
      Fn::Base64: !Sub |
        #!/bin/bash -xe
        # Get the latest CloudFormation package
        dnf update -y aws-cfn-bootstrap
        # Run cfn-init
        /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region} || 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'
        # Signal success or failure
        /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region}
  Metadata:
    AWS::CloudFormation::Init:
      config:
        packages:
          dnf:
            httpd: []
            php: []
        files:
          /var/www/html/index.php:
            content: |
              <?php
              echo "<h1>Hello World!</h1>";
              echo "<p>This is an updated version of our application.</p>";
              echo "<p>Running on Amazon Linux 2023!</p>";
              ?>
            mode: '000644'
            owner: apache
            group: apache
          /etc/cfn/cfn-hup.conf:
            content: !Sub |
              [main]

```



```

        stack=${AWS::StackId}
        region=${AWS::Region}
        # The interval used to check for changes to the resource metadata in
minutes. Default is 15
        interval=2
        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.WebServerInstance.Metadata.AWS::CloudFormation::Init
            action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}
        runas=root
services:
    systemd:
    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
CreationPolicy:
    ResourceSignal:
    Timeout: PT5M

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

```

Outputs:**WebsiteURL:**

Value: !Sub 'http://\${WebServerInstance.PublicDnsName}'

Description: URL of the web application

Aktualisieren Sie den Stack mit den gleichen Schritten wie unter beschrieben [Schritt 2: Aktualisieren Sie die Anwendung](#).

Sobald die neue Instanz ausgeführt wird, werden die anderen Ressourcen im Stack so CloudFormation aktualisiert, dass sie auf die neue Ressource verweisen. Wenn alle neuen Ressourcen erstellt sind, wird die alte Ressource gelöscht, ein Vorgang, der als UPDATE_CLEANUP bekannt ist. 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 Ereignistabelle enthalten die Beschreibung „Die angeforderte Aktualisierung hat eine Änderung an einer unveränderlichen Eigenschaft zur Folge und erzeugt somit eine neue physische Ressource“, die darauf hinweist, dass eine Ressource ersetzt wurde.

Alternative: Wenn Sie Anwendungscode in das AMI geschrieben haben, den Sie aktualisieren möchten, können Sie denselben Stack-Aktualisierungsmechanismus verwenden, um das AMI zu aktualisieren, um Ihre neue Anwendung zu laden.

Um das AMI mit benutzerdefiniertem Anwendungscode zu aktualisieren

1. Erstellen Sie Ihr neues AMI, das 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 die neue AMI-ID zu integrieren.
3. Aktualisieren Sie den Stack mit den gleichen Schritten wie unter beschrieben [Schritt 2: Aktualisieren Sie die Anwendung](#).

Wenn Sie den Stack aktualisieren, wird CloudFormation erkannt, dass sich die AMI-ID geändert hat, und löst dann ein Stack-Update auf dieselbe Weise aus, wie wir das obige initiiert haben.

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 beispielsweise der Alarm während der Aktualisierung inaktiv. Wie wir gesehen haben, erfordert das Ändern 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 mit CloudFormation Opscode Chef](#)
- [Integration CloudFormation mit Puppet](#)

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 Amazon-EC2-Instances, sowie 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>\n" > /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 CloudFormation , damit der Stack erfolgreich bereitgestellt werden kann. Später in der Vorlage angegebene Ressourcen verweisen auf diese Werte und verwenden die Daten.

- **InstanceType**: Der Typ der EC2-Instance, die Amazon EC2 Auto Scaling bereitstellt. Wenn Sie keinen anderen Wert angeben, wird der Standardwert von `t3.micro` verwendet.
- **KeyName**: Ein vorhandenes EC2-Schlüsselpaar, um SSH-Zugriff auf die Instances zu ermöglichen.
- **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::SecurityGroup](#)Ressource 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::SecurityGroup](#)Ressource 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 [GetAtt](#)Funktion wird verwendet, um die ID der Sicherheitsgruppe mit dem logischen Namen `ELBSecurityGroup` abzurufen.

[AWS::ElasticLoadBalancingV2::TargetGroup](#)Ressource EC2TargetGroup

- `Port`, `Protocol`, und `HealthCheckProtocol` geben den Port (80) und das Protokoll (HTTP) der EC2-Instance an, an die `ApplicationLoadBalancer` den Datenverkehr weitergeleitet und die Elastic Load Balancing verwendet, um den Zustand der EC2-Instances zu überprüfen.
- `HealthCheckIntervalSeconds` gibt an, dass die EC2-Instances einen Abstand von 30 Sekunden zwischen den Zustandsprüfungen 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 der Zeitüberschreitung markiert Elastic Load Balancing die Zustandsprüfung der EC2-Instance als instabil. Wenn eine EC2-Instance drei aufeinanderfolgende Zustandsprüfungen (`UnhealthyThresholdCount`) nicht besteht, stoppt Elastic Load Balancing das Weiterleiten von Datenverkehr an diese EC2-Instance, bis diese Instance fünf aufeinanderfolgende fehlerfreie Zustandsprüfungen (`HealthyThresholdCount`) 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::Listener](#)Ressource 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::LoadBalancer](#)Ressource 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 eingehenden Datenverkehr zu steuern. Die [GetAtt](#)Funktion wird verwendet, um die ID der Sicherheitsgruppe mit dem logischen Namen abzurufen `ELBSecurityGroup`.

[AWS::EC2::LaunchTemplate](#)Ressource 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-Schlüsselpaar.
- `SecurityGroupIds` ruft die ID der Sicherheitsgruppe mit dem logischen Namen `EC2SecurityGroup` ab, die als virtuelle Firewall für Ihre EC2-Instances fungiert, um den eingehenden Datenverkehr zu kontrollieren.
- `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::Topic](#)Ressource NotificationTopic

- `Subscription` verwendet den Wert des `OperatorEmail`-Eingabeparameters als E-Mail-Adresse für den Empfänger der Benachrichtigungen, wenn Skalierungsaktivitäten stattfinden.

`AWS::AutoScaling::AutoScaling`Ressource [gruppieren](#) `WebServerGroup`

- `MinSize` und `MaxSize` bestimmen die Mindest- und Höchstanzahl von EC2-Instances in der Auto-Scaling-Gruppe.

- `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 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-Managementkonsole und öffnen Sie die Konsole unter / cloudformation CloudFormation . https://console.aws.amazon.com](https://console.aws.amazon.com/cloudformation)
2. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
3. Wählen Sie unter Vorlage angeben, Vorlagendatei hochladen, Datei auswählen um die `sampleloadbalancedappstack.template` Datei 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 CloudFormation Konsole in der Spalte Status einsehen. Wenn 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 erhalten kann. Weitere Informationen finden Sie unter [Erhalten Sie Amazon-SNS-Benachrichtigungen, wenn Ihre Auto-Scaling-Gruppe skaliert](#) im Benutzerhandbuch für Amazon EC2 Auto Scaling.

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 CloudFormation Konsole den Stack aus. `SampleLoadBalancedAppStack`
2. Wählen Sie Löschen aus.
3. Wählen Sie in der Bestätigungsmitteilung die Option Stack löschen aus.

Der Status für `SampleLoadBalancedAppStack` Änderungen an `DELETE_IN_PROGRESS`. Wenn das Löschen des Stacks CloudFormation abgeschlossen ist, wird der Stapel aus der Liste entfernt.

Verwenden Sie diese Beispieldvorlage aus dieser Anleitung, um eigene Stacks zu erstellen. Weitere Informationen finden Sie unter [Tutorial: Einrichten einer skalierten Anwendung mit Load Balancing](#) im Benutzerhandbuch zu Amazon EC2 Auto Scaling.

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 über VPC-Peering und dessen 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 kontenübergreifenden Rolle erteilt, finden Sie unter [Einem Benutzer Berechtigungen zum Rollenwechsel erteilen](#) im IAM-Benutzerhandbuch.

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-Managementkonsole und öffnen Sie die CloudFormation Konsole. https://console.aws.amazon.com](#)
2. Wählen Sie auf der Seite Stack oben rechts Stack erstellen und dann Mit neuen Ressourcen (Standard).
3. Für Voraussetzung - Vorlage vorbereiten, wählen Sie Vorhandene Vorlage auswählen und dann Vorlagendatei hochladen, Datei auswählen.

4. Öffnen Sie einen Texteditor auf Ihrem lokalen Rechner 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 Feld ein. PeerRequesterAccountId
7. Akzeptieren Sie die Standardwerte und wählen Sie dann Weiter aus.
8. Wählen Sie Ich bestätige, dass 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 [AWS::EC2::VPCPeeringVerbindungsressource](#) enthält

1. Gehen Sie zurück zur Startseite der CloudFormation Konsole.
2. Wählen Sie auf der Seite Stack oben rechts Stack erstellen und dann Mit neuen Ressourcen (Standard).
3. Für Voraussetzung - Vorlage vorbereiten, wählen Sie Vorhandene Vorlage auswählen und dann Vorlagendatei hochladen, Datei auswählen.
4. Öffnen Sie einen Texteditor auf Ihrem lokalen Rechner 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 CloudFormation möglicherweise IAM-Ressourcen erstellt werden, und wählen Sie dann Stack 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}"
                }
              }
            ],
            "Condition":{
              "StringEquals":{
                "ec2:AcceptorVpc":{

```

```

        "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/${vpc}'
            - Action: 'ec2:acceptVpcPeeringConnection'
              Condition:
                StringEquals:
                  'ec2:AcceptorVpc':
                    'Fn::Sub': 'arn:aws:ec2:${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 Zugangsverwaltung für VPC-Peering](#) im Amazon VPC Peering Guide.

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 durch wechselnde Anwendungsversionen zu minimieren.

In einer blue/green Bereitstellung erstellen Sie neben Ihrer aktuellen Live-Umgebung (als blau bezeichnet) eine neue Anwendungsumgebung (als grün bezeichnet). So können Sie die grüne Umgebung überwachen und testen, bevor Sie den Live-Datenverkehr von der blauen Umgebung in die grüne Umgebung leiten. Nachdem die grüne Umgebung den Live-Verkehr bedient hat, können Sie die blaue Umgebung sicher beenden.

Um Bereitstellungen in CodeDeploy Blau/Grün auf ECS durchzuführen CloudFormation, nehmen Sie die folgenden Informationen in Ihre Stack-Vorlage auf:

- Ein Hooks-Abschnitt, der einen `AWS::CodeDeploy::BlueGreen-Hook` beschreibt.
- Ein Transform-Abschnitt, der die `AWS::CodeDeployBlueGreen-Transformation` angibt.

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 blue/green ECS-Bereitstellungen mit CloudFormation](#)
- [AWS::CodeDeploy::BlueGreen-Hook-Syntax](#)
- [Beispiel für eine blau/grüne Einsatzvorlage](#)

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](#)
- [Ressourcenaktualisierungen, die grüne Bereitstellungen initiieren](#)
- [Bereiten Sie Ihre Vorlage für die Durchführung von blue/green ECS-Bereitstellungen vor](#)
- [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 Stapel 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.

Ressourcenaktualisierungen, die grüne Bereitstellungen initiieren

Wenn Sie ein Stack-Update durchführen, das bestimmte Eigenschaften bestimmter ECS-Ressourcen aktualisiert, wird ein CloudFormation umweltfreundlicher Bereitstellungsprozess eingeleitet. Die Ressourcen, die diesen Prozess in Gang setzen, sind:

- [AWS::ECS::TaskDefinition](#)
- [AWS::ECS::TaskSet](#)

Wenn die Aktualisierungen dieser Ressourcen jedoch keine Eigenschaftsänderungen beinhalten, die eine Ersetzung erfordern, wird keine grüne Bereitstellung initiiert. Weitere Informationen finden Sie unter [Verstehen des Aktualisierungsverhaltens von Stack-Ressourcen](#).

Es ist wichtig zu wissen, dass Sie Aktualisierungen der oben genannten Ressourcen nicht mit Aktualisierungen anderer Ressourcen im selben Stapelaktualisierungsvorgang kombinieren können. Wenn Sie sowohl die aufgeführten Ressourcen als auch andere Ressourcen innerhalb desselben Stapels aktualisieren müssen, haben Sie zwei Möglichkeiten:

- Führen Sie zwei separate Stack-Aktualisierungsvorgänge durch: einen, der 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.

Bereiten Sie Ihre Vorlage für die Durchführung von blue/green ECS-Bereitstellungen vor

Um blue/green Bereitstellungen auf Ihrem Stack zu ermöglichen, 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::BlueGreen-Hook-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	Nein

Anforderung	Ressource	Erforderlich/optional	Initiiert die blue/green Bereitstellung, falls sie ersetzt wird?
		kann verwendet werden.	
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. So können Sie die tatsächlichen Änderungen, die an Ihrem Stack vorgenommen werden, sehen, bevor Sie die Stack-Aktualisierung 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 [Fortschritt des Stapels 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 blue/green ECS-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 CloudFormation](#) im AWS CodeDeploy Benutzerhandbuch.

Bei der Verwaltung Ihrer blue/green Bereitstellung mithilfe von CloudFormation sind bestimmte Einschränkungen und Überlegungen zu beachten:

- Nur bei Aktualisierungen bestimmter Ressourcen wird eine grüne Bereitstellung initiiert. Weitere Informationen finden Sie unter [Ressourcenaktualisierungen, die grüne Bereitstellungen initiieren](#).
- Sie können keine Aktualisierungen für Ressourcen, die grüne Bereitstellungen initiieren, und Aktualisierungen für andere Ressourcen in dieselbe Stack-Aktualisierung aufnehmen. Weitere Informationen finden Sie unter [Ressourcenaktualisierungen, die grüne Bereitstellungen initiieren](#).
- Sie können nur einen einzelnen ECS-Service als Bereitstellungsziel angeben.

- Parameter, deren Werte verschleiert werden, 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 Informationen über dynamische Referenzen finden Sie unter [Abrufen von in anderen Diensten gespeicherten Werten mit Hilfe von dynamischen Referenzen](#).
- Um eine grüne Bereitstellung abubrechen, die noch im Gange ist, brechen Sie das Stack-Update in CloudFormation, Not CodeDeploy oder ECS ab. Weitere Informationen finden Sie unter [Eine Stapelaktualisierung abbrechen](#). Nach Abschluss einer Aktualisierung können Sie diese nicht mehr stornieren. Sie können jedoch einen Stapel mit allen vorherigen Einstellungen erneut aktualisieren.
- Die folgenden CloudFormation Funktionen werden derzeit nicht für Vorlagen unterstützt, die blue/green ECS-Bereitstellungen definieren:
 - [Ausgaben](#) deklarieren oder [Fn::](#) verwenden `ImportValue`, um Werte aus anderen Stacks zu importieren.
 - Ressourcen importieren. Weitere Informationen zum Importieren von Ressourcen finden Sie unter [AWS Ressourcen in einen CloudFormation Stack importieren](#).
 - Verwendung des `AWS::CodeDeploy::BlueGreenHooks` in einer Vorlage, die verschachtelte Stackressourcen enthält. Weitere Informationen über verschachtelte Stapel finden Sie unter [Aufteilung einer Vorlage in wiederverwendbare Teile mit verschachtelten Stapeln](#).
 - Verwendung des Hooks `AWS::CodeDeploy::BlueGreen` in einem verschachtelten Stack.

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 Abschnitt Hooks der Vorlage deklariert ist. 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 zur Ausführung des Docker-Containers, der Ihre Amazon ECS-Anwendung enthält.

Erforderlich: Ja

TaskSets

Die logischen IDs [AWS::ECS::TaskSet](#) Ressourcen, 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 blau/grüne Einsatzvorlage

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 das Erstellen eines Stacks mit der Vorlage wird die anfängliche Konfiguration 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:
  HttpStatusCode: '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 CloudFormation -Vorlagenteile deklariert werden. Sie können die Ausschnitte auch als Ausgangspunkt für Abschnitte von benutzerdefinierten Vorlagen verwenden.

Note

Da CloudFormation Vorlagen JSON-konform sein müssen, ist kein Zeilenfortsetzungszeichen vorgesehen. Der Umbruch der Schnipsel in diesem Dokument kann zufällig sein, wenn die Zeile länger als 68 Zeichen ist.

Themen

- [Allgemeine Vorlagenausschnitte](#)
- [CloudFormation Vorlagenausschnitte für automatische Skalierung](#)
- [AWS Vorlagenausschnitte für die Rechnungskonsole](#)
- [CloudFormation Schnipsel aus Vorlagen](#)
- [CloudFront Amazon-Vorlagenschnipsel](#)
- [CloudWatch Amazon-Vorlagenschnipsel](#)
- [Amazon CloudWatch Logs-Vorlagenausschnitte](#)
- [Amazon DynamoDB-Vorlagenausschnitte](#)
- [EC2 CloudFormation Amazon-Vorlagenschnipsel](#)
- [Amazon Elastic Container Service Beispielvorlagen](#)
- [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 Vorlage Schnipsel](#)

Allgemeine Vorlagenausschnitte

Die folgenden Beispiele zeigen verschiedene CloudFormation Vorlagenfunktionen, die nicht spezifisch für einen AWS Dienst sind.

Themen

- [Base64 encoded UserData property](#)
- [Base64 encoded UserData property with AccessKey and SecretKey](#)
- [Parameters-Abschnitt mit einem literalen String-Parameter](#)
- [Parameters-Abschnitt mit String-Parameter mit Beschränkung auf reguläre Ausdrücke](#)
- [Parameters-Abschnitt mit Zahlenparameter mit MinValue und MaxValue-Beschränkungen](#)
- [ParametersAbschnitt mit Zahlenparameter mit AllowedValues Einschränkung](#)
- [Parameters-Abschnitt mit einem literalen CommaDelimitedList-Parameter](#)
- [Parameters-Abschnitt mit Parameterwert basierend auf Pseudoparameter](#)
- [AbschnittMapping mit drei Zuordnungen](#)
- [Description basierend auf einer wörtlichen Zeichenkette](#)
- [Outputs-Abschnitt mit einer literalen Stringausgabe](#)
- [Outputs-Abschnitt mit einer Ressourcenreferenz und einer Pseudoreferenzausgabe](#)
- [Outputs-Abschnitt mit einer Ausgabe, die auf einer Funktion, einem literalen String, einer Referenz und einem Pseudoparameter basiert](#)
- [Vorlagenformatversion](#)
- [AWS Tags Eigenschaft](#)

Base64 encoded UserData property

Dieses Beispiel zeigt den Aufbau einer Eigenschaft `UserData` unter Verwendung der Funktionen `Fn::Base64` und `Fn::Join`. Die Referenzen `MyValue` und `MyName` sind Parameter, die im

Abschnitt `Parameters` 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
```

Base64 encoded UserData property with AccessKey and SecretKey

Dieses Beispiel zeigt den Aufbau einer Eigenschaft `UserData` unter Verwendung der Funktionen `Fn::Base64` und `Fn::Join`. 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}
```

Parameters-Abschnitt mit einem literalen String-Parameter

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
```

Parameters-Abschnitt mit String-Parameter mit Beschränkung auf reguläre Ausdrücke

Das folgende Beispiel beschreibt eine gültige Deklaration für den Abschnitt „Parameters“, in welcher ein einzelner Parameter vom Typ `String` deklariert wird. Der Parameter `AdminUserAccount` 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]*'
```

Parameters-Abschnitt mit Zahlenparameter mit MinValue und MaxValue-Beschränkungen

Das folgende Beispiel beschreibt eine gültige Deklaration für den Abschnitt „Parameters“, in welcher ein einzelner Parameter vom Typ Number deklariert wird. Der Parameter WebServerPort hat einen Standardwert von 80, einen Mindestwert 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 Zahlenparameter mit AllowedValues Einschränkung

Das folgende Beispiel beschreibt eine gültige Deklaration für den Abschnitt „Parameters“, in welcher ein einzelner Parameter vom Typ Number deklariert wird. Der Parameter WebServerPort hat den Standardwert 80 und erlaubt nur Werte zwischen 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
```

Parameters-Abschnitt mit einem literalen CommaDelimitedList-Parameter

Das folgende Beispiel zeigt eine gültige Parameters-Abschnittsdeklaration, in der ein einzelner CommaDelimitedList-Typ-Parameter deklariert ist. Die Eigenschaft NoEcho wird auf TRUE gesetzt,

wodurch ihr Wert in der describe-stacks-Ausgabe mit Sternchen (*****) maskiert wird, mit Ausnahme der Informationen, die an den unten angegebenen Stellen 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
```

Parameters-Abschnitt mit Parameterwert basierend auf Pseudoparameter

Das folgende Beispiel zeigt Befehle in den EC2-Benutzerdaten, welche die Pseudoparameter `AWS::StackName` und `AWS::Region` verwenden. 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}
```

AbschnittMapping mit drei Zuordnungen

Das folgende Beispiel zeigt eine gültige Mapping-Abschnittserklärung, 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"
```

Description basierend auf einer wörtlichen Zeichenkette

Das folgende Beispiel zeigt eine gültige `Description`-Abschnittsdeklaration, bei der der Wert auf einer literalen Zeichenkette basiert. Dieses Codefragment kann für Vorlagen, Parameter, Ressourcen, Eigenschaften oder Ausgaben verwendet werden.

JSON

```
"Description" : "Replace this value"
```

YAML

```
Description: "Replace this value"
```

Outputs-Abschnitt mit einer literalen Stringausgabe

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
```

Outputs-Abschnitt mit einer Ressourcenreferenz und einer Pseudoreferenzausgabe

Dieses Beispiel zeigt einen `Outputs`-Abschnitt mit zwei Ausgangszuweisungen. 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
```

Outputs-Abschnitt mit einer Ausgabe, die auf einer Funktion, einem literalen String, einer Referenz und einem Pseudoparameter basiert

Dieses Beispiel zeigt einen Outputs-Abschnitt mit einer Ausgabenzuweisung. Die Join-Funktion wird verwendet, um den Wert zu verketteten, 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 `AWSTemplateFormatVersion`-Abschnittsdeklaration.

JSON

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

YAML

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

AWS Tags Eigenschaft

Dieses Beispiel zeigt eine AWS Tags Eigenschaft. 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 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 Verwaltung und Automatisierung der Bereitstellung von automatisch skalierenden Ressourcen auf wiederholbare und konsistente Weise.

Die folgenden Beispielvorlagen beschreiben 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 CloudFormation](#)
- [Konfigurieren Sie Auto Scaling-Ressourcen für Anwendungen mit CloudFormation](#)

Konfigurieren Sie Amazon EC2 Auto Scaling Scaling-Ressourcen mit CloudFormation

Die folgenden Beispiele zeigen verschiedene Snippets, die in Vorlagen für die Verwendung mit Amazon EC2 Auto Scaling aufgenommen werden können.

Kategorien von Ausschnitten

- [Erstellen Sie eine Auto Scaling-Gruppe für eine einzelne Instance](#)
- [Erstellen Sie eine Auto Scaling-Gruppe mit einem angeschlossenen Load Balancer](#)
- [Erstellen Sie eine Auto-Scaling-Gruppe mit Benachrichtigungen](#)
- [Erstellen Sie eine Autoskalierungsgruppe, die ein CreationPolicy und ein UpdatePolicy verwendet.](#)
- [Erstellen Sie eine Skalierungsrichtlinie](#)
- [Gemischte Instances-Gruppen](#)

- [Beispiele für Launch-Konfigurationen](#)

Erstellen Sie eine Auto Scaling-Gruppe für eine einzelne Instance

Dieses Beispiel zeigt eine [AWS::AutoScaling::AutoScalingGroup](#)-Ressource mit einer einzelnen Instance, um Ihnen den Einstieg zu erleichtern. Die `VPCZoneIdentifizier`-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 Eigenschaft `LaunchTemplate` verweist auf eine [AWS::EC2::LaunchTemplate](#)-Ressource mit dem logischen Namen `myLaunchTemplate`, die an anderer Stelle in Ihrer Vorlage definiert ist.

 Note

Beispiele für Startvorlagen finden Sie unter [Erstellen Sie Startvorlagen mit CloudFormation](#) im Abschnitt Amazon EC2 Snippets und im Abschnitt [Beispiele](#) in der `AWS::EC2::LaunchTemplate` Ressource.

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifizier" : [ "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-Gruppe mit einem angeschlossenen Load Balancer

Dieses Beispiel zeigt eine [AWS::AutoScaling::AutoScalingGroup](#)-Ressource für den Lastausgleich über mehrere Server. Es gibt die logischen Namen von AWS Ressourcen an, die an anderer Stelle in derselben Vorlage deklariert wurden.

1. Die Eigenschaft `VPCZoneIdentifier` gibt die logischen Namen von zwei [AWS::EC2::Subnet](#)-Ressourcen an, in denen die EC2-Instances der Auto Scaling-Gruppe erstellt werden: `myPublicSubnet1` und `myPublicSubnet2`.
2. Die Eigenschaft `LaunchTemplate` gibt eine [AWS::EC2::LaunchTemplate](#) Ressource mit dem logischen Namen `myLaunchTemplate` an.
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-Gruppe mit Benachrichtigungen

Dieses Beispiel zeigt eine [AWS::AutoScaling::AutoScalingGroup](#)-Ressource, die Amazon SNS-Benachrichtigungen sendet, wenn die angegebenen Ereignisse eintreten. Die `NotificationConfigurations` Eigenschaft gibt das SNS-Thema an, an das eine Benachrichtigung CloudFormation gesendet wird, und die Ereignisse, die CloudFormation dazu führen, dass Benachrichtigungen gesendet werden. Wenn die von angegebenen Ereignisse `NotificationTypes` eintreten, CloudFormation wird eine Benachrichtigung an das von angegebene SNS-Thema gesendet. `TopicARN` Wenn Sie den Stack starten, 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 Eigenschaft `LaunchTemplate` verweist auf den logischen Namen einer [AWS::EC2::LaunchTemplate](#)-Ressource, die an anderer Stelle in derselben Vorlage deklariert ist.

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 Autoskalierungsgruppe, die ein **CreationPolicy** und ein **UpdatePolicy** verwendet.

Das folgende Beispiel zeigt, wie Sie die Attribute `CreationPolicy` und `UpdatePolicy` zu einer [AWS::AutoScaling::AutoScalingGroup](#)-Ressource hinzufügen.

Die Richtlinie zur Probenerstellung verhindert, dass die Auto Scaling Scaling-Gruppe `CREATE_COMPLETE` den Status erreicht, bis sie 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 `sendTimeout`, CloudFormation wird davon ausgegangen, dass die Instanzen 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 die Stack-Aktualisierung ausgeführt wird, werden die folgenden EC2-Auto Scaling-Prozesse ausgesetzt: `HealthCheck`, `ReplaceUnhealthy`, `AZRebalance`, `AlarmNotification` und `ScheduledActions`. Hinweis: Unterbrechen Sie die Prozessstypen `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 Eigenschaft `LaunchTemplate` verweist auf den logischen Namen einer [AWS::EC2::LaunchTemplate](#)-Ressource, die an anderer Stelle in derselben Vorlage deklariert ist.

Weitere Informationen zu den Attributen `CreationPolicy` und `UpdatePolicy` finden Sie unter [Ressourcenattribut-Referenz](#).

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-Gruppe mithilfe einer stufenweisen Skalierungsrichtlinie 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 wird

ScalingAdjustment auf 1 festgelegt. Wenn der Schwellenwert für den Alarm überschritten wird, wird die Anzahl der EC2-Instances in der Gruppe also von der Richtlinie um 1 erhöht.

Die Ressource [AWS::CloudWatch::Alarm](#) CPUAlarmHigh legt die Skalierungsrichtlinie ASGScalingPolicyHigh als Aktion fest, die ausgeführt werden soll, wenn sich der Alarm im Zustand ALARM befindet (AlarmActions). Die Eigenschaft Dimensions verweist auf den logischen Namen einer [AWS::AutoScaling::AutoScalingGroup](#)-Ressource, die an anderer Stelle in derselben Vorlage deklariert ist.

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 Instancegruppe mit attributbasierter Instancetypauswahl 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 Eigenschaft LaunchTemplate verweist auf den logischen Namen einer [AWS::EC2::LaunchTemplate](#)-Ressource, die an anderer Stelle in derselben Vorlage deklariert ist.

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-Skalierungsgruppe, für die Sie Werte für die Eigenschaften `ImageId`, `InstanceType` und `SecurityGroups` angeben. Die Eigenschaft `SecurityGroups` gibt sowohl den logischen Namen einer [AWS::EC2::SecurityGroup](#)-Ressource an, die an anderer Stelle in der Vorlage angegeben ist, als auch eine bestehende EC2-Sicherheitsgruppe namens `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-Gruppe, die eine Startkonfiguration verwendet

Dieses Beispiel zeigt eine [AWS::AutoScaling::AutoScalingGroup](#)-Ressource mit einer einzigen Instance. 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 Eigenschaft `LaunchConfigurationName` verweist auf eine [AWS::AutoScaling::LaunchConfiguration](#)-Ressource mit dem logischen Namen `mySimpleConfig`, die 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 CloudFormation

Dieser Abschnitt enthält CloudFormation Vorlagenbeispiele für Richtlinien zur Skalierung von Application Auto Scaling und geplante Aktionen für verschiedene AWS Ressourcen.

Important

Wenn ein Snippet für die automatische Skalierung von Anwendungen in der Vorlage enthalten ist, müssen Sie möglicherweise mit dem Attribut `DependsOn` eine Abhängigkeit von der spezifischen skalierbaren Ressource deklarieren, die durch die Vorlage erstellt wird. Dies überschreibt die Standardparallelität und weist 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 AppStream Flotte](#)
- [Erstellen Sie eine Skalierungsrichtlinie für einen Aurora DB-Cluster](#)
- [Erstellen einer Skalierungsrichtlinie für eine DynamoDB-Tabelle](#)
- [Erstellen Sie eine Skalierungsrichtlinie für einen Amazon ECS-Service \(Metriken: durchschnittliche CPU und Speicher\)](#)
- [Erstellen Sie eine Skalierungsrichtlinie für einen Amazon ECS-Service \(Metrik: durchschnittliche Anzahl der Anfragen pro Ziel\)](#)
- [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 AppStream Flotte

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie erstellen und sie auf eine [AWS::AppStream::Fleet](#)-Ressource unter Verwendung der [AWS::ApplicationAutoScaling::ScalingPolicy](#)-Ressource anwenden. Die Ressource [AWS::ApplicationAutoScaling::ScalableTarget](#) 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 intrinsischen Funktionen `Fn::Join` und `Ref`, um die Eigenschaft `ResourceId` mit dem logischen Namen der Ressource `AWS::AppStream::Fleet` zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter [Intrinsische Funktionsreferenz](#).

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 Schnipsel registrieren Sie eine [AWS::RDS::DBCluster](#) Ressource. Die Ressource [AWS::ApplicationAutoScaling::ScalableTarget](#) zeigt an, dass der DB-Cluster dynamisch skaliert werden soll, so dass er über ein bis acht Aurora-Replikate verfügt. Sie wenden auch eine Zielverfolgungs-Skalierungsrichtlinie auf den Cluster an, indem Sie die Ressource [AWS::ApplicationAutoScaling::ScalingPolicy](#) verwenden.

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 intrinsische Funktion `Fn::Sub` verwendet, um die Eigenschaft `ResourceId` mit dem logischen Namen der Ressource `AWS::RDS::DBCluster` zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter [Intrinsische Funktionsreferenz](#).

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 einer Skalierungsrichtlinie für eine DynamoDB-Tabelle

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie mit dem Richtlinientyp `TargetTrackingScaling` erstellen und sie auf eine [AWS::DynamoDB::Table](#)-Ressource unter Verwendung der [AWS::ApplicationAutoScaling::ScalingPolicy](#)-Ressource anwenden. Die Ressource [AWS::ApplicationAutoScaling::ScalableTarget](#) deklariert ein skalierbares Ziel, auf das diese Richtlinie angewendet wird, mit einem Minimum von fünf Schreibkapazitätseinheiten und einem Maximum von 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 intrinsischen Funktionen `Fn::Join` und `Ref`, um die Eigenschaft `ResourceId` mit dem logischen Namen der Ressource `AWS::DynamoDB::Table` zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter [Intrinsische Funktionsreferenz](#).

Note

Weitere Informationen zum Erstellen einer CloudFormation Vorlage für DynamoDB-Ressourcen finden Sie im Blogbeitrag [How to use 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 Speicher)

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie erstellen und sie auf eine [AWS::ECS::Service](#)-Ressource unter Verwendung der [AWS::ApplicationAutoScaling::ScalingPolicy](#)-Ressource anwenden. Die Ressource [AWS::ApplicationAutoScaling::ScalableTarget](#) 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. Es verwendet die intrinsischen Funktionen Fn::Join und Ref, um die Eigenschaft ResourceId mit den logischen Namen der Ressourcen [AWS::ECS::Cluster](#) (myContainerCluster) und AWS::ECS::Service (myService) zu konstruieren, die in derselben Vorlage angegeben sind. Weitere Informationen finden Sie unter [Intrinsische Funktionsreferenz](#).

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 der 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 intrinsischen Funktionen `Fn::Join` und `Fn::GetAtt`, um die Eigenschaft `ResourceLabel` 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 [Intrinsische Funktionsreferenz](#).

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

Dieser Ausschnitt registriert die bereitgestellte Gleichzeitigkeit für einen Funktionsalias

([AWS::Lambda::Alias](#)) namens BLUE unter Verwendung der Ressource

[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 intrinsischen Funktionen `Fn::Join` und `Ref` in der Eigenschaft `RoleARN`, um den ARN der mit dem Dienst verknüpften Rolle anzugeben. Es verwendet die intrinsische Funktion `Fn::Sub`, um die Eigenschaft `ResourceId` mit dem logischen Namen der Ressource [AWS::Lambda::Function](#) oder [AWS::Serverless::Function](#) zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter [Intrinsische Funktionsreferenz](#).

Note

Sie können keine bereitgestellte Gleichzeitigkeit für einen Alias zuweisen, der auf die unveröffentlichte Version (\$LATEST) verweist.

Weitere Informationen zum Erstellen einer 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" : "AWS::Lambda::ProvisionedConcurrency"
    }
  },
  "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 Sie zwei geplante Aktionen erstellen, die nur einmal für eine [AWS::EC2::SpotFleet](#)-Ressource unter Verwendung der [AWS::ApplicationAutoScaling::ScalableTarget](#)-Ressource stattfinden. Die Zeitzone für jede einmalig geplante Aktion ist UTC.

Es verwendet die intrinsischen Funktionen `Fn::Join` und `Ref`, um die Eigenschaft `ResourceId` mit dem logischen Namen der Ressource `AWS::EC2::SpotFleet` zu erstellen, die in derselben Vorlage angegeben ist. Weitere Informationen finden Sie unter [Intrinsische Funktionsreferenz](#).

 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 beigelegt. 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'

```

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 CloudFormation aus der Vorlage ein Stapel erstellt wird, wird der erstellte myStack, 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 Format `Outputs.nestedstackoutputname` ist für die Angabe von Ausgabewerten aus verschachtelten Stapeln reserviert und kann an beliebiger Stelle innerhalb der enthaltenen 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 CloudFormation erstellt wird, werden die in der `Parameters` Eigenschaft deklarierten Wertepaare als Eingabeparameter für die Vorlage verwendet, mit der der `myStackWithParams` Stapel erstellt wurde. 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

Verwendung einer Wartebedingung mit einer Amazon EC2-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 die folgende Beispielvorlage, in der eine Amazon EC2-Instance mit einer Wartebedingung deklariert wird. Die Wartebedingung `myWaitCondition` verwendet `myWaitConditionHandle` für die Signalisierung, verwendet das Attribut `DependsOn`, um anzugeben, dass die Wartebedingung ausgelöst wird, nachdem die Amazon EC2-Instanceressource erstellt wurde, und verwendet die Eigenschaft `Timeout`, um eine Dauer von 4500 Sekunden für die Wartebedingung anzugeben. Außerdem wird die vorgezeichenbehaftete URL, die die Wartebedingung signalisiert, mit der Eigenschaft `UserData` der `Ec2Instance`-Ressource an die Amazon EC2-Instance weitergegeben. So kann eine Anwendung oder ein Skript, das auf dieser Amazon EC2-Instance läuft, die vorgezeichenbehaftete URL abrufen und sie verwenden, um einen Erfolg oder Misserfolg der Wartebedingung zu signalisieren. Sie müssen `cfn-signal` verwenden oder die Anwendung oder das Skript erstellen, das die Wartebedingung signalisiert. Der Ausgabewert `ApplicationData` enthält die Daten, die vom Wartezustandssignal zurückgegeben werden.

Weitere Informationen finden Sie unter [Wartebedingungen in einer CloudFormation Vorlage erstellen](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {
        "AMI" : "ami-0123456789abcdef0"
      },
      "us-west-1" : {
        "AMI" : "ami-0987654321fedcba0"
      },
      "eu-west-1" : {
```



```

        "AMI" : "ami-0abcdef123456789a"
    },
    "ap-northeast-1" : {
        "AMI" : "ami-0fedcba987654321b"
    },
    "ap-southeast-1" : {
        "AMI" : "ami-0c1d2e3f4a5b6c7d8"
    }
}
},
"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-0123456789abcdef0
    us-west-1:
      AMI: ami-0987654321fedcba0
    eu-west-1:
      AMI: ami-0abcdef123456789a
    ap-northeast-1:
      AMI: ami-0fedcba987654321b
    ap-southeast-1:
      AMI: ami-0c1d2e3f4a5b6c7d8
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 den Erfolg einer Wartebedingung signalisiert. Sie müssen die Befehlszeile in der Eigenschaft `UserData` der EC2-Instance 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=ik1twT6hpS4cgNAw7wy0oRejVoo
%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=ik1twT6hpS4cgNAw7wy0oRejVoo
%3D"
```

CloudFront Amazon-Vorlagenschnipsel

Verwenden Sie diese Beispielvorgangsausschnitte zusammen mit Ihrer CloudFront Amazon-Vertriebsressource in 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"
          }
        }
      }
    }
  }
}
```



```

- 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 [DistributionConfig](#) wird eine Liste von Ursprüngen bereitgestellt und [DefaultCacheBehavior](#) ein 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",

```



```
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 unter [Alias-Ressourcendatensatz für eine Verteilung CloudFront](#).

CloudWatch Amazon-Vorlagenschnipsel

Verwenden Sie diese Beispielvorlagenausschnitte, um Ihre CloudWatch Amazon-Ressourcen in zu beschreiben. 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 folgenden Beispiel-Ausschnitt erstellt einen Alarm, der eine Benachrichtigung versendet, wenn die durchschnittliche CPU-Auslastung einer Amazon EC2-Instance 90 Prozent länger als 60 Sekunden überschreitet und das über drei Bewertungszeiträume.

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" : "GreaterThanOrEqualToThreshold",
    "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: GreaterThanOrEqualToThreshold
    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": "{\n  \"widgets\": [\n    {\n      \"type\": \"metric\",\n      \"x\": 0,\n      \"y\": 0,\n      \"width\": 12,\n      \"height\": 6,\n      \"properties\": {\n        \"metrics\": [\n          [\n            \"AWS/EC2\",\n            \"CPUUtilization\",\n            \"InstanceId\",\n            \"i-012345\"\n          ],\n          {\n            \"period\": 300,\n            \"stat\": \"Average\",\n            \"region\": \"us-east-1\",\n            \"title\": \"EC2 Instance CPU\"\n          }\n        ]\n      }\n    },\n    {\n      \"type\": \"text\",\n      \"x\": 0,\n      \"y\": 7,\n      \"width\": 3,\n      \"height\": 3,\n      \"properties\": {\n        \"markdown\": \"Hello world\"\n      }\n    }\n  ]\n}"
    }
  }
}

```

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 Amazon EC2 EC2-Instances oder anderen Quellen überwachen. Sie können 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

Die folgende Vorlage zeigt, wie Sie einen Webserver auf Amazon Linux 2023 mit CloudWatch Logs-Integration einrichten. Die Vorlage führt die folgenden Aufgaben aus:

- Installiert Apache und PHP.
- Konfiguriert den CloudWatch Agenten so, dass er CloudWatch Apache-Zugriffsprotokolle an Logs weiterleitet.
- Richtet eine IAM-Rolle ein, damit der CloudWatch Agent Protokolldaten an CloudWatch Logs senden kann.
- Erstellen Sie benutzerdefinierte Alarme und Benachrichtigungen zur Überwachung von 404-Fehlern oder hoher Bandbreitennutzung.

Protokollereignisse vom Webserver liefern Metrikdaten für CloudWatch Alarmer. Die beiden Metrikfilter beschreiben, wie die Protokollinformationen in CloudWatch Metriken umgewandelt werden. Die 404-Metrik zählt die Zahl von 404-Vorkommnissen. Die Größenmetrik verfolgt die Größe einer Anforderung. Die beiden CloudWatch Alarmer 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": "Sample template that sets up and configures CloudWatch Logs on
Amazon Linux 2023 instance.",
  "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 when CloudWatch alarms are
triggered (404 errors or high bandwidth usage)",
      "Type": "String"
    }
  },
  "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:PutLogEvents",
                                "logs:DescribeLogStreams",
                                "logs:DescribeLogGroups",
                                "logs:CreateLogGroup",
                                "logs:CreateLogStream"
                            ],
                            "Resource": "*"
                        }
                    ]
                }
            }
        ],
        "LogRoleInstanceProfile": {
            "Type": "AWS::IAM::InstanceProfile",
            "Properties": {
                "Path": "/",
                "Roles": [{"Ref": "LogRole"}]
            }
        }
    },
    "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 on Amazon Linux 2023",
      "AWS::CloudFormation::Init": {
        "config": {
          "packages": {
            "dnf": {
              "httpd": [],
              "php": [],
              "php-fpm": []
            }
          },
          "files": {
            "/etc/amazon-cloudwatch-agent/amazon-cloudwatch-
agent.json": {
              "content": {
                "logs": {
                  "logs_collected": {
                    "files": {
                      "collect_list": [{

```

```

        "file_path": "/var/log/httpd/
access_log",
        "log_group_name": {"Ref":
        "WebServerLogGroup"},
        "log_stream_name": "{instance_id}/
apache.log",
        "timestamp_format": "%d/%b/%Y:%H:
%M:%S %z"
    }
}
}
},
    "mode": "000644",
    "owner": "root",
    "group": "root"
},
    "/var/www/html/index.php": {
        "content": "<?php\\necho '<h1>AWS CloudFormation sample
PHP application on Amazon Linux 2023</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": {
            "systemd": {
                "httpd": {
                    "enabled": "true",
                    "ensureRunning": "true"
                },
                "php-fpm": {
                    "enabled": "true",
                    "ensureRunning": "true"
                }
            }
        },
        "CreationPolicy": {
            "ResourceSignal": {
                "Timeout": "PT5M"
            }
        },
        "Properties": {
            "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}",
            "KeyName": {"Ref": "KeyName"},

```

```

    "InstanceType": "t3.micro",
    "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
    "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
    "UserData": {"Fn::Base64": {"Fn::Join": [ "", [
        "#!/bin/bash\n",
        "dnf update -y aws-cfn-bootstrap\n",
        "dnf install -y amazon-cloudwatch-agent\n",
        "/opt/aws/bin/cfn-init -v --stack ", {"Ref": "AWS::StackName"}, "
--resource WebServerHost --region ", {"Ref": "AWS::Region"}, "\n",
        "\n",
        "# Verify Apache log directory exists and create if needed\n",
        "mkdir -p /var/log/httpd\n",
        "\n",
        "# Start CloudWatch agent\n",
        "/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -
a fetch-config -m ec2 -c file:/etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json
-s\n",
        "\n",
        "# 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",
    "DeletionPolicy": "Retain",
    "UpdateReplacePolicy": "Retain",
    "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::Sub": "http://${WebServerHost.PublicDnsName}"},
            "Description": "URL for the 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": "WebServerLogGroup"}
        }
    }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Sample template that sets up and configures CloudWatch Logs on Amazon
  Linux 2023 instance.
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 when CloudWatch alarms are triggered (404 errors or high bandwidth usage)

Type: String

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:PutLogEvents'
- 'logs:DescribeLogStreams'
- 'logs:DescribeLogGroups'
- 'logs:CreateLogGroup'
- 'logs:CreateLogStream'

Resource: '*'

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 on Amazon Linux 2023
    'AWS::CloudFormation::Init':
      config:
        packages:
          dnf:
            httpd: []
            php: []
            php-fpm: []
        files:
          /etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json:
            content: !Sub |
              {
                "logs": {
                  "logs_collected": {
                    "files": {
                      "collect_list": [
                        {
                          "file_path": "/var/log/httpd/access_log",
                          "log_group_name": "${WebServerLogGroup}",
                          "log_stream_name": "{instance_id}/apache.log",
                          "timestamp_format": "%d/%b/%Y:%H:%M:%S %z"
                        }
                      ]
                    }
                  }
                }
              }

```

```

        mode: '000644'
        owner: root
        group: root
/var/www/html/index.php:
        content: |
            <?php echo '<h1>AWS CloudFormation sample PHP application on Amazon
Linux 2023</h1>';
            ?>
        mode: '000644'
        owner: apache
        group: apache
/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.WebServerHost.Metadata.AWS::CloudFormation::Init
            action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerHost --
region ${AWS::Region}
            runas=root
    services:
        systemd:
            httpd:
                enabled: 'true'
                ensureRunning: 'true'
            php-fpm:
                enabled: 'true'
                ensureRunning: 'true'
    CreationPolicy:
        ResourceSignal:
            Timeout: PT5M
    Properties:
        ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
        KeyName: !Ref KeyName
        InstanceType: t3.micro
        SecurityGroupIds:

```

```

    - !Ref WebServerSecurityGroup
    IamInstanceProfile: !Ref LogRoleInstanceProfile
    UserData: !Base64
      Fn::Sub: |
        #!/bin/bash
        dnf update -y aws-cfn-bootstrap
        dnf install -y amazon-cloudwatch-agent
        /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServerHost
--region ${AWS::Region}

        # Verify Apache log directory exists and create if needed
        mkdir -p /var/log/httpd

        # Start CloudWatch agent
        /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -c file:/etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json -s

        # Signal success
        /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerHost --region ${AWS::Region}
        echo "Done"
    WebServerLogGroup:
      Type: AWS::Logs::LogGroup
      DeletionPolicy: Retain
      UpdateReplacePolicy: Retain
      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: !Sub 'http://${WebServerHost.PublicDnsName}'
  Description: URL for the 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 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.",
  "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 when CloudWatch alarms are
triggered (404 errors)",
        "Type": "String"
    }
},
"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": {
                            "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": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

ApplicationEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSystemEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

SystemEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    }
}

```

```

    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSecurityEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
SecurityEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchEC2ConfigLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
EC2ConfigLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchCfnInitLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
CfnInitLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchIISLogs",

```

```

        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

IISLogs",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatch",
        "Parameters": {
            "AccessKey": "",
            "NameSpace": "Windows/Default",
            "Region": {"Ref": "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": {
                "Fn::Sub": "cfn-signal.exe -e 0 --resource
WebServerHost --stack ${AWS::StackName} --region ${AWS::Region}"
            },
            "waitAfterCompletion": "0"
        }
    }
}
},
"Properties": {
    "KeyName": {

```

```

        "Ref": "KeyPair"
    },
    "ImageId": "{{resolve:ssm:/aws/service/ami-windows-latest/
Windows_Server-2012-R2_RTM-English-64Bit-Base}}",
    "InstanceType": "t2.xlarge",
    "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
    "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
    "UserData": {
        "Fn::Base64": {
            "Fn::Join": [
                "",
                [
                    "<script>\n",
                    "wmic product where \"description='Amazon SSM Agent' \"
uninstall\n",
                    "wmic product where \"description='aws-cfn-bootstrap'
\" uninstall \n",
                    "start /wait c:\\\\Windows\\system32\\msiexec /passive /
qn /i https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi\n",
                    "powershell.exe -Command \"iwr https://
s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe -UseBasicParsing -OutFile C:\\\\AmazonSSMAgentSetup.exe\\\"\\n",
                    "start /wait C:\\\\AmazonSSMAgentSetup.exe /install /
quiet\n",
                    "cfn-init.exe -v -c config -s ", {"Ref":
"AWS::StackName"}, " --resource WebServerHost --region ", {"Ref": "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 the 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.
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 when CloudWatch alarms are triggered (404
    errors)
    Type: String
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: "{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2012-
R2_RTM-English-64Bit-Base}}"
        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 the 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::LogGroup](#)[AWS::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`-Skalierungsrichtlinie, 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 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 Amazon EC2 EC2-Instances mit CloudFormation](#)
- [Erstellen Sie Startvorlagen mit CloudFormation](#)
- [Sicherheitsgruppen verwalten mit CloudFormation](#)
- [Zuweisen und Zuordnen von Elastic IP-Adressen zu CloudFormation](#)
- [Konfigurieren Sie Amazon VPC-Ressourcen mit CloudFormation](#)

Konfigurieren Sie Amazon EC2 EC2-Instances mit CloudFormation

Die folgenden Ausschnitte zeigen, wie Sie Amazon-EC2-Instances mit CloudFormation konfigurieren.

Kategorien von Ausschnitten

- [Allgemeine Amazon-EC2-Konfigurationen](#)
- [Angaben der Blockgerät-Zuweisungen für eine Instance](#)

Allgemeine Amazon-EC2-Konfigurationen

Die folgenden Ausschnitte zeigen allgemeine Konfigurationen für Amazon-EC2-Instances mit CloudFormation.

Beispielausschnitte

- [Erstellen einer Amazon-EC2-Instance in einer angegebenen Availability Zone](#)
- [Konfigurieren einer getaggten Amazon-EC2-Instance mit einem EBS-Volume und Benutzerdaten](#)
- [Definieren des DynamoDB-Tabellennamens in den Benutzerdaten für den Start einer Amazon-EC2-Instance](#)
- [Erstellen eines Amazon-EBS-Volumes mit DeletionPolicy](#)

Erstellen einer Amazon-EC2-Instance in einer angegebenen Availability Zone

Das folgende Snippet erstellt mithilfe einer Ressource eine Amazon EC2 EC2-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
```

Konfigurieren einer getaggtten Amazon-EC2-Instance mit einem EBS-Volume und Benutzerdaten

Mit dem folgenden Ausschnitt wird eine Amazon-EC2-Instance mit einem Tag, einem EBS-Volume und Benutzerdaten erstellt. Es verwendet eine Ressource. [AWS::EC2::Instance](#) In derselben Vorlage müssen Sie eine [AWS::EC2::SecurityGroup](#)Ressource, eine [AWS::SNS::Topic](#)Ressource und eine [AWS::EC2::Volume](#)Ressource 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 Sie Ihre Amazon EC2-Ressourcen](#) im Amazon EC2-Benutzerhandbuch.

Informationen über Benutzerdaten finden Sie unter [Verwenden Sie Instance-Metadaten zur Verwaltung Ihrer EC2-Instance](#) im Amazon EC2-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 des DynamoDB-Tabellennamens in den Benutzerdaten für den Start einer Amazon-EC2-Instance

Mit dem folgenden Ausschnitt wird eine Amazon-EC2-Instance erstellt und in den Benutzerdaten ein DynamoDB-Tabellenname definiert, der beim Start an die Instance übergeben wird. Es verwendet

eine [AWS::EC2::Instance](#) Ressource. Sie können Parameter oder dynamische Werte in den Benutzerdaten definieren, um sie beim Start an eine EC2-Instance zu übergeben.

Weitere Informationen über Benutzerdaten finden Sie unter [Verwenden Sie Instance-Metadaten zur Verwaltung Ihrer EC2-Instance](#) im Amazon EC2-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 ein Amazon EBS-Volume mithilfe einer Amazon EC2 EC2-Ressource. [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::Volume](#) Ressource 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::Volume](#) Ressource 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::Volume](#) Ressource 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::Volume](#) Ressource 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 über Blockgeräte-Zuordnungen finden Sie unter [Blockgeräte-Zuordnungen für Volumes auf Amazon EC2-Instances](#) im Amazon EC2-Benutzerhandbuch.

Szenarien

- [Angaben der Blockgerät-Zuweisungen für zwei EBS-Volumes](#)
- [Anzeigen der Blockgerät-Zuweisung für ein Instance-Speicher-Volume](#)

Angaben der Blockgerät-Zuweisungen für zwei EBS-Volumes

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": "{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}",
    "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: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
  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" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}",
    "KeyName" : { "Ref" : "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups" : [{ "Ref" : "Ec2SecurityGroup" }],
    "BlockDeviceMappings" : [
      {
        "DeviceName" : "/dev/sdc",
        "VirtualName" : "ephemeral0"
      }
    ]
  }
}

```

YAML

```

EC2Instance:
  Type: AWS::EC2::Instance
  Properties:

```

```
ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-
gp2}}'
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 Amazon EC2 EC2-Startvorlage mithilfe von CloudFormation. Mit Startvorlagen können Sie Vorlagen für die Konfiguration und Bereitstellung von Amazon EC2-Instances innerhalb von AWS erstellen. Mit Startvorlagen können Sie Startparameter speichern, so dass Sie diese nicht bei jedem Start einer Instance neu angeben müssen. Weitere Beispiele finden Sie im Abschnitt [Beispiele](#) in der `AWS::EC2::LaunchTemplate` Ressource.

Weitere Informationen über Startvorlagen finden Sie unter [Speichern von Instance-Startparametern in Amazon EC2-Startvorlagen](#) im Amazon EC2 User Guide.

Informationen zur Erstellung von Startvorlagen für die Verwendung mit Auto Scaling-Gruppen finden Sie unter [Auto Scaling-Startvorlagen](#) im Amazon EC2 Auto Scaling User Guide.

Kategorien von Ausschnitten

- [Erstellen Sie eine Startvorlage, in der Sicherheitsgruppen, Tags, Benutzerdaten und eine IAM-Rolle angegeben sind.](#)

Erstellen Sie eine Startvorlage, in der Sicherheitsgruppen, Tags, Benutzerdaten und eine IAM-Rolle angegeben sind.

Dieser Ausschnitt zeigt eine `AWS::EC2::LaunchTemplate` Ressource, 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::SecurityGroup` Ressource `abmyNewEC2SecurityGroup`, 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 laufen, ermöglicht, Aktionen in Ihrem Namen durchzuführen. Dieses Beispiel zeigt eine [AWS::IAM::Role](#) Ressource für die Startvorlage, die die `IamInstanceProfile` Eigenschaft verwendet, um die IAM-Rolle anzugeben. Die `Ref` Funktion ruft den Namen der [AWS::IAM::InstanceProfile](#) Ressource `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 Sicherheitsgruppen und Amazon EC2 EC2-Instances verwalten können, CloudFormation um den Zugriff auf Ihre Ressourcen zu kontrollieren. AWS

Kategorien von Ausschnitten

- [Verknüpfen einer Amazon-EC2-Instance mit einer Sicherheitsgruppe](#)

- [Erstellen von Sicherheitsgruppen mit Regeln für eingehenden Datenverkehr](#)
- [Erstellen eines Elastic Load Balancers mit einer Regel für eingehenden Datenverkehr für eine Sicherheitsgruppe](#)

Verknüpfen einer Amazon-EC2-Instance mit einer Sicherheitsgruppe

Die folgenden Beispielausschnitte zeigen, wie Sie eine Amazon-EC2-Instance mithilfe von CloudFormation mit einer standardmäßigen Amazon-VPC-Sicherheitsgruppe verknüpfen.

Beispielausschnitte

- [Verknüpfen einer Amazon-EC2-Instance mit einer standardmäßigen VPC-Sicherheitsgruppe](#)
- [Erstellen einer Amazon-EC2-Instance mit einem angefügten Volume und einer Sicherheitsgruppe](#)

Verknüpfen einer Amazon-EC2-Instance mit einer standardmäßigen VPC-Sicherheitsgruppe

Mit dem folgenden Ausschnitt wird eine Amazon-VPC, ein Subnetz in der VPC und eine Amazon-EC2-Instance erstellt. 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 in der VPC mithilfe einer [AWS::EC2:: Subnet](#)-Ressource 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 Ressource gestartet. [AWS::EC2::Instance](#) 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. Der `ImageId` verwendet einen Systems Manager-Parameter, um dynamisch das neueste Amazon Linux 2 AMI abzurufen.

Die ID der Sicherheitsgruppe wird mit der Funktion `Fn::GetAtt` abgerufen, die die Standardsicherheitsgruppe aus der Ressource `MyVPC` abrufen.

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": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
    "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: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-
gp2}}'
    SecurityGroupIds:
      - Fn::GetAtt:
          - MyVPC
          - DefaultSecurityGroup
    SubnetId:
      Ref: MySubnet

```

Erstellen einer Amazon-EC2-Instance mit einem angefügten Volume und einer Sicherheitsgruppe

Das folgende Snippet erstellt eine Amazon EC2 EC2-Instance mithilfe einer [AWS::EC2::Instance](#) Ressource, die von einem bestimmten AMI aus gestartet wird. Die Instance ist einer Sicherheitsgruppe zugeordnet, die eingehenden SSH-Verkehr über Port 22 von einer angegebenen IP-Adresse aus unter Verwendung einer Ressource zulässt. [AWS::EC2::SecurityGroup](#) Es erstellt mithilfe einer [AWS::EC2::Volume](#) Ressource 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 über Sicherheitsgruppen finden Sie unter [Amazon EC2-Sicherheitsgruppen für Ihre Amazon EC2-Instances](#) im Amazon EC2-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

Mit dem folgenden Ausschnitt wird eine Sicherheitsgruppe für eine Amazon-EC2-Instance mit zwei Regeln für eingehenden Datenverkehr erstellt. Diese Regeln erlauben eingehenden TCP-Verkehr an den angegebenen Ports aus den festgelegten CIDR-Bereichen. Eine [AWS::EC2::SecurityGroup](#) Ressource 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 über Sicherheitsgruppen finden Sie unter [Amazon EC2-Sicherheitsgruppen für Ihre Amazon EC2-Instances](#) im Amazon EC2-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::SecurityGroup](#) Ressource, um zwei Amazon EC2-Sicherheitsgruppen zu erstellen, und. SGroup1 SGroup2 [Eingangsregeln, die die Kommunikation zwischen den beiden Sicherheitsgruppen ermöglichen, werden mithilfe der Ingress-Ressource erstellt.](#) [AWS::EC2::SecurityGroup](#) SGroup1Ingressrichtet eine Eingangsregel einSGroup1, die eingehenden TCP-Verkehr auf Port 80 von der Quellsicherheitsgruppe zulässt,. SGroup2 SGroup2Ingressrichtet eine Eingangsregel einSGroup2, 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::LoadBalancer](#) Ressource in der angegebenen Availability Zone. Die [AWS::ElasticLoadBalancing::LoadBalancer](#) Ressource 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::SecurityGroup](#) Ressource, 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

```

Zuweisen und Zuordnen von Elastic IP-Adressen zu CloudFormation

Die folgenden Vorlagenausschnitte sind Beispiele für Elastic-IP-Adressen (EIPs) in Amazon EC2. Diese Beispiele behandeln die Zuweisung, Zuordnung und Verwaltung von EIPs für Ihre Instances.

Beispielausschnitte

- [Zuweisen einer Elastic IP-Adresse und deren Verknüpfung mit einer Amazon-EC2-Instance](#)
- [Verknüpfen einer Elastic-IP-Adresse mit einer Amazon-EC2-Instance durch Angabe der IP-Adresse](#)
- [Verknüpfen einer Elastic-IP-Adresse mit einer Amazon-EC2-Instance durch Angabe der Zuweisungs-ID der IP-Adresse](#)

Zuweisen einer Elastic IP-Adresse und deren Verknüpfung mit einer Amazon-EC2-Instance

Das folgende Snippet weist eine Amazon EC2 Elastic IP (EIP) -Adresse zu und verknüpft sie mithilfe einer Ressource mit einer Amazon EC2 EC2-Instance. [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 Adresspool, der aus einem öffentlichen IPv4 Adressbereich erstellt wurde, den Sie AWS zur Verwendung mit Ihren AWS Ressourcen verwenden.

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-Adressen](#) im Benutzerhandbuch von Amazon EC2.

JSON

```
"ElasticIP": {
  "Type": "AWS::EC2::EIP",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    }
  }
}
```

YAML

```
ElasticIP:
  Type: AWS::EC2::EIP
  Properties:
    InstanceId: !Ref EC2Instance
```

Verknüpfen einer Elastic-IP-Adresse mit einer Amazon-EC2-Instance durch Angabe der IP-Adresse

Der folgende Ausschnitt verknüpft eine bestehende Amazon EC2 Elastic IP-Adresse mit einer EC2-Instance, die eine [AWS::EC2::](#) -Ressource verwendet. EIPAssociation 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
```

Verknüpfen einer Elastic-IP-Adresse mit einer Amazon-EC2-Instance durch Angabe der Zuweisungs-ID der IP-Adresse

Der folgende Ausschnitt verknüpft eine bestehende Elastic IP-Adresse mit einer Amazon EC2 EC2-Instance, indem die Zuweisungs-ID mithilfe einer [AWS::EC2::](#) -Ressource angegeben wird. EIPAssociation 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. VPCs ermöglichen 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

- [Aktivieren Sie den Internetzugang IPv6 nur für ausgehenden Datenverkehr in einer VPC](#)
- [Elastic Network-Schnittstelle \(ENI\) Vorlagenausschnitte](#)

Aktivieren Sie den Internetzugang IPv6 nur für ausgehenden Datenverkehr in einer VPC

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-Resource](#) verwendet. Eine Routentabelle ist dieser VPC-Ressource mithilfe einer [AWS::EC2::RouteTable](#) Ressource zugeordnet. Über die Routing-Tabelle werden Routen für Instances in der VPC verwaltet. An [AWS::EC2::EgressOnlyInternetGateway](#) wird 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::Route](#) Ressource 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 Amazon EC2 EC2-Instance mit angehängten elastischen Netzwerkschnittstellen () ENIs

Das folgende Beispiel-Snippet erstellt eine Amazon EC2-Instance unter Verwendung einer [AWS::EC2::Instance](#)-Ressource in der angegebenen Amazon VPC und dem angegebenen Subnetz. Sie verbindet zwei Netzwerkschnittstellen (ENIs) mit der Instance, ordnet Elastic IP-Adressen den Instances über die angehängte ENIs Verbindung 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 mit [AWS::EC2::NetworkInterface](#) Ressourcen erstellt. Elastische IP-Adressen werden über [AWS::EC2::EIP](#) Ressourcen angegeben, die der `vpc`-Domain zugewiesen sind. Diese elastischen IP-Adressen sind mit den Netzwerkschnittstellen über die Ressourcen [AWS::EC2::EIPAssociation](#) verbunden.

Im Abschnitt `Outputs` werden Werte oder Ressourcen definiert, auf die Sie nach Erstellung des Stacks zugreifen möchten. In diesem Codeausschnitt lautet die definierte Ausgabe `InstancePublicIp`, was für die öffentliche IP-Adresse der vom Stack erstellten EC2-Instance steht. [Sie können diese Ausgabe auf der Registerkarte „Ausgabe“ in der 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
```

Amazon Elastic Container Service Beispielvorlagen

Amazon Elastic Container Service (Amazon ECS) ist ein Container-Management-Service, mit dem Sie unkompliziert Docker-Container in einem Cluster aus Amazon Elastic Compute Cloud (Amazon EC2)-Instances ausführen, beenden und verwalten können.

Erstellen Sie einen Cluster mit AL2023 Amazon ECS-Optimized-AMI

Definieren Sie einen Cluster, der einen Kapazitätsanbieter verwendet, der AL2023-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 association 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 communicate 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

```

Einen Dienst einrichten

Die folgende Vorlage definiert einen Service, der den Kapazitätsanbieter verwendet, um Kapazität zur Ausführung anzufordern AL2023 . Container werden auf den AL2023 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`\n"
    ] ] },
    "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,rsiz=1048576,wsiz=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::VPCEGatewayAttachment
  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
        fi
    done
done

```

```

        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 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-Umgebung, die eine Legacy-Startkonfiguration verwendet. Informationen zur Verwendung einer Startvorlage finden Sie unter [Startvorlagen](#) im AWS Elastic Beanstalk Entwicklerhandbuch.

Ersetzen Sie *solution-stack* durch einen Lösungs-Stack-Namen (Plattformversion). Verwenden Sie den Befehl, um eine Liste der verfügbaren Lösungss Stapel zu erhalten. AWS CLI `aws elasticbeanstalk list-available-solution-stacks`

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"
            ]
          }
        ]
      },
      "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 `AWS::ElasticLoadBalancingV2` beginnen. Um einen Classic Load Balancer zu erstellen, verwenden Sie die Ressourcentypen, die mit `AWS::ElasticLoadBalancing` beginnen.

Inhalt

- [ELBv2 Ressourcen](#)
- [Classic-Load-Balancer-Ressourcen](#)

ELBv2 Ressourcen

Dieses Beispiel definiert einen Application Load Balancer mit einem HTTP-Listener und einer Standardaktion, die den Datenverkehr an die Zielgruppe weiterleitet. Der Load Balancer verwendet die Standardeinstellungen für die Zustandsprüfung. 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
      }
    ]
  }
}
```

Classic-Load-Balancer-Ressourcen

Dieses Beispiel definiert einen Classic Load Balancer mit einem HTTP-Listener und ohne registrierte EC2 Instances. Der Load Balancer verwendet die Standardeinstellungen für die Zustandsprüfung.

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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).

Deklarieren einer IAM-Benutzerressource

Dieser Ausschnitt zeigt, wie Sie eine [AWS::IAM::User](#)-Ressource deklarieren, 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 Funktion `Fn::GetAtt` erhält das Attribut `Arn` der Ressource [AWS::SQS::Queue](#) `myqueue`.

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
```

Deklariert eine IAM-Zugriffsschlüssel-Ressource

Dieser Ausschnitt zeigt eine [AWS::IAM::AccessKey](#) Ressource. Die Ressource myaccesskey erstellt einen Zugriffsschlüssel und weist ihn einem IAM-Benutzer zu, der in der Vorlage als Ressource [AWS::IAM::User](#) 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, um den Zugriffsschlüssel und den geheimen Schlüssel für die myaccesskey-Ressource zu übergeben.

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

Dieser Ausschnitt 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 Ausschnitt ruft die Funktion `Fn::GetAtt` den ARN der Ressourcenwarteschlange [AWS::SQS::Queue](#) 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 Ressource [AWS::IAM::UserToGroupAddition](#) fügt Benutzer zu einer Gruppe hinzu. Im folgenden Ausschnitt fügt die Ressource `addUserToGroup` die folgenden Benutzer zu einer bestehenden

Gruppe namens `myexistinggroup2` hinzu: den bestehenden Benutzer `existinguser1` und den Benutzer `myuser`, der in der Vorlage als Ressource [AWS::IAM::User](#) 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
```

Deklariere eine IAM-Richtlinie

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie erstellen und sie auf mehrere Gruppen anwenden, indem Sie eine Ressource mit dem Namen [AWS::IAM::Policy](#) `mypolicy` verwenden. 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 Ressource `mypolicy` wendet die Richtlinie auf eine bestehende Gruppe namens `myexistinggroup1` und eine Gruppe `mygroup` an, die in der Vorlage als Ressource [AWS::IAM::Group](#) 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

```

Deklariere eine Amazon S3-Bucket-Richtlinie

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie erstellen und sie auf einen Amazon S3-Bucket anwenden, indem Sie die Ressource [AWS::S3::BucketPolicy](#) verwenden. Die mybucketpolicy-Ressource deklariert ein Richtlinienokument, 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

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie erstellen und sie mithilfe der Ressource [AWS::SNS::TopicPolicy](#) auf ein Amazon SNS-Thema anwenden. Die `mysnspolicy`-Ressource enthält eine `PolicyDocument`-Eigenschaft, die es der [AWS::IAM::User](#)-Ressource `myuser` ermöglicht, die `Publish`-Aktion für eine [AWS::SNS::Topic](#)-Ressource `mytopic` durchzuführen. 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

Dieser Ausschnitt zeigt, wie Sie eine Richtlinie erstellen und sie mit der Ressource [AWS::SQS::QueuePolicy](#) auf eine Amazon SQS-Warteschlange anwenden. Die Eigenschaft `PolicyDocument` ermöglicht es dem bestehenden Benutzer `myapp` (angegeben durch seinen ARN), die Aktion `SendMessage` für eine bestehende Warteschlange, die durch ihre URL angegeben ist, und eine [AWS::SQS::Queue](#) Ressource `myqueue` durchzuführen. 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-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 Informationen über den Ressourcentyp 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",
            "t3.micro"
        ]
    },
},

"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": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",

```

```

        "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
      - t3.micro
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: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
    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 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 Eigenschaft `EngineVersion` 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 `MyDbSecurityByCIDRIPGroup`. Details hierzu finden Sie unter [AWS::RDS::DBInstance](#). Die Ressource `DB Instance` hat auch ein Attribut `DeletionPolicy`, das auf `Snapshot` gesetzt ist. Mit diesem `Snapshot DeletionPolicy Set` 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 Ressource `DB Instance` hat auch ein Attribut `DeletionPolicy`, das auf `Snapshot` gesetzt ist. Mit diesem `Snapshot DeletionPolicy Set` 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 RDS DBSecurityGroup-Ressource mit Ingress-Autorisierung für den angegebenen CIDR-Bereich im Format `ddd.ddd.ddd.ddd/dd`. Einzelheiten finden Sie unter [AWS: :RDS:: DBSecurity Group](#) und [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-ffd72222",
        "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-ffd72222
    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 `mysite.example.com` enthält, der die Eigenschaft `HostedZoneId` verwendet, um die gehostete Zone anzugeben.

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 Domainnamen „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 [AliasTarget](#)Eigenschaft 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 Domännennamen 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. myCloudFrontDistribution bezieht 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-Ressourcen-Datensätze erstellen, müssen Sie Z2FDTNDATAQYW2 für die Eigenschaft HostedZoneId 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 mit CloudFormation zu 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 Domain](#)

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 die Blockierung des öffentlichen Zugriffs deaktiviert (öffentliche Leserechte sind für Buckets erforderlich, die für das Hosting von Websites eingerichtet wurden). Eine öffentliche Bucket-Policy wird dann zum Bucket hinzugefügt. Da für diese Bucket-Ressource ein `DeletionPolicy` Attribut auf `Retain` gesetzt ist, CloudFormation wird dieser Bucket beim Löschen des Stacks nicht gelöscht. Der Abschnitt `Output` verwendet `Fn::GetAtt` zum Abrufen des Attributs `WebsiteURL` und des Attributs `DomainName` der Ressource `S3Bucket`.

Note

Die folgenden Beispiele gehen davon aus, dass die Einstellungen `BlockPublicPolicy` und `RestrictPublicBuckets` Öffentlicher Zugriff sperren 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 Domain

Sie können Route 53 mit einer registrierten Domain verwenden. Im folgenden Beispiel wird vorausgesetzt, dass Sie in Route 53 bereits eine gehostete Zone für Ihre Domain 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 Domain 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: Konfigurieren einer statischen Website unter Verwendung einer bei Route 53 registrierten benutzerdefinierten Domain](#) im Amazon Simple Storage Service User Guide.

Note

Die folgenden Beispiele gehen davon aus, dass die Einstellungen `BlockPublicPolicy` und `RestrictPublicBuckets` Öffentlicher Zugriff sperren 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" : "Z300J2DXBE1FTB",
"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: Z3AQBSTGIFYJSTF

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: Z300J2DXBE1FTB

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 Vorlage Schnipsel

Amazon Timestream for InfluxDB macht es Anwendungsentwicklern und DevOps Teams leicht, vollständig verwaltete InfluxDB-Datenbanken AWS für Echtzeit-Zeitreihenanalysen 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 mit automatisiertem Software-Patching, -Backups und -Recovery einrichten,

migrieren, betreiben und skalieren. AWS [Sie finden diese Beispiele auch unter `aws-labs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb` on GitHub](https://github.com/aws-labs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb)

Themen

- [Minimale Stichprobe mit Standardwerten](#)
- [Vollständigeres Beispiel mit Parametern](#)

Diese 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
- Ein 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 eingerichtet, die nach Möglichkeit Standardwerte verwendet.

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. Die Parameter umfassen `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 EC2-basierten Instances, SQL Server auf Amazon RDS und Microsoft Active Directory über diese verwenden. Directory Service

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 anderen Windows 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 [AWS Windows AMI-Referenz](#).

CloudFormation unterstützt auch Softwarekonfigurationstools wie UserData Skripte, die Befehle ausführen PowerShell oder Batch-Befehle ausführen können, wenn eine EC2-Instance zum ersten Mal gestartet wird. Es bietet auch Hilfsskripte (cfn-init, cfn-signal cfn-get-metadata und cfn-hup) und unterstützt die `AWS::CloudFormation::Init`-Metadaten zur Verwaltung von Paketen, Dateien und Diensten auf Windows-Instances.

Für Unternehmensumgebungen CloudFormation ermöglicht es den Domänenbeitritt, die Windows Lizenzverwaltung über EC2-Lizenzmodelle 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	Description
AWS::EC2::Instance	Zum Starten von Windows EC2-Instances.
AWS::EC2::SecurityGroup	So definieren Sie Firewall-Regeln für Windows-Workloads.
AWS::AutoScaling::AutoScalingGroup (Gruppieren)	Für die Skalierung von Windows EC2-Instances.
AWS::EC2::LaunchTemplate	
AWS::DirectoryService::MicrosoftAD	Für die Bereitstellung von Microsoft Active Directory.

Ressourcentyp	Description
AWS::FSx::FileSystem	Für die Bereitstellung FSx für Windows Dateiserver.
AWS::RDS::DBInstance	Für die Bereitstellung von SQL Server auf Amazon RDS.
AWS::CloudFormation::Init	Wird in den 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-Kennwörtern.
AWS::SSM::Parameter	Zum sicheren Speichern von Konfigurationswerten.
AWS::IAM::InstanceProfile AWS::IAM::Role	Für die Erteilung von Berechtigungen für Anwendungen, die auf EC2-Instances laufen.

Bootstrapping-basierte Stacks Windows CloudFormation

In diesem Thema wird beschrieben, wie Sie einen Windows-Stack booten und Probleme bei der Stackerstellung beheben.

Themen

- [Benutzerdaten in EC2-Instances](#)
- [CloudFormation Hilfsskripte](#)
- [Beispiel für das Bootstrapping eines Windows-Stacks](#)
- [Backslashes in Windows-Dateipfaden entschärfen](#)
- [Dienste Windows-Verwaltung](#)
- [Beheben Sie Probleme bei der Stapelerstellung](#)

Benutzerdaten in EC2-Instances

Benutzerdaten sind ein Amazon EC2-Feature, mit dem Sie Skripte oder Konfigurationsinformationen an eine EC2-Instance übergeben können, wenn diese gestartet wird.

Für Windows EC2-Instances:

- Sie können entweder Batch-Skripte (mit `<script>`-Tags) oder PowerShell-Skripte (mit `<powershell>`-Tags) verwenden.
- Die Ausführung des Skripts wird von EC2Launch übernommen.

Important

Wenn Sie Ihr eigenes Windows AMI für die 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 Sie den EC2Launch v2-Agent, um Aufgaben während des Starts der EC2-Windows-Instance auszuführen](#) im Amazon EC2-Benutzerhandbuch.

Informationen dazu AWS Windows AMIs finden Sie in der [AWSWindowsAMI-Referenz](#).

CloudFormation Hilfsskripte

Hilfsskripte sind Dienstprogramme für die Konfiguration von Instances während des Bootstrapping-Prozesses. In Verbindung mit Amazon EC2-Benutzerdaten bieten sie leistungsstarke Konfigurationsoptionen.

CloudFormation stellt die folgenden Python-Hilfsskripten bereit, mit denen Sie Software installieren und Dienste auf einer Amazon EC2 EC2-Instance starten können, die Sie als Teil Ihres Stacks erstellen:

- `cfn-init`- Zum Abrufen und Interpretieren von Ressourcen-Metadaten, Installieren von Paketen, Erstellen von Dateien und Starten von Diensten.
- `cfn-signal`- Signalisieren Sie mit `CreationPolicy` damit Sie andere Ressourcen im Stack synchronisieren können, wenn die vorausgesetzte Ressource oder Anwendung bereit ist.

- `cfn-get-metadata`- Zum Abrufen von Metadaten für eine Ressource oder eines Pfads zu einem bestimmten Schlüssel.
- `cfn-hup`- Verwenden Sie diese Option, um nach Aktualisierungen von Metadaten zu suchen und benutzerdefinierte Hooks auszuführen, wenn Änderungen entdeckt 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 laufen in der Amazon EC2-Instance während der Stack-Erstellung.

Weitere Informationen finden Sie in der [Referenz zu CloudFormation Hilfsskripten](#) im CloudFormation Template Reference Guide.

Beispiel für das Bootstrapping eines Windows-Stacks

Schauen wir uns ein Beispiel für eine Windows Server-Vorlage an, die die folgenden Aktionen ausführt:

- Startet eine EC2-Instance mit dem Namen `TestInstance` von einem Windows Server 2022 AMI.
- Erzeugt eine einfache Testdatei, um zu überprüfen, ob `cfn-init` funktioniert.
- Konfiguriert `cfn-hup` für die laufende Konfigurationsverwaltung.
- Verwendet ein `CreationPolicy`, um sicherzustellen, dass die Instance den erfolgreichen Abschluss signalisiert.

Das Hilfsskript `cfn-init` wird verwendet, um jede dieser Aktionen auf der Grundlage der Informationen in der Ressource `AWS::CloudFormation::Init` in der Vorlage durchzuführen.

Der Abschnitt `AWS::CloudFormation::Init` heißt `TestInstance` und beginnt mit der folgenden Erklärung.

```
TestInstance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      configSets:
        default:
          - create_files
          - start_services
```

Danach wird der Abschnitt `files` 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}
```

Hier werden drei Dateien erstellt und im Verzeichnis `C:\cfn` auf der Serverinstance abgelegt:

- `test.txt`, eine einfache Testdatei, die überprüft, ob `cfn-init` korrekt funktioniert und Dateien mit dynamischem Inhalt erstellen kann.
- `cfn-hup.conf`, die Konfigurationsdatei für `cfn-hup` mit einem 2-Minuten-Prüfintervall.
- `cfn-auto-reloader.conf`, die Konfigurationsdatei für den Hook, der von `cfn-hup` verwendet wird, um eine Aktualisierung (Aufruf von `cfn-init`) zu initiieren, wenn sich die Metadaten in `AWS::CloudFormation::Init` ändern.

Als nächstes folgt der Abschnitt `start_services`, der die Windows-Dienste konfiguriert.

```
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 Dienst `cfn-hup` gestartet wird und automatisch neu startet, wenn die Konfigurationsdateien geändert werden. Der Dienst sucht nach Änderungen an den CloudFormation Metadaten und wird erneut ausgeführt `cfn-init`, wenn Aktualisierungen erkannt werden.

Als nächstes folgt der Abschnitt `Properties`.

```
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 Eigenschaft `UserData` ein PowerShell-Skript, das von EC2Launch ausgeführt wird und von `<powershell>`-Tags umgeben ist. 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 Instance ordnungsgemäß konfiguriert ist, bevor die Stack-Erstellung als abgeschlossen gilt.

Die Eigenschaft `ImageId` verwendet einen öffentlichen Parameter des Systems Manager Parameter Store, um automatisch die neueste Windows Server 2022 AMI-ID abzurufen. Dieser Ansatz macht

regionalspezifische AMI-Zuordnungen überflüssig und stellt sicher, dass Sie immer das aktuellste 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, vergewissern Sie sich, dass die Eigenschaft `SecurityGroupIds` auf eine Sicherheitsgruppe verweist, die den RDP-Zugriff erlaubt.

Das `CreationPolicy` wird als Teil der Ressourceneigenschaften deklariert und gibt eine Timeout-Periode an. Der Befehl `cfn-signal` in den Benutzerdaten signalisiert, wenn die Konfiguration der Instance abgeschlossen ist:

```
TestInstance:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT20M
  Properties:
    # ... other properties ...
```

Da der Bootstrapping-Prozess minimal ist und nur Dateien erstellt und Dienste gestartet werden, `CreationPolicy` wartet der 20 Minuten (PT20M), bevor das Timeout eintritt. Die Zeitüberschreitung wird im Format ISO 8601 Dauer angegeben. Beachten Sie, dass Windows-Instances in der Regel länger zum Starten brauchen als Linux-Instances. Testen Sie daher gründlich, um die besten Timeout-Werte für Ihre Bedürfnisse zu ermitteln.

Wenn alles gut geht, wird `CreationPolicy` erfolgreich abgeschlossen und Sie können auf die Instance Windows Server über ihre öffentliche IP-Adresse zugreifen. Sobald die Stack-Erstellung abgeschlossen ist, werden die Instanz-ID und die öffentliche IP-Adresse auf der Registerkarte Ausgaben der Konsole angezeigt. CloudFormation

```
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 korrekt funktioniert hat, indem Sie sich über RDP mit der Instance verbinden und überprüfen, ob die Datei `C:\cfn\test.txt` existiert und

den erwarteten Inhalt enthält. Weitere Informationen zum Verbinden mit Windows Instances finden Sie unter [Verbinden mit Ihrer Windows Instance über RDP](#) im Amazon EC2-Benutzerhandbuch.

Backslashes in Windows-Dateipfaden entschärfen

Denken Sie beim Verweisen auf Windows Pfade in CloudFormation Vorlagen immer daran, Backslashes (\) entsprechend dem verwendeten Vorlagenformat korrekt zu maskieren.

- Für JSON-Vorlagen müssen Sie doppelte Backslashes in Windows-Dateipfaden verwenden, da JSON den Backslash als Escape-Zeichen behandelt. Der erste Backslash umgeht den zweiten, was zu der Interpretation eines einzelnen, buchstäblichen Backslashes führt.

```
"commands" : {
  "1-extract" : {
    "command" : "C:\\\\SharePoint\\SharePointFoundation2010.exe /extract:C:\\\\SharePoint
\\SPF2010 /quiet /log:C:\\\\SharePoint\\SharePointFoundation2010-extract.log"
  }
}
```

- Für YAML-Vorlagen sind einfache 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 Windows-Verwaltung

Sie verwalten Windows-Dienste auf die gleiche Weise wie Linux-Dienste, nur dass Sie einen `windows`-Schlüssel anstelle von `sysvinit` verwenden. Das folgende Beispiel startet den Dienst `cfn-hup`, setzt ihn auf Automatisch und startet den Dienst neu, wenn `cfn-init` die Konfigurationsdateien `c:\cfn\cfn-hup.conf` oder `c:\cfn\hooks.d\cfn-auto-reloader.conf` ändert.

```
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 Stapelerstellung

Wenn Ihr Stack bei der Erstellung fehlschlägt, wird standardmäßig ein Rollback durchgeführt. 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 Fehlern bei der Bereitstellung von Ressourcen umgegangen werden soll](#). Damit können Sie sich bei Ihrer Instance anmelden und die Protokolldateien einsehen, um Probleme bei der Ausführung Ihrer Startskripte zu ermitteln.

Wichtige Protokolle, die sich Sie ansehen sollten:

- Das EC2-Konfigurationsprotokoll unter %ProgramData%\Amazon\EC2Launch\log\agent.log
- Das cfn-init-Protokoll bei C:\cfn\log\cfn-init.log (überprüfen Sie Exit-Codes und Fehlermeldungen für bestimmte Fehlerpunkte)

Weitere Protokolle finden Sie in den folgenden Themen im Amazon-EC2-Benutzerhandbuch:

- [EC2LaunchVerzeichnisstruktur](#)
- [EC2Launch v2 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	Description	Dokumentation
Benutzerdefinierte Ressourcen	Der <code>AWS::CloudFormation::CustomResource</code> Ressourcentyp ermöglicht es Ihnen, benutzerdefinierte Ressourcen zu erstellen, die bestimmte Bereitstellungsaufgaben ausführen oder Ressourcen einbeziehen können, die nicht als CloudFormation Ressourcentypen verfügbar sind.	Benutzerdefinierte Ressourcen
Makros	Der <code>AWS::CloudFormation::Macro</code> 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 bei der Erstellung oder Aktualisierung von Stacks durchführen.	Vorlagenmakros
Verschachtelte Stacks	Der <code>AWS::CloudFormation::Stack</code> Ressourcentyp ermöglicht es Ihnen, verschachtelte Stapel innerhalb Ihrer CloudFormation Vorlagen zu erstellen, um eine modularere und wiederverwendbarere Stack-Architektur zu erreichen.	Verschachtelte Stacks
StackSet	Der <code>AWS::CloudFormation::StackSet</code> Ressourcentyp erstellt oder aktualisiert	Verwaltung von Stacks mit StackSets

Ressourcentyp	Description	Dokumentation
	einen. CloudFormation StackSet Dabei handelt es sich um einen Container für Stacks, die in mehreren Regionen eingesetzt werden können. AWS-Konten	
Wartebedingung	Der Ressourcentyp <code>AWS::CloudFormation::WaitCondition</code> pausiert die Erstellung oder Aktualisierung von Stackn, bis eine bestimmte Bedingung erfüllt ist, beispielsweise der erfolgreiche Abschluss eines lang laufenden Prozesses oder die Verfügbarkeit externer Ressourcen.	Bedingungen warten
Warten Bedingung Griff	Der Ressourcentyp <code>AWS::CloudFormation::WaitConditionHandle</code> funktioniert zusammen mit dem Ressourcentyp <code>AWS::CloudFormation::WaitCondition</code> . Es bietet eine vordefinierte URL, die verwendet wird, um Signale zu senden, die anzeigen, dass eine bestimmte Bedingung erfüllt ist. Diese Signale ermöglichen es, den Prozess der Stack-Erstellung oder Aktualisierung fortzusetzen.	Bedingungen warten

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 immer noch alle Ihre 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.

In den folgenden Themen finden Sie Informationen über die Verwendung von benutzerdefinierten Ressourcen.

Themen

- [So funktionieren benutzerdefinierte Ressourcen](#)
- [Zeitüberschreitung bei der Antwort](#)
- [CloudFormation Referenz für benutzerdefinierte Ressourcenanfragen und Antworten](#)
- [Amazon SNS-gestützte benutzerdefinierte Ressourcen](#)
- [Lambda-gestützte benutzerdefinierte Ressourcen](#)

Note

Die CloudFormation Registrierung und die benutzerdefinierten Ressourcen bieten jeweils ihre eigenen Vorteile. Benutzerdefinierte Ressourcen bieten die folgenden Vorteile:

- Sie brauchen die Ressource nicht zu registrieren.
- Sie können eine ganze Ressource als Teil einer Vorlage einfügen, ohne sich zu registrieren.
- Unterstützt die Operationen, Create, Update und Delete .

Zu den Vorteilen, die registrierungsbasierte Ressourcen bieten, gehören die folgenden:

- 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

Der allgemeine Prozess zur Einrichtung einer neuen benutzerdefinierten Ressource umfasst die folgenden Schritte. An diesen Schritten sind zwei Rollen beteiligt: der Anbieter einer benutzerdefinierten Ressource der die benutzerdefinierte Ressource besitzt und der Vorlagenentwickler der eine Vorlage erstellt, die einen benutzerdefinierten Ressourcentyp enthält. Dies kann dieselbe Person sein, aber wenn nicht, sollte der Anbieter der benutzerdefinierten Ressourcen mit dem Entwickler der Vorlage zusammenarbeiten.

1. Der Anbieter für benutzerdefinierte Ressourcen 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-Thema oder die Lambda-Funktion muss sich in der gleichen Region befinden, in der der Stack erstellt wird.
3. Der benutzerdefinierte Ressourcenanbieter übergibt den Amazon SNS-Topic-ARN oder Lambda-Funktions-ARN an den Vorlagenentwickler.
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 von dem benutzerdefinierten Ressourcenanbieter definiert. Das Service-Token gibt den Amazon SNS-Topic-ARN oder Lambda-Funktions-ARN an 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 ist der Ablauf für die Erstellung eines Stapels aus der Vorlage zusammengefasst:

1. CloudFormation sendet eine Anfrage an das angegebene Service-Token. Die Anfrage enthält Informationen wie den Anfragetyp und eine vorsignierte Amazon S3 S3-Bucket-URL, an die die benutzerdefinierte Ressource Antworten sendet. Weitere Informationen zu den in der Anforderung enthaltenen Informationen finden Sie unter [CloudFormation Referenz für benutzerdefinierte Ressourcenanfragen und Antworten](#).

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",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "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 [CloudFormation Referenz für benutzerdefinierte Ressourcenanfragen und Antworten](#).

 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 über die Verwendung von NoEcho zur Maskierung sensibler Informationen finden Sie in der bewährten Methode [Keine Anmeldeinformationen in Vorlagen einbetten](#).

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",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MyTestResource",
  "PhysicalResourceId" : "TestResource1",
  "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 \(AWS PrivateLink\)](#).

Zeitüberschreitung bei der Antwort

Der Standard-Timeout für Ihre benutzerdefinierte Ressource beträgt 3600 Sekunden (1 Stunde). Wenn während dieser Zeit keine Antwort empfangen wird, schlägt der Stapelvorgang fehl.

Sie können den Timeout-Wert anpassen, je nachdem, wie lange die Antwort von der benutzerdefinierten Ressource voraussichtlich dauern wird. Wenn Sie beispielsweise eine benutzerdefinierte Ressource bereitstellen, die eine Lambda-Funktion aufruft, von der erwartet wird, dass sie innerhalb von fünf Minuten antwortet, können Sie in der Stack-Vorlage eine Zeitüberschreitung von fünf Minuten festlegen, indem Sie die Eigenschaft `ServiceTimeout` angeben. Weitere Informationen finden Sie unter [CloudFormation Referenz für benutzerdefinierte Ressourcenanfragen und Antworten](#). 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 anzusetzen. Um unerwartete Timeouts zu vermeiden, stellen Sie sicher, dass Ihre benutzerdefinierte Ressource genügend Zeit hat, um die erforderlichen Aktionen durchzuführen und eine Antwort zurückzugeben.

CloudFormation Referenz für benutzerdefinierte Ressourcenanfragen und Antworten

CloudFormation verwaltet benutzerdefinierte Ressourcen über ein Request-Response-Protokoll, das mit Ihrem Anbieter für benutzerdefinierte Ressourcen kommuniziert. Jede Anfrage umfasst einen Anforderungstyp (`CreateUpdate`, oder `Delete`) und folgt diesem allgemeinen Arbeitsablauf:

1. Ein Vorlagenentwickler definiert eine benutzerdefinierte Ressource mit einem `ServiceToken` und `ServiceTimeout` in der Vorlage und initiiert einen Stack-Vorgang.
2. CloudFormation sendet über SNS oder Lambda eine JSON-Anfrage an den Anbieter benutzerdefinierter Ressourcen.

3. Der benutzerdefinierte Ressourcenanbieter verarbeitet die Anfrage und gibt eine JSON-Antwort an eine vorsignierte Amazon S3 S3-Bucket-URL zurück, bevor der Timeout-Zeitraum abläuft.
4. CloudFormation liest die Antwort und fährt mit dem Stack-Vorgang fort. Wenn vor Ablauf des Timeouts keine Antwort eingeht, gilt die Anfrage als nicht erfolgreich und der Stack-Vorgang schlägt fehl.

Weitere Informationen finden Sie unter [So funktionieren benutzerdefinierte Ressourcen](#).

In diesem Abschnitt werden die Struktur, die Parameter und die erwarteten Antworten für jeden Anforderungstyp beschrieben.

 Note

Die Gesamtgröße des Antworttexts darf 4 096 Bytes nicht überschreiten.

Einrichtung der Vorlage

Bei der Definition einer benutzerdefinierten Ressource in einer Vorlage verwendet [AWS::CloudFormation::CustomResource](#) der Vorlagenentwickler die folgenden Eigenschaften:

ServiceToken

Ein Amazon SNS SNS-Themen-ARN oder eine Lambda-Funktions-ARN aus derselben Region wie der Stack.

Erforderlich: Ja

Typ: Zeichenfolge

ServiceTimeout

Die maximale Zeit in Sekunden, bis das Timeout eines benutzerdefinierten Ressourcenvorgangs abläuft. Es muss ein Wert zwischen 1 und 3600 sein. Standard: 3600 Sekunden (1 Stunde).

Required: No

Typ: Zeichenfolge

Zusätzliche Ressourceneigenschaften werden unterstützt. Ressourceneigenschaften werden wie `ResourceProperties` in der Anfrage enthalten sein. Der benutzerdefinierte Ressourcenanbieter muss ermitteln, welche Eigenschaften gültig sind und welche Werte zulässig sind.

Objekt anfordern

Create

Wenn der Vorlagenentwickler einen Stapel erstellt, der eine benutzerdefinierte Ressource enthält, CloudFormation sendet er eine Anfrage mit der `RequestType` Einstellung auf `Create`.

Erstellenanforderungen enthalten die folgenden Felder:

`RequestType`

`Create`.

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

`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

ResponseURL

Die Antwort-URL identifiziert einen vorsignierten S3-Bucket, der Antworten vom Anbieter für benutzerdefinierte Ressourcen empfängt. CloudFormation

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 CloudFormation -Vorlage.

Erforderlich: Ja

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

Beispiel

```
{
  "RequestType" : "Create",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
```

```
"ResponseURL" : "pre-signed-url-for-create-response",
"ResourceType" : "Custom::MyCustomResourceType",
"LogicalResourceId" : "resource-logical-id",
"ResourceProperties" : {
  "key1" : "string",
  "key2" : [ "list" ],
  "key3" : { "key4" : "map" }
}
```

Update

Wenn der Vorlagenentwickler Änderungen an den Eigenschaften einer benutzerdefinierten Ressource innerhalb der Vorlage vornimmt und den Stack aktualisiert, CloudFormation sendet er eine Anfrage an den Anbieter für benutzerdefinierte Ressourcen mit der `RequestType` Einstellung auf `Update`. Das bedeutet, dass Ihr benutzerdefinierter Ressourcencode keine Änderungen in den Ressourcen erkennen muss, da er weiß, dass sich seine Eigenschaften geändert haben, wenn der Anfragetyp `Update` ist.

Aktualisierungsanforderungen enthalten die folgenden Felder:

RequestType

`Update`.

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

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

`ResponseURL`

Die Antwort-URL identifiziert einen vorsignierten S3-Bucket, der Antworten vom Anbieter für benutzerdefinierte Ressourcen empfängt. CloudFormation

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: `_@-`. Bei einer Aktualisierung kann der Typ nicht geändert werden.

Erforderlich: Ja

Typ: Zeichenfolge

`LogicalResourceId`

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der CloudFormation -Vorlage.

Erforderlich: Ja

Typ: Zeichenfolge

`PhysicalResourceId`

Eine benutzerdefinierte, vom Ressourcenanbieter definierte physische ID, die für diesen Anbieter eindeutig ist.

Erforderlich: Ja

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. Gibt die bisherigen Ressourceneigenschaftswerte an, die der Vorlagenentwickler zu einem früheren Zeitpunkt in der CloudFormation - Vorlage deklariert hatte.

Erforderlich: Ja

Typ: JSON-Objekt

Beispiel

```
{
  "RequestType" : "Update",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-update-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "PhysicalResourceId" : "provider-defined-physical-id",
  "ResourceProperties" : {
    "key1" : "new-string",
    "key2" : [ "new-list" ],
    "key3" : { "key4" : "new-map" }
  },
  "OldResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Delete

Wenn der Vorlagenentwickler den Stack löscht oder die benutzerdefinierte Ressource aus der Vorlage entfernt und dann den Stapel aktualisiert, CloudFormation sendet er eine Anfrage mit `RequestType` der Einstellung auf `Delete`.

Anforderungen zum Löschen enthalten die folgenden Felder:

`RequestType`

`Delete`.

Erforderlich: Ja

Typ: Zeichenfolge

`RequestId`

Eine eindeutige ID für die Anforderung.

Erforderlich: Ja

Typ: Zeichenfolge

`StackId`

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält.

Erforderlich: Ja

Typ: Zeichenfolge

`ResponseURL`

Die Antwort-URL identifiziert einen vorsignierten S3-Bucket, der Antworten vom Anbieter für benutzerdefinierte Ressourcen empfängt. CloudFormation

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 CloudFormation -Vorlage.

Erforderlich: Ja

Typ: Zeichenfolge

PhysicalResourceId

Eine benutzerdefinierte, vom Ressourcenanbieter definierte physische ID, die für diesen Anbieter eindeutig ist.

Erforderlich: Ja

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

Beispiel

```
{
  "RequestType" : "Delete",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-delete-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "PhysicalResourceId" : "provider-defined-physical-id",
  "ResourceProperties" : {
```



```
"key1" : "string",
"key2" : [ "list" ],
"key3" : { "key4" : "map" }
}
}
```

Antwortobjekt

Der benutzerdefinierte Ressourcenanbieter sendet für alle Anforderungstypen eine Antwort an die vorsignierte URL. Wenn der benutzerdefinierte Ressourcenanbieter keine Antwort sendet, CloudFormation wartet er, bis das Zeitlimit für den Vorgang überschritten wird.

Die Antwort muss ein JSON-Objekt mit den folgenden Feldern sein:

Status

Es muss entweder SUCCESS oder FAILED sein.

Erforderlich: Ja

Typ: Zeichenfolge

RequestId

Eine eindeutige ID für die Anforderung. Kopieren Sie diesen Wert genau so, wie er in der Anfrage erscheint.

Erforderlich: Ja

Typ: Zeichenfolge

StackId

Der Amazon-Ressourcenname (ARN), der den Stack identifiziert, der die benutzerdefinierte Ressource enthält. Kopieren Sie diesen Wert genau so, wie er in der Anfrage erscheint.

Erforderlich: Ja

Typ: Zeichenfolge

LogicalResourceId

Der vom Entwickler der Vorlagen gewählte Name (logische ID) der benutzerdefinierten Ressource in der CloudFormation -Vorlage. Kopieren Sie diesen Wert genau so, wie er in der Anfrage erscheint.

Erforderlich: Ja

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.

Beim Aktualisieren von benutzerdefinierten Ressourcen `PhysicalResourceId` bestimmt der zurückgegebene Wert für das Aktualisierungsverhalten. Wenn der Wert gleich bleibt, wird dies CloudFormation als normales Update betrachtet. Wenn sich der Wert ändert, CloudFormation interpretiert das Update als Ersatz und sendet eine Löschanforderung an die alte Ressource. Weitere Informationen finden Sie unter [AWS::CloudFormation::CustomResource](#).

Erforderlich: Ja

Typ: Zeichenfolge

Reason

Beschreibt den Grund für eine Reaktion auf einen Fehler.

Erforderlich, wenn `Status` ein `FAILED` ist. Er ist andernfalls optional.

Required: Conditional

Typ: Zeichenfolge

NoEcho

Gibt an, ob die Ausgabe der benutzerdefinierten Ressource maskiert wird, wenn sie mithilfe der `Fn::GetAtt`-Funktion abgerufen wird. Wenn diese Option auf `true` gesetzt ist, werden alle zurückgegebenen Werte mit Sternchen (`*****`) maskiert außer denen, die im *Metadata* Abschnitt der Vorlage gespeichert sind. CloudFormation transformiert, ändert oder bearbeitet keine Informationen im *Metadata* Abschnitt. Der Standardwert ist `false`.

Weitere Informationen über die Verwendung von `NoEcho` zur Maskierung sensibler Informationen finden Sie in der bewährten Methode [Keine Anmeldeinformationen in Vorlagen einbetten](#).

Nur für Update Antworten `Create` und verfügbar. Wird für `Delete` Antworten nicht unterstützt.

Required: No

Typ: Boolesch

Data

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.

Nur für Create Update Antworten verfügbar. Wird für Delete Antworten nicht unterstützt.

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 diese Oberflächeneigenschaftswerte (z. B. `DescribeStackEvents`) sichtbar.

Required: No

Typ: JSON-Objekt

Beispiele für erfolgreiche Antworten

Create und Update Antwort

```
{
  "Status": "SUCCESS",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id",
  "NoEcho": true,
  "Data": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

Delete-Antwort

```
{
```

```
"Status": "SUCCESS",
"RequestId": "unique-request-id",
"StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
"LogicalResourceId": "resource-logical-id",
"PhysicalResourceId": "provider-defined-physical-id"
}
```

Beispiel für eine gescheiterte Antwort

```
{
  "Status": "FAILED",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id",
  "Reason": "Required failure reason string"
}
```

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 Abfolge von Ereignissen und Nachrichten kennen, die als Ergebnis der Erstellung, Aktualisierung und Löschung eines benutzerdefinierten Ressourcenstapels gesendet und empfangen werden.

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 Amazon EC2 EC2-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 zu verarbeiten und die erforderlichen Aktionen durchzuführen, müssen Sie einen unterstützten Endpunkt bei dem SNS-Thema abonniert haben.

Eine Einführung in benutzerdefinierte Ressourcen und ihre Funktionsweise finden Sie unter [So funktionieren benutzerdefinierte Ressourcen](#). Informationen über Amazon SNS und dessen Funktionsweise finden Sie im [Amazon-Simple-Notification-Service-Entwicklerleitfaden](#).

Verwendung von Amazon SNS zur Erstellung 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 Stack, der eine benutzerdefinierte Ressource enthält.

In der Vorlage unten verwenden wir den benutzerdefinierten Ressourcentyp `Custom::SeleniumTester` für die benutzerdefinierte Ressource mit der logischen 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-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 zurückgegeben. CloudFormation 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",
    "RequestId" : "unique-request-id",
    "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",

```

```

"ResponseURL" : "http://pre-signed-S3-url-for-response",
"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 über das Anfrageobjekt für Create-Anfragen finden Sie im Thema [Referenz für Anfrage und Antwort](#) .

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 FAILED zu senden.

Je nach Antworttyp werden unterschiedliche Antwortfelder von CloudFormation erwartet. Informationen über die Antwortfelder für einen bestimmten Anfragetyp finden Sie in der Dokumentation für diesen Anfragetyp im Abschnitt [Referenz für Anfrage und Antwort](#) .

Als Antwort auf eine Erstellungs- oder Aktualisierungsanfrage kann der benutzerdefinierte Ressourcenanbieter Datenelemente in das 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",
    "RequestId" : "unique-request-id",
    "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
    "LogicalResourceId" : "MySeleniumTester",
    "PhysicalResourceId" : "Tester1",
    "Data" : {
        "resultsPage" : "http://www.myexampledomain/test-results/guid",

```

```

    "lastUpdate" : "2012-11-14T03:30Z"
  }
}

```

Ausführliche Informationen über das Antwortobjekt für Create-Anfragen finden Sie im Thema [Referenz für Anfrage und Antwort](#) .

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 des Aktualisierungsverhaltens 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. Wenn 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 gezeigt [Schritt 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 Aufruf 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",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "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" ]
  },
  "OldResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
  }
}
```

Ausführliche Informationen über das Anfrageobjekt für Update-Anfragen finden Sie im Thema [Referenz für Anfrage und Antwort](#) .

3. Der benutzerdefinierte Ressourcenanbieter verarbeitet die von CloudFormation gesendeten Daten. 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",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester2"
}
```

Ausführliche Informationen über das Antwortobjekt für Update-Anfragen finden Sie im Thema [Referenz für Anfrage und Antwort](#).

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",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "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 über das Anfrageobjekt für Delete-Anfragen finden Sie im Thema [Referenz für Anfrage und Antwort](#) .

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 Löschanfrage antworten.

Das folgende Beispiel zeigt die custom resource provider-Antwort auf eine Delete-Anforderung.

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1"
```

```
}
```

Ausführliche Informationen über das Antwortobjekt für Delete-Anfragen finden Sie im Thema [Referenz für Anfrage und Antwort](#) .

Die StackId, RequestId und LogicalResourceId-Felder müssen wörtlich aus der Anforderung kopiert werden.

4. CloudFormation deklariert den Stack-Status als DELETE_COMPLETE oderDELETE_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 ihre Funktionsweise finden Sie unter [So funktionieren benutzerdefinierte 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 [Abrufen eines Klartextwertes aus dem Systems Manager Parameter Store](#).

Themen

- [-Übersicht](#)
- [Beispielvorlage](#)
- [Exemplarische Vorgehensweise für eine Vorlage](#)
- [Voraussetzungen](#)
- [Starten des Stapels](#)
- [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 enthält die folgenden Ressourcen:

- 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 Service, 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 über Lambda-Preise finden Sie unter [AWS Lambda Preisgestaltung](#).

Beispielvorlage

Sie können die Lambda-gestützte Beispielvorlage für benutzerdefinierte Ressourcen mit dem Verzögerungsmechanismus unten sehen:

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(\"ServiceTimeout", 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"] },
        "ServiceTimeout": 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("ServiceTimeout", 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
    ServiceTimeout: 60
Outputs:
  TimeWaited:
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited
    Export:
      Name: TimeWaited
  WaiterId:
    Value: !GetAtt CFNWaiterCustomResource.Id
    Export:
      Name: WaiterId

```

Exemplarische Vorgehensweise für eine Vorlage

Die folgenden Schnipsel erläutern relevante Teile der Beispielvorlage, damit Sie verstehen, wie die Lambda-Funktion mit einer benutzerdefinierten Ressource verknüpft ist und die Ausgabe verstehen.

[AWS::Lambda::Function](#)Ressource CFNWaiter

-Bei der `AWS::Lambda::Function` Ressource gilt der Quellcode der Funktion, der Handler-Name, die Laufzeitumgebung sowie der Amazon Resource Name (ARN) der Ausführungsrolle.

Die Handler Eigenschaft ist auf `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 wird als `python3.9` angegeben, da die Quelldatei ein Python-Code ist.

Der Wert `Timeout` ist auf 900 Sekunden eingestellt.

Die Eigenschaft `Role` verwendet die Funktion `Fn::GetAtt`, um den ARN der Ausführungsrolle `LambdaExecutionRole` zu erhalten, die in der Ressource `AWS::IAM::Role` in der Vorlage deklariert ist.

Die Eigenschaft `Code` definiert den Funktionscode inline mit einer Python-Funktion. Die Python-Funktion in der Beispielvorlage macht Folgendes:

- Erstellen Sie eine eindeutige ID mit der UUID
- Prüfen Sie, ob die Anfrage eine Erstellungs- oder Aktualisierungsanfrage ist
- Ruhezustand für die für `ServiceTimeout` angegebene Dauer während `Create` oder `Update` Anfragen
- Geben Sie 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(\"ServiceTimeout", 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("ServiceTimeout", 0))
                sleep(wait_seconds)
            response = {
                "TimeWaited": wait_seconds,
                "Id": id
            }
            cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
"Waiter-"+id)
...

```

AWS::IAM::RoleRessource LambdaExecutionRole

Die Ressource `AWS::IAM::Role` erstellt eine Ausführungsrolle für die Lambda-Funktion, die eine "assume role"-Richtlinie enthält, die es Lambda erlaubt, sie zu verwenden. Es 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::CustomResource](#)Ressource CFNWaiterCustomResource

Die benutzerdefinierte Ressource verlinkt auf die Lambda-Funktion mit ihrem ARN über !GetAtt CFNWaiter.Arn. Es wird eine Wartezeit von 60 Sekunden für Erstellungs- und Aktualisierungsvorgänge implementiert, wie in ServiceTimeout festgelegt. Die Ressource wird nur dann für einen Aktualisierungsvorgang aufgerufen, wenn die Eigenschaften geändert werden.

JSON

```

...
  "CFNWaiterCustomResource": {
    "Type": "AWS::CloudFormation::CustomResource",
    "Properties": {
      "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
      "ServiceTimeout": 60
    }
  },
...

```

YAML

```

...
  CFNWaiterCustomResource:

```

```
Type: AWS::CloudFormation::CustomResource
Properties:
  ServiceToken: !GetAtt CFNWaiter.Arn
  ServiceTimeout: 60
...
```

Outputs

Die Outputs dieser Vorlage sind die `TimeWaited` und die `WaiterId`. Der `TimeWaited`-Wert verwendet eine `Fn::GetAtt`-Funktion, um die Zeit anzugeben, die die Kellnerressource tatsächlich gewartet hat. Die `WaiterId`-Funktion verwendet eine `Fn::GetAtt`-Funktion, um die eindeutige ID bereitzustellen, die erzeugt und mit der Ausführung verknüpft 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

Starten des Stapels

So erstellen Sie den Stack

1. Suchen Sie die gewünschte Vorlage (YAML oder JSON) aus dem Abschnitt [Beispielvorlage](#) und speichern Sie sie auf Ihrem Rechner unter dem Namen `samplelambdabackedcustomresource.template`.
2. Öffnen Sie die CloudFormation Konsole unter. <https://console.aws.amazon.com/cloudformation/>
3. Wählen Sie auf der Seite Stack oben rechts Stack erstellen und dann Mit neuen Ressourcen (Standard).
4. Für Voraussetzung - Vorlage vorbereitenwählen Sie Vorhandene Vorlage auswählen.
5. Für Vorlage angeben, wählen Sie Eine Vorlagendatei hochladenund dann Datei auswählen.
6. Wählen Sie die Vorlagendatei `samplelambdabackedcustomresource.template`, die Sie zuvor gespeichert haben.
7. Wählen Sie Weiter aus.
8. Für Stapelname, geben Sie **SampleCustomResourceStack** ein und wählen Weiter.
9. Für diese Anleitung müssen Sie keine Tags hinzufügen oder erweiterte Einstellungen angeben, wählen Sie daher Weiter aus.
10. Vergewissern Sie sich, dass der Name des Stapels korrekt ist, und wählen Sie dann Erstellen.

Es kann mehrere Minuten dauern CloudFormation , bis Ihr Stack erstellt ist. 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 war, wurden alle Ressourcen im Stack, wie 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.

Da Sie nun wissen, wie Sie Lambda-gestützte benutzerdefinierte Ressourcen erstellen und verwenden können, können Sie die Beispielvorlage und den Code aus dieser Anleitung 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 Eigenschaft `ZipFile` 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 Modul `cfn-response` ist eine Bibliothek, die das Senden von Antworten an die benutzerdefinierte Ressource, die Ihre Lambda-Funktion aufgerufen hat, vereinfacht. Das Modul verfügt über eine `send`-Methode, die ein [Antwortobjekt](#) über eine von Amazon S3 vorgegebene 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 -Entwicklerleitfaden.

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 über die Verwendung von NoEcho zur Maskierung sensibler Informationen finden Sie in der bewährten Methode [Keine Anmeldeinformationen in Vorlagen einbetten](#).

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 Node.js Quellcode](#)
- [Node.js Quellcode](#)
- [Python Quellcode](#)

Asynchroner Node.js Quellcode

Im Folgenden finden Sie den Quellcode des Antwortmoduls für die Node.js-Funktionen, 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("statusCode: ", response.statusCode);
            resolve();
        });
        request.on("error", function(error) {
            reject(error);
        });
        request.write(responseBody);
        request.end();
    });
};
```

```

    };

    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=*****');
}

```

Node.js Quellcode

Im Folgenden finden Sie den Quellcode des Antwortmoduls für die Node.js-Funktionen, wenn 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)
```

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::Include transform](#) ermöglicht es Ihnen, Boilerplate-Vorlagenschnipsel in Ihre Vorlagen einzufügen.
- [AWS::Serverless transform 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 zur Ersetzung von Strings](#)

- [Fehlersuche in der verarbeiteten Vorlage](#)

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 `AWS::Transform` 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, die die Verarbeitung der Vorlage durchführt. 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

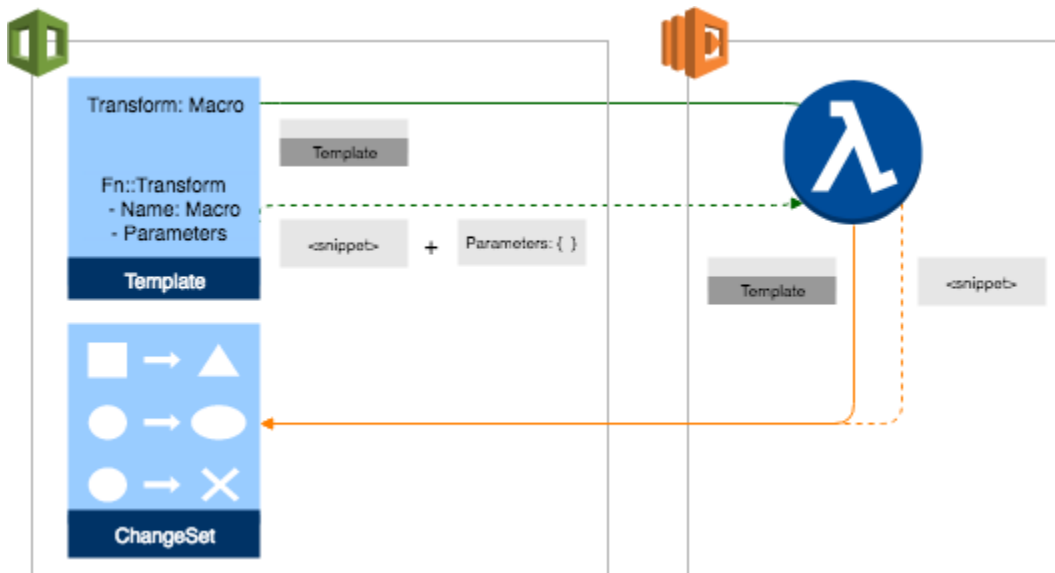
den ARN der Lambda-Funktion, die für dieses Makro aufgerufen werden soll, und zusätzliche optionale Eigenschaften, die beim Debuggen helfen. Um diese Ressource in einem Konto zu erstellen, erstellen Sie eine Vorlage, die die `AWS::CloudFormation::Macro` Ressource enthält, und erstellen Sie dann entweder einen Stack oder ein Stack-Set mit selbstverwalteten Berechtigungen aus der Vorlage. CloudFormation StackSets unterstützt 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 einen Teil einer Vorlage zu bearbeiten, referenzieren Sie das Makro in einer `Fn::Transform`-Funktion, die sich relativ zu dem Inhalt der Vorlage befindet, den Sie umwandeln 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 Sie Stapel direkt erstellen

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, 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 zuvor die vorgeschlagenen Änderungen in einem Änderungssatz zu prüfen, geben Sie bei einer `CreateStack`- oder `UpdateStack`-Anfrage die `CAPABILITY_AUTO_EXPAND`-Fähigkeit 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 [CreateStack](#) oder [UpdateStack](#) in 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 `package` Befehle und verwenden. `deploy` AWS CLI 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

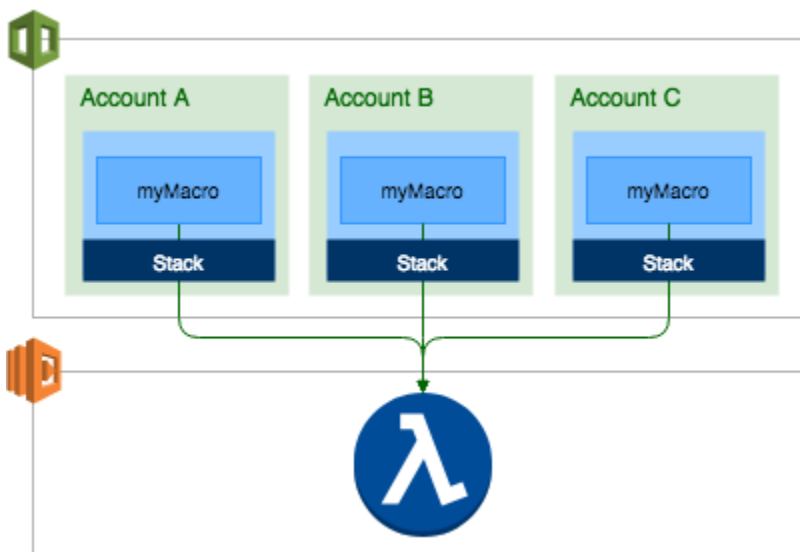
Wenn Sie mit Makros arbeiten, beachten Sie bitte die folgenden Hinweise und Einschränkungen:

- 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 [Makro-Bewertungsreihenfolge](#).

- 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 innerhalb von 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 Stack-Sets mit vom Service verwalteten Berechtigungen anhand von Vorlagen, die auf Makros verweisen. CloudFormation

Umfang und Berechtigungen des Makrokontos

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 [Verwalten von Berechtigungen in AWS Lambda](#) im AWS Lambda Entwicklerhandbuch und [Aktionen, Ressourcen und Bedingungsschlüssel für AWS Lambda](#) in der 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.

Ereignis-Mapping

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 [Umfang der Makro-Vorlage](#).
- `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.

Beispiel:

```
Error creating change set: Transform
      AWS-Konto account
      number::macro name failed with:
      error message string.
```

Erstellen Sie eine Makrodefinition

Um eine Makrodefinition zu erstellen CloudFormation

1. [Erstellen Sie eine Lambda-Funktion](#) die die Verarbeitung der Vorlageninhalte übernimmt. Es kann jeden Teil einer Vorlage verarbeiten, bis hin zur gesamten Vorlage.

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 die Fehlersuche zu erleichtern, können Sie beim Erstellen des Ressourcentyps `AWS::CloudFormation::Macro` für Ihr Makro auch die Eigenschaften `LogGroupName` und `LogRoleArn` angeben. 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 einen Stapel](#) unter Verwendung der Vorlage mit dem Makro in dem Konto, in dem Sie es verwenden möchten. Oder [erstellen Sie ein Stack-Set mit selbstverwalteten Berechtigungen](#) unter Verwendung der Vorlage mit dem Makro im Administratorkonto und erstellen Sie dann Stack-Instances in den Zielkonten.
5. Nachdem CloudFormation die Stacks, die die Makrodefinition enthalten, erfolgreich erstellt wurden, kann das Makro innerhalb dieser Konten verwendet werden. Sie verwenden ein Makro, indem Sie es in einer Vorlage an der Stelle referenzieren, die für die zu bearbeitenden Vorlageninhalte relevant ist.

Umfang der Makro-Vorlage

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

```

Makro-Bewertungsreihenfolge

Sie können in einer bestimmten Vorlage auf mehrere Makros verweisen, einschließlich Transformationen, die von bereitgestellt werden CloudFormation, z. B. auf und.

[AWS::IncludeAWS::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 zur Ersetzung von Strings

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 `LogGroupName`- oder `LogRoleARN`-Eigenschaft für die Fehlerprotokollierung eingefügt).

In diesem Beispiel nehmen wir an, dass der mit dieser Vorlage erstellte Stapel den Namen `JavaMacroFunc` trägt. Da die Makroeigenschaft `Name` auf den Stapelnamen gesetzt ist, heißt das resultierende Makro auch `JavaMacroFunc`.

```
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'
```

Das Makro verwenden

Um unser Makro zu verwenden, fügen wir es in eine Vorlage ein, indem wir die intrinsische Funktion `Fn::Transform` verwenden.

CloudFormation ruft unser Beispielmakro 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 `JavaMacroFunc` angegeben, das aus dem vorherigen Makrodefinitionsbeispiel 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.

Code der Lambda-Funktion

Im Folgenden finden Sie den eigentlichen Code für die Lambda-Funktion, die dem Beispielmakro `JavaMacroFunc` 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 Zuordnung 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.

Ergebnis der verarbeiteten 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 intrinsische Funktionsaufruf `Fn::Transform`, der auf das Makro `JavaMacroFunc` verwies, 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"
    }
  }
}
```

Fehlersuche in der verarbeiteten Vorlage

Wenn Sie ein Makro verwenden, finden Sie die verarbeitete Vorlage in der CloudFormation Konsole.

Das Stadium einer Vorlage zeigt ihren Bearbeitungsstatus 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 Fehlersuche 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-template](#)Befehl 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 [Aufteilung einer Vorlage in wiederverwendbare Teile mit verschachtelten Stapeln](#).

- 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.

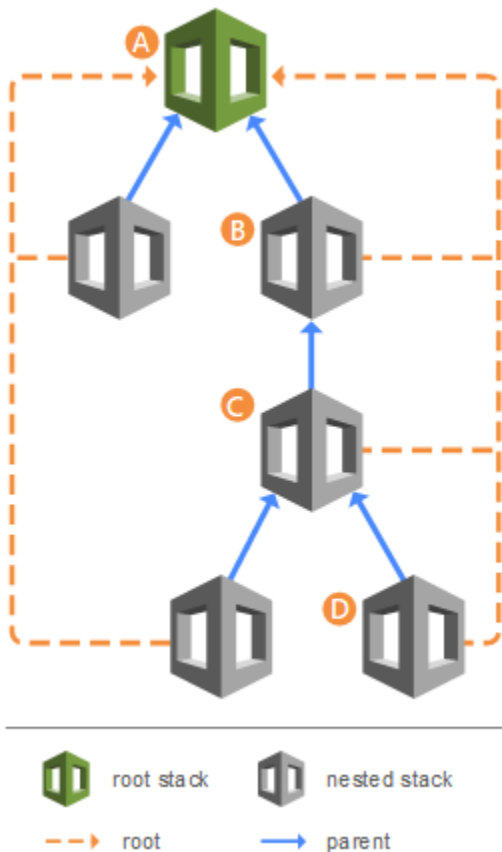
Aufteilung einer Vorlage in wiederverwendbare Teile mit verschachtelten Stapeln

Wenn Ihre Infrastruktur wächst, müssen Sie möglicherweise immer wieder identische Ressourcenkonfigurationen für mehrere Vorlagen erstellen. Um diese Redundanz zu vermeiden, können Sie diese gemeinsamen Konfigurationen in spezielle Vorlagen aufteilen. Dann können Sie die Ressource [AWS::CloudFormation::Stack](#) in anderen Vorlagen verwenden, um auf diese speziellen Vorlagen zu verweisen und so verschachtelte Stapel zu erstellen.

Nehmen wir zum Beispiel 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 diese Vorlage in anderen Vorlagen referenzieren, die dieselbe Load Balancer-Konfiguration benötigen.

Verschachtelte Stapel können ihrerseits weitere verschachtelte Stapel enthalten, so dass eine Hierarchie von Stapeln entsteht, wie in der folgenden Abbildung dargestellt. Der-WurzelStack ist der oberste Stack, zu dem letztlich alle verschachtelten Stack gehören. Jeder verschachtelte Stapel hat einen unmittelbaren übergeordneten Stapel. Für die erste Ebene von verschachtelten Stapeln ist der Stammstapel auch der übergeordnete Stapel.

- 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

- [Vorher-Nachher-Beispiel für die Aufteilung einer Vorlage](#)
- [Beispiel für eine verschachtelte Stack-Architektur](#)
- [Ausführen von Stapeloperationen auf verschachtelten Stapeln](#)
- [Ähnliche Informationen](#)

Vorher-Nachher-Beispiel für die Aufteilung einer Vorlage

Dieses Beispiel zeigt, wie Sie eine einzelne, große CloudFormation Vorlage mithilfe verschachtelter Vorlagen in ein strukturierteres und wiederverwendbares Design umstrukturieren können. Die Vorlage "Vor der Verschachtelung von Stapeln" zeigt zunächst alle in einer Datei definierten Ressourcen an. Das kann unübersichtlich und schwer zu verwalten werden, wenn die Anzahl der Ressourcen wächst. Die Vorlage "Nach Verschachtelung der Stapel" teilt die Ressourcen in kleinere, separate Vorlagen auf. Jeder verschachtelte Stapel verwaltet einen bestimmten Satz zusammengehöriger Ressourcen, wodurch die Gesamtstruktur übersichtlicher und leichter zu pflegen ist.

Vor der Verschachtelung von Stapeln	Nach der Verschachtelung von Stapeln
<pre> AWSTemplateFormatVersion: 2010-09-09 Parameters: InstanceType: Type: String Default: t2.micro Description: The EC2 instance type Environment: Type: String Default: Production Description: The deployment environment Resources: MyEC2Instance: Type: AWS::EC2::Instance Properties: ImageId: ami-1234567890abcdef0 InstanceType: !Ref InstanceType MyS3Bucket: Type: AWS::S3::Bucket </pre>	<pre> AWSTemplateFormatVersion: 2010-09-09 Resources: MyFirstNestedStack: Type: AWS::CloudFormation::Stack Properties: TemplateURL: https://s3.amazonaws.com/ amzn-s3-demo-bucket /first-nested-stack.yaml Parameters: # 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 </pre>

Beispiel für eine verschachtelte Stack-Architektur

Dieser Abschnitt demonstriert eine verschachtelte Stack-Architektur, die aus einem Top-Level-Stack besteht, der auf einen verschachtelten Stack verweist. Der verschachtelte Stack setzt eine Node.js Lambda-Funktion ein, erhält einen Parameterwert vom Stack der obersten Ebene und gibt eine Ausgabe zurück, die über den Stack der obersten Ebene exponiert 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: Paketieren und verteilen Sie die Vorlagen](#)

Schritt 1: Erstellen Sie eine Vorlage für den verschachtelten Stack auf Ihrem lokalen System

Das folgende Beispiel zeigt das Format der verschachtelten Stapelvorlage.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Nested stack template for Lambda function deployment
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 Stack-Vorlage der obersten Ebene und der Ressource [AWS::CloudFormation::Stack](#), 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: Paketieren und verteilen Sie die 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 zu Amazon S3 (einschließlich TemplateURL) und erzeugt 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 Befehl [package](#) verwenden, um die verschachtelte Vorlage in einen Amazon 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 erzeugt eine neue Vorlage unter dem durch `--output-template-file` angegebenen Pfad. Es ersetzt die Referenz `TemplateURL` durch den Amazon S3-Speicherort, 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 Befehl `package` ausgeführt haben, können Sie die verarbeitete Vorlage mit dem Befehl [deploy](#) bereitstellen. Bei verschachtelten Stapeln, die IAM-Ressourcen enthalten, müssen Sie die IAM-Fähigkeiten bestätigen, indem Sie die Option `--capabilities` einfügen.

```
aws cloudformation deploy \  
  --template-file packaged-template.yaml \  
  --stack-name stack-name \  
  --capabilities CAPABILITY_NAMED_IAM
```

Ausführen von Stapeloperationen auf verschachtelten Stapeln

Wenn Sie mit verschachtelten Stapeln arbeiten, müssen Sie diese bei Operationen sorgfältig behandeln. Bestimmte Stack-Operationen, wie beispielsweise Stack-Aktualisierungen, sollten vom Root-Stack aus initiiert werden und nicht direkt auf den verschachtelten Stacks ausgeführt werden. Wenn Sie einen Stammstapel aktualisieren, werden nur verschachtelte Stapel mit Vorlagenänderungen aktualisiert.

Außerdem kann das Vorhandensein von verschachtelten Stapeln die Operationen auf dem Stammstapel beeinflussen. Wenn zum Beispiel ein verschachtelter Stack im Zustand `UPDATE_ROLLBACK_IN_PROGRESS` stecken bleibt, wartet der Root-Stack, bis dieser verschachtelte Stack sein Rollback beendet hat, bevor er fortfährt. Bevor Sie mit den Aktualisierungsvorgängen fortfahren, vergewissern Sie sich, dass Sie über IAM-Berechtigungen verfügen, um eine Stack-Aktualisierung abzubrechen, falls diese zurückgesetzt wird. Weitere Informationen finden Sie unter [Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management](#).

Verwenden Sie die folgenden Verfahren, um den Stammstapel und die verschachtelten Stapel zu finden.

So zeigen Sie den Root-Stack eines verschachtelten Stacks an

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie auf der Seite Stapel den Namen des verschachtelten Stapels, dessen Stammstapel Sie sehen möchten.

Verschachtelte Stapel zeigen NESTED über ihrem Stapelnamen an.

3. Wählen Sie auf der Registerkarte Stackinfo im Abschnitt Übersicht den Stacknamen aus, der als Wurzelstackaufgeführt ist.

So zeigen Sie die verschachtelten Stacks eines Root-Stacks an

1. Wählen Sie im Stammstapel, dessen verschachtelte Stapel Sie anzeigen möchten, die Registerkarte Ressourcen .
2. Suchen Sie in der Spalte Typ nach Ressourcen des Typs `AWS::CloudFormation::Stack`

Ähnliche Informationen

- [Verschachteln eines vorhandenen Stacks](#)
- [Verstehen des Aktualisierungsverhaltens von Stack-Ressourcen](#)
- [Weiteres Rollback von fehlgeschlagenen verschachtelten Stack-Updates](#)
- [Rollback-Fehler bei verschachtelten Stapeln](#)

Wartebedingungen in einer CloudFormation Vorlage erstellen

In diesem Thema wird erklärt, wie Sie eine Wartebedingung in einer Vorlage erstellen, um die Erstellung von Stapelressourcen 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:

- Sie erstellt eine Wartebedingung wie jede andere Ressource 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 er die erforderliche Anzahl von Erfolgssignalen erhält, bevor die Timeout-Periode abläuft:
 - Der Status der Wartebedingung ändert sich zu `CREATE_COMPLETE`.
 - Die Erstellung von Stapeln geht weiter
- Wenn die Zeitüberschreitung abläuft oder ein Fehlersignal empfangen wird:
 - Der Status der Wartebedingung ändert sich zu `CREATE_FAILED`.
 - Stapel rollt zurück

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 [Kontrolle des Zugriffs von VPC-Endpunkten mit Bucket-Richtlinien](#).

Themen

- [Erstellen einer Wartebedingung in Ihrer Vorlage](#)
- [Syntax des Wartezustandssignals](#)
- [Zugriff auf Signaldaten](#)

Erstellen einer Wartebedingung in Ihrer Vorlage

1. Warten Bedingung Griff

Sie beginnen mit der Definition einer [AWS::CloudFormation::WaitConditionHandle](#)-Ressource in der Vorlage des Stacks. Diese Ressource generiert die vordefinierte URL, die zum Senden von Signalen benötigt wird. So können Sie ein Signal senden, ohne Ihre AWS -Anmeldedaten angeben zu müssen. Beispiel:

Resources:

MyWaitHandle:

Type: AWS::CloudFormation::WaitConditionHandle

2. Wartebedingung

Als nächstes definieren Sie eine [AWS::CloudFormation::WaitCondition](#)-Ressource in der Vorlage des Stacks. Die Grundstruktur eines `AWS::CloudFormation::WaitCondition` sieht wie folgt aus:

MyWaitCondition:

Type: `AWS::CloudFormation::WaitCondition`

Properties:

Handle: *String*

Timeout: *String*

Count: *Integer*

Die Ressource `AWS::CloudFormation::WaitCondition` hat zwei erforderliche Eigenschaften und eine optionale Eigenschaft.

- **Handle** (erforderlich)- Ein Verweis auf ein in der Vorlage deklariertes `WaitConditionHandle` .
- **Timeout**(erforderlich) — Die Anzahl der Sekunden, die auf den CloudFormation Empfang der erforderlichen Anzahl von Signalen gewartet werden müssen. `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.

Normalerweise möchten Sie, dass eine Wartebedingung sofort 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 demonstriert eine Wartebedingung, die:

- Beginnt nach der erfolgreichen Erstellung der Ressource `MyEC2Instance` .
- Verwendet die Ressource `MyWaitHandle` als `WaitConditionHandle`.
- Hat eine Zeitüberschreitung von 4500 Sekunden
- Hat den Standardwert `Count` von 1 (da keine Eigenschaft `Count` angegeben ist).

MyWaitCondition:

Type: `AWS::CloudFormation::WaitCondition`

`DependsOn`: *MyEC2Instance*

Properties:

```
Handle: !Ref MyWaitHandle  
Timeout: '4500'
```

3. Senden eines Signals

Um Erfolg oder Misserfolg zu signalisieren CloudFormation, führen Sie in der Regel Code oder ein Skript aus. Beispielsweise kann eine Anwendung, die auf einer EC2-Instance ausgeführt wird, einige zusätzliche Konfigurationsaufgaben ausführen und dann ein Signal an senden, CloudFormation um den Abschluss anzuzeigen.

Das Signal muss an die vordefinierte URL gesendet werden, die durch das Handle der Wartebedingung erzeugt wurde. Sie verwenden diese vordefinierte URL, um Erfolg oder Misserfolg zu signalisieren.

Um ein Signal zu senden

1. Um die vordefinierte URL innerhalb der Vorlage abzurufen, verwenden Sie die intrinsische Funktion Ref mit dem logischen Namen des Handles der Wartebedingung.

Wie im folgenden Beispiel gezeigt, kann Ihre Vorlage eine Amazon EC2-Instance deklarieren und die vordefinierte URL mit der Eigenschaft Amazon EC2 UserData an EC2-Instances übergeben. Auf diese Weise können Skripts 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" }
```

Das Ergebnis ist eine UserData-Ausgabe, die ähnlich aussieht:

```
SignalURL=https://amzn-s3-demo-bucket.s3.amazonaws.com/...
```

Hinweis: In den Befehlszeilentools AWS-Managementkonsole 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 den Wartezustand eintritt, 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-Managementkonsole dem AWS CLI 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 Wartezustandssignals](#) angegeben ist.

Sie müssen die in der Count Eigenschaft angegebene Anzahl von Erfolgssignalen senden, um mit der Stack-Erstellung CloudFormation fortzufahren. 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=ik1twT6hpS4cgNAw7wy0oRejVo
  %3D"
```

wobei die Datei `/tmp/a` die folgende 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, 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=ik1twT6hpS4cgNAw7wy0oRejVo  
%3D"
```

Syntax des Wartezustandssignals

Wenn Sie Signale an die URL senden, die vom Handle der Wartebedingung erzeugt wird, müssen Sie das folgende JSON-Format verwenden:

```
{  
  "Status" : "StatusValue",  
  "UniqueId" : "Some UniqueId",  
  "Data" : "Some Data",  
  "Reason" : "Some Reason"  
}
```

Eigenschaften

Das Feld `Status` muss einen der folgenden Werte enthalten:

- 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 Feld Data kann jede beliebige Information enthalten, die Sie mit dem Signal zurücksenden möchten. Sie können auf den Data Wert zugreifen, indem Sie die GetAtt Funktion [Fn::](#) in der Vorlage verwenden.

Das Feld Reason ist eine Zeichenkette, deren Inhalt außer der JSON-Konformität keinen weiteren Einschränkungen unterliegt.

Zugriff auf Signaldaten

Um auf die von gültigen Signalen gesendeten Daten zuzugreifen, können Sie in Ihrer CloudFormation Vorlage einen Ausgabewert für die Wartebedingung erstellen. 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-stacks](#)Befehl 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. 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 von einem 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 der Modulversionierung](#)

- [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 der Modulversionierung

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, so dass Sie angeben können, welche Version eines Moduls Sie verwenden möchten. Diese Versionskontrolle ist besonders nützlich, wenn Sie ein Modul aktualisieren oder ändern müssen, ohne dass bestehende Stacks, die von diesem Modul abhängen, beschädigt werden.

Beachten Sie die folgenden Punkte, 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 der Region 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 verschiedene Versionen desselben Moduls als Standardversion in verschiedenen Konten oder Regionen registriert haben.

- CloudFormation Verwendet bei Stack-Operationen 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 in allen Konten und Regionen, in denen Sie Ihre Stack-Instances einsetzen wollen, dieselbe Version des Moduls als Standardversion eingestellt 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](#).

Voraussetzungen für die Aktivierung der öffentlichen Module von Drittanbietern

Um ein öffentliches Modul eines Drittanbieters in Ihrem Konto und Ihrer Region erfolgreich zu aktivieren, muss für jede öffentliche Erweiterung (Ressource oder Modul) eines Drittanbieters, die in dem Modul enthalten ist, Folgendes zutreffen:

- **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 in der Region mit demselben Typnamen-Alias registriert werden. Weitere Informationen finden Sie unter [Verwenden Sie Aliase, um auf Erweiterungen zu verweisen](#).
- **Versionskompatibilität-** Die derzeit aktivierte Version der Erweiterung muss eine der unterstützten Hauptversionen dieser Erweiterung sein, die im Modul angegeben sind.

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. Betrachten Sie Module als vorgefertigte Bündel von Ressourcen, die Sie zu Ihren Vorlagen hinzufügen können.

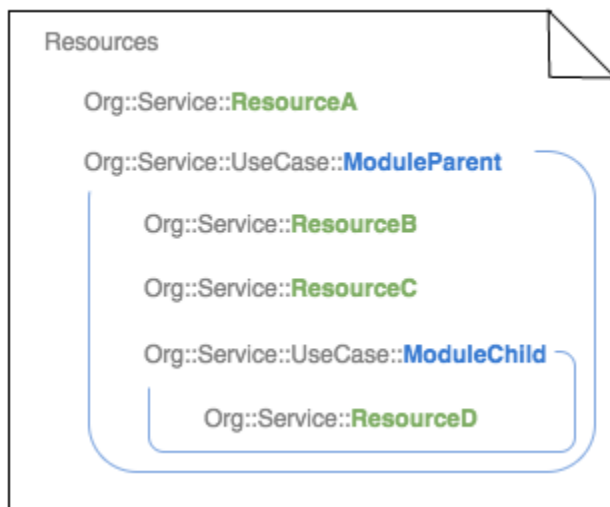
Um ein Modul zu verwenden, gehen Sie wie folgt vor:

- **Modul registrieren** — Sie registrieren Module in der CloudFormation Registrierung als private Erweiterungen. Vergewissern Sie sich, dass es in der AWS-Konto und der Region, in der Sie arbeiten, registriert ist. 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.
- Vorschau der Änderungen- Bevor Sie Änderungen vornehmen, können Sie mit einem Änderungssatz 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, „ResourceA“ sowie ein Modul, „ModuleParent“. Dieses Modul enthält zwei Ressourcen, „ResourceB“ und „ResourceC“ sowie ein verschachteltes Modul, „ModuleChild“. „ModuleChild“ enthält eine einzelne Ressource, „ResourceD“. Wenn Sie aus dieser Vorlage einen Stack erstellen, wird die Vorlage CloudFormation verarbeitet und die Module in die entsprechenden Ressourcen aufgelöst. Der resultierende Stack hat vier Ressourcen: „ResourceA“, „ResourceB“, „ResourceC“ und „ResourceD“.

Template



Processed Template



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 nach Ihren Bedürfnissen ändern. Weitere Informationen zur Definition von Vorlagenparametern finden Sie unter [CloudFormation ParametersVorlagensyntax](#).

In ähnlicher Weise können Module auch Parameter haben. Mit diesen Modulparametern können Sie benutzerdefinierte Werte aus der Vorlage (oder einem anderen Modul), die das Modul verwendet, in das Modul eingeben. Das Modul kann dann diese benutzerdefinierten Werte verwenden, um Eigenschaftswerte für die darin enthaltenen Ressourcen festzulegen.

Sie können auch Vorlagenparameter definieren, die Moduleigenschaften festlegen, so dass Sie Werte eingeben können, die zum Zeitpunkt des Stapelvorgangs an das Modul übergeben werden.

Wenn ein Modul ein verschachteltes Modul enthält, das seine eigenen Modulparameter hat, 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` enthält, definiert einen Vorlagenparameter „`BucketName`“ der es dem Benutzer ermöglicht, 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:

- Es setzt den Parameter `AccessControl` von `My::S3::SampleBucketPrivate::MODULE` auf `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 nicht die Durchsetzung von Beschränkungen. Um eine Einschränkungsprüfung für einen Modulparameter durchzuführen, erstellen Sie einen Vorlagenparameter mit den gewünschten Einschränkungen. Dann referenzieren Sie diesen Vorlagenparameter in Ihrem Modulparameter. Weitere Informationen zur Definition von Vorlagenparametern finden Sie unter [CloudFormation 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 (.) zwischen ihnen kombinieren. Wenn der logische Name des Moduls beispielsweise `MyModule` und der logische Name der Ressource `MyBucket` lautet, können Sie sich auf diese Ressource entweder als `MyModule.MyBucket` oder `MyModuleMyBucket` beziehen.

Um die logischen Namen von Ressourcen innerhalb eines Moduls zu finden, können Sie das Schema des Moduls konsultieren, das in der CloudFormation Registrierung oder mithilfe der [DescribeType](#) Operation verfügbar ist. 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 zum Beispiel ein Modul `My::S3::SampleBucket::MODULE`, das eine Ressource `AWS::S3::Bucket` mit dem logischen Namen `S3Bucket` enthält. Um mit der Funktion `Ref` 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 lautet entweder `MyBucket.S3Bucket` oder `MyBucketS3Bucket`.

-Beispielvorlage

Die folgende Beispielvorlage erstellt einen S3-Bucket unter Verwendung des Moduls `My::S3::SampleBucket::MODULE`. Außerdem wird eine Amazon SQS-Warteschlange erstellt und ihr Name mit dem Bucket-Namen aus dem Modul identisch gemacht. Außerdem gibt die Vorlage den Amazon Resource Name (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 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.

Die Erstellung 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 des Aktualisierungsverhaltens 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.

Das Löschen eines Stapels löscht die mit ihm verbundenen Ressourcen. 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 je nach Bedarf 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](#)
- [Aktualisieren Sie Ihre Stapelvorlage](#)
- [Verstehen des Aktualisierungsverhaltens von Stack-Ressourcen](#)
- [CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren](#)
- [Stack-Bereitstellungen validieren](#)
- [Stapel direkt aktualisieren](#)
- [Eine Stapelaktualisierung abbrechen](#)
- [Einen Stack aus der CloudFormation Konsole löschen](#)
- [Fortschritt des Stapels überwachen](#)
- [Setzen Sie Ihren CloudFormation Stack bei einem Alarmverstoß mit Rollback-Triggern zurück](#)
- [Erkennen Sie nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen mit Drift Detection](#)
- [AWS Ressourcen in einen CloudFormation Stack importieren](#)
- [Stapel-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 zur Verfügung, über die Sie auf Ihre Stacks zugreifen können. Sie können auf die CloudFormation Konsole zugreifen, indem Sie sich bei der anmelden AWS-Managementkonsole, das Suchfeld in der Navigationsleiste verwenden CloudFormation, um nach den Suchergebnissen zu suchen und dann CloudFormation aus 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 SDK für .NET aufbauen. 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 die Handhabung von Low-Level-Details schreiben, wie beispielsweise die Generierung des Hashs zum Signieren der Anfrage. Weitere Informationen zu den API-Aktionen für CloudFormation finden Sie unter [Aktionen](#) in der API-Referenz.AWS CloudFormation
- 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 zum Aufbauen auf 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](#)
- [Stapeloptionen konfigurieren](#)
- [Vorschau der Konfiguration Ihres Stacks](#)

Erstellen eines Stacks

Folgen Sie den Schritten in diesem Abschnitt, um Ihre Vorlage bereitzustellen und einen Stapel zu erstellen.

Voraussetzungen

- Sie müssen eine Stapelvorlage erstellt haben. Weitere Informationen finden Sie unter [Mit CloudFormation Vorlagen arbeiten](#).

So erstellen Sie einen Stapel (Konsole)

1. Öffnen Sie die 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 Stapel erstellen und dann Mit neuen Ressourcen (Standard).

Alternativ können Sie die Option Mit vorhandenen Ressourcen (Ressourcen importieren) wählen, um vorhandene AWS Ressourcen zu importieren, die in Ihrer Vorlage beschrieben sind. Weitere Informationen zu dieser Option finden Sie unter [AWS Ressourcen in einen CloudFormation Stack importieren](#).

4. Führen Sie auf der Seite Stapel erstellen einen der folgenden Schritte aus:

- Um eine vorhandene Vorlage zu verwenden, wählen Sie für Voraussetzung- Vorlage vorbereitendie Option Vorhandene Vorlage auswählen. Wählen Sie dann unter Vorlage angebenentweder Amazon S3 URL oder Eine Vorlagendatei hochladen je nachdem, wo sich die Vorlage befindet.
- Wenn Sie Amazon S3 URLwählen, geben Sie eine URL zu der Vorlagendatei in einem S3-Bucket an.

Wenn Ihre Vorlage verschachtelte Stacks enthält (beispielsweise Stacks, die in anderen Vorlagendokumenten beschrieben sind, die sich in Unterverzeichnissen befinden), stellen Sie sicher, dass Ihr S3-Bucket die erforderlichen Dateien und Verzeichnisse enthält.

Wenn Sie eine Vorlage aus einem Bucket mit Versionskontrolle haben, können Sie eine bestimmte Version der Vorlage angeben, indem Sie `?versionId=version-id` an die URL anhängen. Weitere Informationen über versionierungsfähige Buckets finden Sie unter [Arbeiten mit Objekten in einem versionierungsfähigen Bucket](#) im Amazon Simple Storage Service User Guide.

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 hochladenwählen, wählen Sie Datei auswählen um eine Vorlagendatei von Ihrem lokalen Computer auszuwählen. Die Dateigröße der Vorlage darf höchstens 1 MB 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.

Bei der Verwendung der S3-Buckets, die von erstellt wurden, sollten Sie Folgendes 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 [Einstellen des standardmäßigen serverseitigen Verschlüsselungsverhaltens für Amazon S3-Buckets](#) im Amazon Simple Storage Service User Guide.

- 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 keine fertige Vorlage haben, können Sie Aus Infrastructure Composer erstellen 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 Stapels fortzufahren.
9. (Optional) Ändern Sie auf der Seite Stack-Optionen konfigurieren die Standard-Stack-Optionen. Weitere Informationen finden Sie unter [Stapeloptionen konfigurieren](#).
10. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie für Fähigkeitendie 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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).

11. Wählen Sie Next (Weiter), um fortzufahren.

12. Auf der Seite Überprüfen und erstellen überprüfen Sie die Details Ihres Stacks.

Um einen der Werte zu ändern, bevor Sie den Stapel 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 Stapels zu erhalten, bevor Sie ihn erstellen. Wählen Sie auf der Seite Überprüfen und erstellen die Option Änderungssatz erstellen und folgen Sie den Anweisungen. Weitere Informationen finden Sie unter [Vorschau der Konfiguration Ihres Stacks](#).

14. Wählen Sie Senden um Ihren Stapel 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 [Fortschritt des Stapels überwachen](#).

So erstellen Sie einen Stapel über die Befehlszeile

Verwenden Sie einen der folgenden Befehle:

- [create-stack](#) (AWS CLI)
- [New-CFNStack](#) (AWS Tools for Windows PowerShell)

Beispiele für die Verwendung der Befehlszeile zur Erstellung eines Stapels finden Sie unter [Beispiele für CloudFormation Stack-Operationsbefehle für AWS CLI und PowerShell](#).

Stapeloptionen konfigurieren

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 für Zwecke wie die 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.

Wenn Sie nach der Stack-Erstellung Tags auf Stack-Ebene hinzufügen, aktualisieren oder entfernen, wird eine Stack-Aktualisierung ausgelöst. Alle Ressourcen, die die Weitergabe von Tags auf Stack-Ebene unterstützen, werden entsprechend aktualisiert.

Berechtigungen

Eine bestehende IAM-Servicerolle, die übernommen CloudFormation werden kann. Anstatt Ihre Kontoanmeldedaten zu verwenden, werden die Anmeldeinformationen der Rolle CloudFormation verwendet, um Ihren Stack zu erstellen. Weitere Informationen finden Sie unter [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 Fehlern 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-Trigger 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.

Vorschau der Konfiguration Ihres Stacks

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.

So erstellen Sie einen Änderungssatz für einen neuen Stapel

1. Wählen Sie auf der Seite Überprüfen und erstellen die Option Änderungssatz erstellen.
2. Geben Sie im Dialogfeld Änderungssatz erstellen einen Namen für den Änderungssatz und eine Beschreibung ein, falls gewünscht. Wählen Sie Create change set.

Wenn Sie einen Änderungssatz für einen neuen Stack 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, hat er den gleichen Status wie `CREATE_IN_PROGRESS`, und der Ausführungsstatus lautet `UNAVAILABLE`. Wenn die Erstellung des Änderungssatzes erfolgreich CloudFormation abgeschlossen ist, wird der Änderungssatzstatus auf `CREATE_COMPLETE` gesetzt, 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 `CREATE_FAILED` gesetzt. 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-Managementkonsole um dessen Daten und Ressourcen einzusehen.

Um Informationen über Ihren CloudFormation Stack anzuzeigen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die 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 Namen des Stacks aus. CloudFormation zeigt die Stack-Details für den ausgewählten Stack an.

 Note

Wenn der Stapel gelöscht wurde, können Sie ihn mit der Option Filterstatus 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 Beendigungsschutz an.

Tags (Markierungen)

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 Alarmer, die während des Stack-Vorgangs oder während des angegebenen Überwachungszeitraums überwacht werden CloudFormation sollen. Wenn einer der Alarmer 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 [Fortschritt des Stapels 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](#).

Parameters

Zeigt die Stack-Parameter und deren Werte an.

Für Stapel, die Systems Manager-Parameter enthalten, zeigt die Spalte Aufgelöster Wert die Werte an, die in der Stapeldefinition für die Systems 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 übermittelte Vorlage anzuzeigen, oder „Verarbeitete Vorlage anzeigen“, um die Vorlage anzuzeigen, nachdem CloudFormation die referenzierten Makros verarbeitet wurden. CloudFormation verwendet die verarbeitete Vorlage, um Ihren Stapel zu erstellen oder zu aktualisieren.

Änderungssätze

Zeigt die Änderungssätze des Stapels an.

Weitere Informationen finden Sie unter [Einen Änderungssatz für einen CloudFormation Stack anzeigen](#).

Git-Synchronisation

Zeigt das Git Sync Dashboard des Stacks an.

Weitere Informationen finden Sie unter [Dashboard für den Git-Synchronisierungsstatus](#).

Aktualisieren Sie Ihre Stapelvorlage

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 Versionskontrollsystem gespeichert haben, verwenden Sie eine Kopie dieser Vorlage als Ausgangspunkt. Andernfalls können Sie eine Kopie der Vorlage von abrufen CloudFormation.

Wenn Sie nur die Parameter oder Einstellungen des Stacks ändern möchten (beispielsweise das Amazon SNS-Thema eines Stacks), können Sie die vorhandene Vorlage wiederverwenden, ohne eine Kopie zu erhalten.

Sie können eine CloudFormation Stack-Vorlage mithilfe eines Texteditors oder aktualisieren [Infrastructure Composer](#).

So aktualisieren Sie eine vorhandene Stack-Vorlage mithilfe von Infrastructure Composer

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie auf der Seite Stapel den Namen des zu aktualisierenden Stapels.
3. Wählen Sie die Registerkarte Vorlage und dann Ansicht im Infrastructure Composer.

CloudFormation öffnet die Vorlage in Infrastructure Composer.

4. Aktualisieren Sie Ihre Vorlage mit einer der folgenden Methoden:

- Canvas interface: Hier können Sie aus der Palette Ressourcen ziehen und ablegen. Konfigurieren Sie die Ressourcen, indem Sie auf eine Karte doppelklicken, um das Fenster

Ressourceneigenschaften zu öffnen. Verbinden Sie Ressourcen nach Bedarf. Ausführliche Anweisungen zur Verwendung der Canvas-Oberfläche finden Sie unter [Wie man in AWS Infrastructure Composer](#) komponiert.

- **Schnittstelle Vorlage** : Wechseln Sie von der Schnittstelle Canvas zur Schnittstelle Template . Nehmen Sie Inline-Aktualisierungen am Code der Vorlage vor. Wechseln Sie je nach Bedarf zwischen den Formaten JSON und YAML.
5. Wählen Sie Validieren um die Vorlage auf Syntaxfehler zu überprüfen.
 6. Wenn Sie bereit sind, Änderungen zu exportieren CloudFormation, wählen Sie Vorlage aktualisieren aus.

Um eine bestehende Stack-Vorlage zu aktualisieren, verwenden Sie AWS CLI

1. Um die Vorlage für den Stack zu erhalten, den Sie aktualisieren möchten, verwenden Sie den CLI-Befehl [get-template](#) .
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 Aktualisierungen 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. Zum Beispiel müssen Änderungen an der Eigenschaft `AllocatedStorage` für eine RDS-Datenbankinstance größer sein als der aktuelle Wert. Wenn Ihre Aktualisierung gegen diese Regeln verstößt, wird dieser Teil fehlschlagen.
- Die Aktualisierung einer Ressource kann sich auch auf andere auswirken, die auf sie verweisen. Wenn Sie Funktionen wie [Die Funktion Ref](#) oder [Die Funktion Fn::GetAtt](#) zum Festlegen einer

Eigenschaft verwenden, die auf einer anderen Ressource basiert, CloudFormation wird auch die referenzierende Ressource aktualisiert, wenn sich die referenzierte ändert.

- Informationen über die Auswirkungen der Aktualisierung bestimmter Ressourceneigenschaften finden Sie in der Referenz [AWS Ressource 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 CLI-Befehl [validate-template](#) verwenden oder Ihre Vorlage in der Konsole angeben. Die Konsole führt die Validierung automatisch durch. Diese Methoden überprüfen jedoch nur die Syntax Ihrer Vorlage und 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 des Aktualisierungsverhaltens 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 CloudFormation aktualisiert. Nicht geänderte Ressourcen werden während des Updates ohne Unterbrechung ausgeführt. CloudFormation verwendet für aktualisierte Ressourcen eines der folgenden Aktualisierungsverhalten:

Aktualisierung ohne Unterbrechung

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::Trail](#)Ressource aktualisieren, wird der Pfad ohne Unterbrechung CloudFormation aktualisiert.

Wird mit einigen Unterbrechungen aktualisiert

CloudFormation aktualisiert die Ressource mit einigen Unterbrechungen. Wenn Sie beispielsweise bestimmte Eigenschaften einer [AWS::EC2::Instance](#)Ressource aktualisieren, kann es zu einer Unterbrechung der Instance kommen, während CloudFormation Amazon EC2 die Instance neu konfiguriert.

Ersatz

CloudFormation erstellt die Ressource während eines Updates neu, wodurch auch eine neue physische ID generiert wird. 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::Instance](#)Ressourcentyps aktualisieren, 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 CloudFormation verwendete Methode hängt davon ab, welche Eigenschaft Sie für einen bestimmten Ressourcentyp aktualisieren. Das Aktualisierungsverhalten für jede Eigenschaft wird in der Referenz [AWS Ressource 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, 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 CloudFormation außerdem ein Update für jeden verschachtelten Stack. Dies ist erforderlich, um festzustellen, ob die verschachtelten Stacks geändert wurden. 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. Änderungssätze ermöglichen Ihnen eine Vorschau darauf, wie sich vorgeschlagene Änderungen an einem Stack auf Ihre laufenden Ressourcen auswirken könnten, einschließlich der Auswirkungen auf Ressourceneigenschaften und-attributen. Ob durch Ihre Änderungen kritische Ressourcen gelöscht oder ersetzt werden, CloudFormation nimmt die Änderungen an Ihrem Stack nur vor, 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](#)
- [Verwenden von Änderungssätzen mit Drift-Erkennung](#)
- [Führt einen Änderungssatz für einen CloudFormation Stack aus](#)

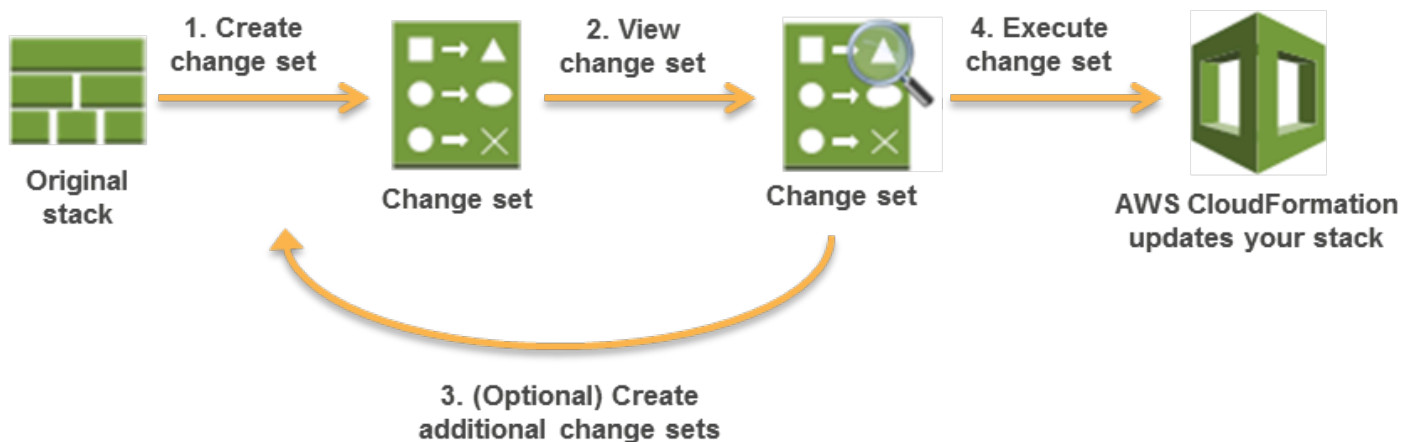
- [Löschen Sie einen Änderungssatz für einen CloudFormation Stack](#)
- [Beispiel für Änderungssätze für CloudFormation Stapel](#)
- [Änderungssätze für verschachtelte Stacks](#)

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 Aktualisierungen unterstützt, oder ob Sie nicht über ausreichende Berechtigungen zum Ändern einer Ressource verfügen, was alles dazu führen kann, 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. Die Erstellung mehrerer Änderungssätze hilft Ihnen zu verstehen und zu bewerten, wie sich verschiedene Änderungen auf Ihre Ressourcen und Eigenschaften auswirken werden. 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)

So erstellen Sie ein Änderungsset

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie auf der Seite Stapel den laufenden Stapel, für den Sie ein Änderungsset erstellen möchten.
4. Wählen Sie in der Detailansicht des Stapels Stapelaktionen und dann Änderungssatz erstellen.
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	Action
So ändern Sie Eingabeparameterwerte	Wählen Sie Vorhandene Vorlage verwenden, und wählen Sie dann Weiter um mit der Eingabe oder Änderung der Eingabeparameterwerte fortzufahren.
So geben Sie den Speicherort einer aktualisierten Vorlage an	Wenn Sie die Vorlage geändert haben, wählen Sie Vorhandene 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 Versionskontrolle haben, können Sie eine bestimmte Version der Vorlage angeben, indem Sie <code>?versionId= <i>version-id</i></code> an die URL anhängen. Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem versionierungsfähigen Bucket im Amazon Simple Storage Service User Guide.

Aufgabe	Action
	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 im Infrastructure Composer bearbeiten und dann Im Infrastructure Composer bearbeiten. Sie werden zu AWS Infrastructure Composer umgeleitet. Nachdem Sie die Vorlage geändert haben, wählen Sie Änderungssatz erstellen und dann Bestätigen aus. Fahren Sie fort CloudFormation, um zu Änderungssatz für **stack-name** Seite erstellen zurückzukehren, und klicken Sie dann auf Weiter.

- Geben Sie auf der Seite Stack-Details an, einen Namen für den Änderungssatz an und geben Sie optional eine Beschreibung des Änderungssatzes an, um seinen Zweck im Abschnitt Übersicht zu identifizieren. 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 Stapel 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 über die Verwendung von NoEcho zur Maskierung sensibler Informationen und die Verwendung dynamischer Parameter zur Verwaltung von Geheimnissen finden Sie in der bewährten Vorgehensweise [Keine Anmeldeinformationen in Vorlagen einbetten](#).

- Auf der Seite Stack-Optionen konfigurieren aktualisieren Sie die Tags des Stacks, die IAM-Service-Rolle, die Stack-Richtlinie, die Rollback-Konfiguration, das Amazon SNS-Benachrichtigungsthema (falls zutreffend) oder die Änderungssätze.

 Note

Änderungssätze für verschachtelte Stacks sind standardmäßig aktiviert, wodurch Änderungssätze 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 über Änderungssätze für verschachtelte Stapel 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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).
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 Details anzeigen in der Spalte Änderungen auf Grundstücksebene um die Änderungen auf Grundstücksebene anzuzeigen.

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.

Create a change set for nested stacks (console)

So erstellen Sie einen Änderungssatz für verschachtelte Stapel

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie auf der Seite Stapel den laufenden Stapel aus, für den Sie einen Änderungssatz erstellen möchten.
4. Wählen Sie in der Detailansicht des Stapels Stapelaktionen und dann Änderungssatz erstellen.
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	Action
So ändern Sie Eingabeparameterwerte	Wählen Sie Vorhandene Vorlage verwenden, und wählen Sie dann Weiter um mit der Eingabe oder Änderung der Eingabeparameterwerte fortzufahren.
So geben Sie den Speicherort einer aktualisierten Vorlage an	Wenn Sie die Vorlage geändert haben, wählen Sie Vorhandene 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 Versionskontrolle haben, können Sie eine bestimmte Version der Vorlage angeben, indem Sie <code>?versionId= <i>version-id</i></code> an die URL anhängen. Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem versionierungsfähigen Bucket im Amazon Simple Storage Service User Guide.

Aufgabe	Action
	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 im Infrastructure Composer bearbeiten und dann Im Infrastructure Composer bearbeiten. Sie werden zu AWS Infrastructure Composer umgeleitet. Nachdem Sie die Vorlage geändert haben, wählen Sie Änderungssatz erstellen und dann Bestätigen aus. Fahren Sie fort CloudFormation, um zu Änderungssatz für **stack-name** Seite erstellen zurückzukehren, und klicken Sie dann auf Weiter.

- Geben Sie auf der Seite Stack-Details an, einen Namen für den Änderungssatz an und geben Sie optional eine Beschreibung des Änderungssatzes an, um seinen Zweck im Abschnitt Übersicht zu identifizieren. 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 Stapel 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 über die Verwendung von NoEcho zur Maskierung sensibler Informationen sowie über die Verwendung dynamischer Parameter zur Verwaltung von Geheimnissen finden Sie in der bewährten Vorgehensweise [Keine Anmeldeinformationen in Vorlagen einbetten](#).

- Auf der Seite Stack-Optionen konfigurieren aktualisieren Sie die Tags des Stacks, die IAM-Service-Rolle, die Stack-Richtlinie, die Rollback-Konfiguration, das Amazon SNS-Benachrichtigungsthema (falls zutreffend) oder die Änderungssätze. Weitere Informationen finden Sie unter [Stapeloptionen konfigurieren](#).

 Note

Änderungssätze für verschachtelte Stacks sind standardmäßig aktiviert, wodurch Änderungssätze für alle in Ihrer Vorlage angegebenen verschachtelten Stacks erstellt werden. Weitere Informationen über Änderungssätze für verschachtelte Stapel 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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).
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 Child-Stack für eine bedingte Ersetzung markieren, wenn sie auf die Ausgabe eines Parent-Stacks verweisen und der Parent-Stack 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 Befehl [create-change-set](#).

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)

So zeigen Sie einen Änderungssatz an

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie auf der Seite Stapel den Namen des Stapels, der den Änderungssatz enthält, den Sie ansehen 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 des Aktualisierungsverhaltens 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`.

- Wählen Sie im Abschnitt Änderungen die Option Details anzeigen in der Spalte Änderungen auf Eigenschaftsebene um die an Ihrer Ressource vorgenommenen Änderungen der Eigenschaftswerte anzuzeigen.
- 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.

Die Tabelle im Abschnitt Änderungen auf Grundstücksebene zeigt den Pfad, Änderungstyp, Vorwert und Nachwert für die betroffenen Grundstücke. Aktivieren Sie in der Tabelle das Kontrollkästchen für jede Änderung, die Sie in den Ansichten Vor und Nach Ihrer Vorlage hervorheben möchten, um zu sehen, welche Änderungen auf der Ebene der Eigenschaften vorgenommen werden.

- Hinzufügen- Hinzugefügte Eigenschaften werden grün hervorgehoben.
- Ändern- Geänderte Eigenschaften werden blau hervorgehoben.
- Entfernen- Entfernte Eigenschaften werden rot hervorgehoben.

View a change set for nested stack (console)

So zeigen Sie einen Änderungssatz für verschachtelte Stacks an (Konsole)

- [Öffnen Sie die CloudFormation Konsole unter /cloudformation. https://console.aws.amazon.com](https://console.aws.amazon.com)
- Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
- Wählen Sie auf der Seite Stapel den Namen des Stapels, der den Änderungssatz enthält, den Sie ansehen möchten.
- Wählen Sie im Navigationsbereich Änderungssätze um eine Liste der Änderungssätze des Stapels anzuzeigen.
- 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) 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 des Aktualisierungsverhaltens 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 Child-Stack für eine bedingte Ersetzung markieren, wenn sie auf die Ausgabe eines Parent-Stacks verweisen und der Parent-Stack 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 Stapels 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. 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` mit `describe-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-Instance aktualisiert.

Eine Beschreibung der einzelnen Felder finden Sie unter dem Datentyp [Change](#) in der AWS CloudFormation API-Referenz.

Weitere Beispiele von Änderungssätzen finden Sie unter [Beispiel für Änderungssätze für CloudFormation Stapel](#).

So zeigen Sie Änderungen auf Eigenschaftsebene in einem Änderungssatz (AWS CLI) an

- Der folgende Befehl listet die Änderungen auf Eigenschaftsebene auf, die mit einem Änderungssatz für eine `AWS::EC2::NetworkInterface`-Ressource verbunden sind, der die `Ipv4Prefixes`-Eigenschaft entfernt, die `Description`-Eigenschaft für die Ressource ändert und eine Tag hinzufügt:

```
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\":\"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",

```

Verwenden von Änderungssätzen mit Drift-Erkennung

Bei Änderungssätzen, bei denen Änderungen berücksichtigt werden, handelt es sich um erweiterte CloudFormation Änderungssätze, mit denen Sie Stackabweichungen auf sichere Weise erkennen und bewältigen können. Stacks weichen von ihrer erwarteten Vorlagenkonfiguration ab, wenn Stack-Ressourcen außerhalb geändert werden CloudFormation, indem das AWS-Managementkonsole AWS CLI, oder SDK der zugrunde liegenden Dienste verwendet wird. Änderungssätze, die Drift berücksichtigen, vergleichen Vorlagen mit dem tatsächlichen Zustand Ihrer

Stack-Ressourcen und passen die veränderten Ressourcen an ihre Vorlagendefinitionen an. Wenn die Vorlagendefinition einer Ressource aktualisiert wird, sodass sie ihrem aktuellen Status entspricht, setzen Änderungssätze, die Abweichungen erkennen, den Drift-Status der Ressource zurück, ohne die Ressource zu ändern.

Die Verwendung von Änderungssätzen, bei denen Abweichungen berücksichtigt werden, bietet unter anderem folgende Vorteile:

- Überschreibungen von Drift als Vorschau anzeigen — Ermitteln Sie, ob bei einer Bereitstellung kritische out-of-band Änderungen rückgängig gemacht werden, die während der Reaktion auf einen Vorfall vorgenommen wurden.
- Systematischer Abgleich von Abweichungen — Richten Sie veränderte Ressourcen sicher an konforme Vorlagendefinitionen ab, indem Sie entweder veränderte Eigenschaften mit Vorlagenwerten überschreiben oder die Vorlage so aktualisieren, dass sie dem tatsächlichen Ressourcenstatus entspricht.
- Verbesserte Rollback-Funktionen — Wenn eine Bereitstellung fehlschlägt, stellen Sie die Ressourcen auf ihren tatsächlichen Zustand vor der Bereitstellung zurück, nicht auf den vorherigen Vorlagenstatus.
- Vollständige Transparenz — Sehen Sie genau, wie sich Ihre Bereitstellung auf die tatsächlichen Ressourcenkonfigurationen auswirkt, bevor Sie Änderungen vornehmen.

Herkömmliche Änderungssätze bieten einen Vergleich Ihrer neuen Vorlage mit der vorherigen Vorlage für einen Stack, berücksichtigen jedoch nicht die Stack-Drift. Änderungssätze, die Drift berücksichtigen, lösen dieses Problem, indem sie einen dreiseitigen Vergleich ermöglichen zwischen:

- Aktueller Status — Die Live-Konfiguration Ihrer Ressourcen. CloudFormation liest die Konfiguration der Ressourcen in Ihrem Konto zum Zeitpunkt der Erstellung des Änderungssatzes über den zugrunde liegenden Dienst APIs.
- Status der vorherigen Bereitstellung — Die in der Vorlage definierte Konfiguration aus Ihrer letzten CloudFormation Bereitstellung.
- Gewünschter Status — Die in Ihrer neuen Vorlage definierte Konfiguration.

Änderungssätze, die Drift berücksichtigen, aktualisieren den aktuellen Status aller Stack-Ressourcen, sodass er dem gewünschten Status entspricht, auch wenn eine Ressource in der Vorlage nicht explizit geändert wurde.

Überlegungen

- **Unterstützung von Ressourcentypen** — Änderungssätze, die Drift-Änderungen berücksichtigen, unterstützen den Vergleich des gewünschten Zustands mit dem tatsächlichen Status für Hunderte von Ressourcentypen. Bei Ressourcentypen, die nicht unterstützt werden, greifen Änderungssätze, die Änderungen erkennen, auf den Vergleich des vorherigen Bereitstellungsstatus mit dem gewünschten Status zurück. Einzelheiten finden Sie im Abschnitt *Unterstützte Ressourcentypen*.
- **Schreibgeschützte Eigenschaften** — Bei Eigenschaften, die vertrauliche Daten (Passwörter, Geheimnisse) enthalten, werden Änderungssätze, bei denen Änderungen berücksichtigt werden, mit früheren Bereitstellungswerten verglichen und nicht mit tatsächlichen Werten.
- **AWS-verwaltete Eigenschaften** — Drift-fähige Änderungssätze behalten out-of-band Änderungen bei, die von AWS Diensten an verwalteten Eigenschaften vorgenommen wurden, wie z. B. die gewünschte Kapazität einer Auto Scaling Group. Einzelheiten finden Sie im Abschnitt *AWS mit verwalteten Eigenschaften*.
- **Externe Tag-Schlüssel** — Bei Änderungssätzen, die Drift-fähig sind, werden keine Tagschlüssel entfernt oder geändert, die nicht in der Vorlage angegeben wurden, wodurch Konflikte mit ABAC-Systemen (attribute-based access control) vermieden werden.
- **Ersatz veränderter Ressourcen** — Änderungssätze, die Drift-Änderungen berücksichtigen, unterstützen keinen Drift-Abgleich für unveränderliche Eigenschaften.
- **Stapelübergreifende Anlagen** — Einige Ressourcentypen, wie z. B. `AWS::IAM::Policy`, können den tatsächlichen Zustand anderer Ressourcentypen beeinflussen, wie z. B. `AWS::IAM::Role`. Änderungssätze, die Drift-fähig sind, behandeln das Anhängen von Ressourcen innerhalb eines Stacks. Wenn eine Ressource durch das Anhängen einer sekundären Ressource aus einem anderen Stapel geändert wird, erkennen Änderungssätze, die Abweichungen berücksichtigen, die Änderung als Drift und können die Zuordnung rückgängig machen. Zu den beliebtesten Ressourcen, die angehängt werden können `AWS::IAM::Policy`, gehören, und `AWS::IAM::ManagedPolicy`. `AWS::EC2::SecurityGroupIngress` `AWS::EC2::SecurityGroupEgress`

AWS-verwaltete Immobilien

Sie können bestimmte Ressourceneigenschaften für die aktive AWS Verwaltung konfigurieren. Sie können beispielsweise Amazon Relational Database Service (Amazon RDS) erlauben, die Nebenversion einer Amazon RDS-Tabelle automatisch zu aktualisieren. Diese Änderungen können sich zeigen, wenn der Stack CloudFormation eindringt. Änderungssätze, die Drift berücksichtigen, erkennen, dass bei Eigenschaften, die AWS verwaltet werden, eine Abweichung zu erwarten ist, und

lassen ihren tatsächlichen Wert unverändert, wenn Sie die Eigenschaft in ihrer Vorlage nicht geändert haben. Die wichtigsten Beispiele für -verwaltete Eigenschaften sind: AWS

- Aktivierung der `AutoMinorVersionUpgrade` Eigenschaft einer Amazon RDS-Tabelle, um automatische Updates der Engine-Version zu ermöglichen.
- Verwendung der `AWS::ApplicationAutoScaling::ScalableTarget` Ressource zur Aktivierung der auto-scaling für Eigenschaften wie die read/write Kapazitätseinheiten einer Amazon DynamoDB-Tabelle und die gewünschte Anzahl eines Amazon Elastic Container Service-Clusters.
- Verwenden der `AWS::AutoScaling::ScalingPolicy` für Auto Scaling Scaling-Gruppen.

Änderungssätze, die Drift berücksichtigen, verdeutlichen die Eigenschaften, die als AWS„veraltet“ identifiziert wurden. Einzelheiten finden Sie im AWS CLI Abschnitt.

Verwendung von Änderungssätzen, die Änderungen berücksichtigen (Konsole)

Sie können über die CloudFormation Konsole änderungsabhängige Änderungssätze mit demselben Arbeitsablauf wie bei herkömmlichen Änderungssätzen erstellen und verwalten, jedoch mit zusätzlichen Optionen für Bereitstellungsmodi.

Änderungssätze erstellen, bei denen Änderungen berücksichtigt werden

So erstellen Sie einen Änderungssatz, der Abweichungen berücksichtigt (Konsole)

1. [Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com /cloudformation.](https://console.aws.amazon.com/cloudformation)
2. Wählen Sie auf der Seite Stacks den Stack aus, den Sie aktualisieren möchten.
3. Wählen Sie Stack-Aktionen und dann Änderungssatz für aktuellen Stack erstellen aus.
4. Wählen Sie auf der Seite „Änderungssatz erstellen“ für „Change set type“ die Option Drift aware change set aus.
5. Wählen Sie Create change set.
6. Sehen Sie sich auf der Seite mit den Details zum Änderungssatz den dreiseitigen Vergleich an, in dem die tatsächliche, vorherige Bereitstellung und der gewünschte Status für jede betroffene Ressource angezeigt werden.
7. Wenn Sie mit den Änderungen zufrieden sind, wählen Sie Änderungssatz ausführen.

Überprüfe Änderungssätze, bei denen Abweichungen berücksichtigt werden

Wenn Sie sich einen Änderungssatz, der Änderungen berücksichtigt, in der Konsole ansehen, werden Ihnen im Vergleich zu herkömmlichen Änderungssätzen erweiterte Informationen angezeigt:

- **Stack-Drift-Status** — Gibt an, ob der Stack seit seiner letzten Bereitstellung nicht mehr richtig funktioniert hat.
- **Eigenschaftenvergleich** — Zeigt einen JSON-Unterschied zwischen dem tatsächlichen und dem gewünschten Status einer betroffenen Ressource an.
- **Drift-Indikatoren** — Kennzeichnet deutlich Eigenschaften innerhalb des JSON-Diffs, die sich verändert haben. Klicken Sie auf Drift anzeigen, um den vorherigen Bereitstellungswert für eine Eigenschaft zu sehen.
- **Indikatoren für die Wertquelle** — Zeigt an, ob die vorherigen Werte für eine Eigenschaft aus dem aktuellen Zustand oder dem vorherigen Bereitstellungsstatus stammen.

Verwendung von Änderungssätzen, bei denen Abweichungen berücksichtigt werden (AWS CLI)

Sie können änderungsabhängige Änderungssätze mithilfe von erstellen und verwalten, AWS CLI indem Sie dem Befehl den `--deployment-mode REVERT_DRIFT` Parameter hinzufügen. `create-change-set`

Änderungssätze erstellen, bei denen Änderungen berücksichtigt werden

So erstellen Sie einen Änderungssatz, der Abweichungen berücksichtigt

Verwenden Sie den [create-change-set](#) Befehl mit dem `--deployment-mode REVERT_DRIFT` Parameter, um einen Änderungssatz zu erstellen, der Abweichungen berücksichtigt.

```
aws cloudformation create-change-set \  
  --stack-name my-stack \  
  --change-set-name my-drift-aware-changeset \  
  --template-body file://updated-template.yaml \  
  --deployment-mode REVERT_DRIFT \  
  --capabilities CAPABILITY_IAM
```

Überprüfung von Änderungssätzen, bei denen Abweichungen berücksichtigt werden

Um die Details eines Änderungssatzes zu überprüfen, bei dem Änderungen berücksichtigt werden

Verwenden Sie den [describe-change-set](#) Befehl, um die Details eines Änderungssatzes zu überprüfen, bei dem Änderungen berücksichtigt wurden:

```
aws cloudformation describe-change-set \  
  --change-set-name my-drift-aware-changeset \  
  --stack-name my-stack
```

Die Antwort umfasst erweiterte Informationen für Änderungssätze, bei denen Abweichungen berücksichtigt werden:

- **StackDriftStatus**— Zeigt an, ob sich der Stapel verändert hat (DRIFTED, IN_SYNCNOT_CHECKED, oder). UNKNOWN
- **ResourceDriftStatus**— Zeigt den Drift-Status für jede Ressource an (DELETED,MODIFIED,IN_SYNC, oderNOT_CHECKED).
- **BeforeValueFrom**— Gibt an, ob der vorherige Wert für eine Eigenschaft von ACTUAL_STATE oder stammtPREVIOUS_DEPLOYMENT_STATE.
- **Drift**— Enthält die Drift-Details für eine EigenschaftPreviousValue, einschließlichActualValue, undDriftDetectionTimestamp.
- **ResourceDriftIgnoredProperties**— Enthält die Eigenschaften einer Ressource, für die der Änderungssatz die Drift nicht rückgängig macht, sowie die Gründe für das Ignorieren der Drift.

Weitere Informationen finden Sie unter [DescribeChangeSet](#) in der AWS CloudFormation -API-Referenz.

Änderungssätze werden ausgeführt, bei denen Abweichungen berücksichtigt werden

Um einen Änderungssatz auszuführen, bei dem Änderungen berücksichtigt werden

Nachdem Sie den Änderungssatz überprüft haben, verwenden Sie den [execute-change-set](#) Befehl, um die Änderungen zu übernehmen:

```
aws cloudformation execute-change-set \  
  --change-set-name my-drift-aware-changeset \  
  --stack-name my-stack
```

Wenn die Bereitstellung fehlschlägt, CloudFormation werden die Ressourcen auf ihren tatsächlichen Zustand vor der Bereitstellung zurückgesetzt und die seit der letzten Bereitstellung out-of-band vorgenommenen Änderungen beibehalten.

Unterstützte Ressourcentypen

Änderungssätze, bei denen Abweichungen berücksichtigt werden, unterstützen einen dreiseitigen Vergleich zwischen dem aktuellen Status, dem Status der vorherigen Bereitstellung und dem gewünschten Status für alle [Ressourcentypen](#), die Drift-Erkennung unterstützen, mit Ausnahme der folgenden Ressourcentypen:

- `AWS::SageMaker::MlflowTrackingServer`
- `AWS::Route53Resolver::FirewallRuleGroup`
- `AWS::MediaLive::Multiplexprogram`
- `AWS::VpcLattice::ResourceConfiguration`
- `AWS::S3::StorageLensGroup`
- `AWS::Bedrock::AgentAlias`
- `AWS::MSK::Cluster`
- `AWS::RDS::DBProxy`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::QBusiness::Index`
- `AWS::NetworkManager::CoreNetwork`
- `AWS::IAM::OIDCProvider`
- `AWS::Organizations::ResourcePolicy`
- `AWS::SNS::TopicInlinePolicy`
- `AWS::Route53::KeySigningKey`
- `AWS::DataZone::PolicyGrant`
- `AWS::Transfer::Certificate`
- `AWS::SageMaker::ImageVersion`
- `AWS::Neptune::DBParameterGroup`
- `AWS::ODB::CloudVmCluster`
- `AWS::RolesAnywhere::TrustAnchor`
- `AWS::Detective::Graph`
- `AWS::Maester::DocumentType`
- `AWS::SageMaker::ModelPackageGroup`

- `AWS::S3Express::BucketPolicy`
- `AWS::Panorama::PackageVersion`
- `AWS::S3Tables::TableBucketPolicy`

Änderungssätze, bei denen Abweichungen berücksichtigt werden, greifen bei Ressourcen, die den dreiseitigen Vergleich nicht unterstützen, auf einen Vergleich zwischen dem vorherigen Bereitstellungsstatus und dem gewünschten Status zurück.

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 einen 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 [Stapel direkt aktualisieren](#).

Execute a change set (console)

So führen Sie einen Änderungssatz aus

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie auf der Seite Stapel den Namen des Stapels, den Sie aktualisieren möchten.

4. Wählen Sie im Navigationsbereich Änderungssätze um eine Liste der Änderungssätze des Stapels anzuzeigen.
5. Wählen Sie den Namen des Änderungssatzes aus, den Sie anzeigen möchten.
6. Wählen Sie auf der Detailseite des Änderungssatzes Änderungssatz ausführen.

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 [Fortschritt des Stapels überwachen](#).

Execute a change set for nested stacks (console)

So führen Sie einen Änderungssatz für verschachtelte Stapel aus

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie auf der Seite Stapel den Namen des Stapels, den Sie aktualisieren möchten. Sie müssen den mit dem Root-Änderungssatz verknüpften Stack-Namen auswählen.
4. Wählen Sie im Navigationsbereich Änderungssätze um eine Liste der Änderungssätze des Stapels anzuzeigen.
5. Wählen Sie den Namen des Root-Änderungssatz, den Sie ausführen möchten.
6. Wählen Sie auf der Detailseite des Änderungssatzes Änderungssatz ausführen.

Note

CloudFormation führt die in Ihrem Root-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 [Fortschritt des Stapels 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. Um den Fortschritt des Stacks zu sehen, verwenden Sie den Befehl [describe-stacks](#).

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 CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihre AWS-Region.
3. Wählen Sie auf der Seite Stapel den Namen des Stapels aus, der den Änderungssatz enthält, den Sie löschen möchten.
4. Wählen Sie im Navigationsbereich Änderungssätze um eine Liste der Änderungssätze des Stapels anzuzeigen.

5. Wählen Sie den Namen des Änderungssatzes aus, den Sie löschen möchten.
6. Wählen Sie auf der Detailseite des Änderungssatzes **Änderungssatz löschen**.

CloudFormation beginnt sofort, den Änderungssatz aus der Liste der Änderungssätze des Stacks zu löschen, und Sie werden zur Stacks-Seite weitergeleitet.

Delete a change set for nested stacks (console)

So löschen Sie einen Änderungssatz für verschachtelte Stapel

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie auf der Seite Stapel den Stapelnamen aus, der mit dem Stammänderungssatz verbunden ist.
4. Wählen Sie im Navigationsbereich Änderungssätze um eine Liste der Änderungssätze des Stapels 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, den Änderungssatz aus der Liste der Änderungssätze des Stacks 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 Stapeln](#) 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 [DescribeChangeSet](#)API-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. CloudFormation bezeichnet 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 Struktur Changes funktioniert ähnlich wie im Beispiel [Direktes Bearbeiten einer Vorlage](#). 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, undKeyPairName) geändert wurden, die alle von der EC2 Instanz verwendet werden. Aufgrund dieser Änderungen muss die Instanz ersetzt CloudFormation werden, da das Replacement Feld gleich istTrue.

```
{
  "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::Instance](#) Ressourcentyps.

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 Ebene der Eigenschaften anzeigen

Das folgende Beispiel zeigt Änderungen auf Eigenschaftsebene an der Tag Eigenschaft einer EC2 Amazon-Instance. Die Tags Value und Key werden 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": "{\n\"Properties\":{\n\"KeyName\":\n\"BatchTest\",\n\"ImageId\n\":\n\"ami-8fcee4e5\",\n\"InstanceType\":\n\"t2.micro\",\n\"Tags\":[{\n\"Value\":\n\"testing\",\n\"Key\":\n\"Purpose\"}]]}\n",
        "AfterContext": "{\n\"Properties\":{\n\"KeyName\":\n\"BatchTest\",\n\"ImageId\n\":\n\"ami-8fcee4e5\",\n\"InstanceType\":\n\"t2.micro\",\n\"Tags\":[{\n\"Value\":\n\"Test\",\n\"Key\n\":\n\"Test\"}]]}\n"
    }
}
]

```

Die Struktur `Details` zeigt die Werte für `Key` und `Value` vor der Ausführung des Änderungssatzes und die Werte, die sie nach der Ausführung des Änderungssatzes haben werden.

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 Features, 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. 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 über Änderungssätze finden Sie unter [CloudFormation Stapel mithilfe von Änderungssätzen aktualisieren](#).
- Ein verschachtelter Stapel ist ein Stapel, der als Teil eines anderen Stapels erstellt wird. 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 Informationen über verschachtelte Stapel finden Sie unter [Aufteilung einer Vorlage in wiederverwendbare Teile mit verschachtelten Stapeln](#).

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 übermitteln haben. Änderungssätze für verschachtelte Stacks sind in der CloudFormation -Konsole standardmäßig aktiviert. Weitere Informationen finden Sie unter [Einen Änderungssatz für einen CloudFormation Stapel erstellen](#).

▼ Change sets

Change sets for nested stacks

Creates change sets for all nested stacks specified in your template. [Learn more](#) 

☒ Enabled

Creates a change set for all nested stacks specified in the template.

☐ Disabled

Create a change set only on the target stack.

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 Stapeloperationen auf verschachtelten Stapeln](#).

- **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 Stapelhierarchie zu erstellen, geben Sie die Option `--include-nested-stacks` 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 aktualisiert einen Stack mit den Eingabedaten, die beim Erstellen des angegebenen Änderungssatzes bereitgestellt wurden. Um einen Änderungssatz für die gesamte Stapelhierarchie zu erstellen, müssen Sie bei der Operation create-change-set die Option `--include-nested-stacks` 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 Änderungsdatensätzen erfolgt asynchron für Änderungsdatensätze, die mit der Option `--include-nested-stacks` 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
```

```
--stack-name my-root-stack \  
--change-set-name my-root-stack-change-set
```

Stack-Bereitstellungen validieren

Mit der Validierung vor der Bereitstellung können Sie potenzielle Probleme bei der Bereitstellung identifizieren und lösen, bevor Sie Ihre CloudFormation Änderungsätze ausführen. Diese Funktion validiert Ihre Vorlagen anhand gängiger Fehlerszenarien und hilft Ihnen, Probleme catch im Entwicklungszyklus zu erkennen.

Themen

- [So funktioniert die Validierung vor der Bereitstellung](#)
- [Überlegungen](#)
- [Voraussetzungen](#)
- [Validieren Sie eine Stack-Bereitstellung \(Konsole\)](#)
- [Validieren Sie eine Stack-Bereitstellung \(AWS CLI\)](#)
- [Arten der Validierung](#)
- [Begrenzte Ressourcen](#)

So funktioniert die Validierung vor der Bereitstellung

Die Validierung vor der Bereitstellung umfasst die folgenden Phasen:

1. Erstellen Sie Ihren Änderungsatz — Generieren Sie einen Änderungsatz, wie Sie es normalerweise für Ihre CloudFormation Stack-Updates tun würden. Die Validierung vor der Bereitstellung ist standardmäßig aktiviert, wenn Sie Ihr Änderungsatz erstellen.
2. Ausführung der Validierung — CloudFormation führt mehrere Validierungsprüfungen anhand Ihrer Vorlage und Ihrer Zielumgebung durch. Derzeit werden drei Arten der Validierung unterstützt: Überprüfung der Eigenschaftssyntax anhand von Ressourcenschemas, Erkennung von Ressourcennamenkonflikten mit vorhandenen Ressourcen und Überprüfung des leeren S3-Buckets bei Löschvorgängen.
3. Überprüfung der Validierungsergebnisse — CloudFormation liefert detailliertes Feedback zu allen gefundenen Problemen, einschließlich einer genauen Pfadangabe, wo das Problem in der Vorlage liegt, wodurch manuelles Debuggen von Vorlagen entfällt.

4. Probleme lösen — Beheben Sie festgestellte Probleme, indem Sie Ihre Vorlagen aktualisieren oder Konflikte lösen, bevor Sie mit der Bereitstellung fortfahren.
5. Zuverlässige Ausführung — Implementieren Sie Ihren Änderungssatz in der Gewissheit, dass häufig auftretende Ausfallszenarien im Voraus validiert wurden.

Überlegungen

Beachten Sie bei der Verwendung der Validierung vor der Bereitstellung Folgendes:

- Die Validierung vor der Bereitstellung konzentriert sich auf die drei häufigsten Fehlerszenarien bei der Bereitstellung. Sie garantiert nicht, dass Ihre Implementierung erfolgreich sein wird, verringert jedoch die Wahrscheinlichkeit häufiger Fehler.
- Die Validierungsmodi verhalten sich unterschiedlich:
 - Der FAIL-Modus verhindert die Ausführung von Änderungssätzen, wenn bei der Validierung Fehler erkannt werden, und stellt so sicher, dass problematische Vorlagen nicht mit der Bereitstellung fortfahren können. Dies gilt für Syntaxfehler bei Eigenschaften und Konflikte bei der Benennung von Ressourcen.
 - Der WARN-Modus ermöglicht die erfolgreiche Erstellung von Änderungssätzen trotz fehlgeschlagener Validierung und bietet Warnmeldungen, die Entwickler vor der Ausführung überprüfen und beheben können. Dies gilt für Verstöße gegen Beschränkungen wie das Leeren eines S3-Buckets, die möglicherweise durch manuelles Eingreifen behoben werden können.
- Die Überprüfungsergebnisse sind an den jeweiligen Änderungssatz gebunden. Wenn Sie Ihre Vorlage ändern, müssen Sie einen neuen Änderungssatz erstellen, um aktualisierte Überprüfungsergebnisse zu erhalten.
- Die S3-Bucket-Validierung prüft nur das Vorhandensein von Objekten, nicht aber auf Bucket-Richtlinien oder andere Einschränkungen, die das Löschen verhindern könnten.

Voraussetzungen

Um die Validierung vor der Bereitstellung verwenden zu können, müssen Sie über Folgendes verfügen:

- Die erforderlichen IAM-Berechtigungen, um Änderungssätze zu erstellen und Ressourcen in Ihrem Konto zu lesen. Für die Prüfung des leeren S3-Buckets benötigen Sie eine entsprechende Genehmigung. `s3:ListBucket`

- Zugriff auf den Ort, AWS-Regionen an dem Ihre Stacks bereitgestellt werden.
- CloudFormation Vorlagen, die Sie vor der Bereitstellung validieren möchten.

Validieren Sie eine Stack-Bereitstellung (Konsole)

Verwenden Sie das folgende Verfahren, um Ihre Stack-Bereitstellung mithilfe der Konsole zu validieren.

Um eine Vorlage vor der Bereitstellung zu validieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm aus, AWS-Region wo sich Ihr Stack befindet.
3. Wählen Sie auf der Seite Stapel den laufenden Stapel, für den Sie ein Änderungssatz erstellen möchten.
4. Wählen Sie im Bereich mit den Stack-Details die Option Stapel aktualisieren und dann Änderungssatz erstellen aus.
5. Laden Sie auf der *stack-name* Seite Änderungssatz erstellen für Ihre aktualisierte Vorlage hoch oder geben Sie die Vorlagenquelle an.
6. Wählen Sie Weiter, um mit den verbleibenden Konfigurationsschritten für den Änderungssatz fortzufahren.
7. Wenn die Vorlage IAM-Ressourcen enthält, wählen Sie für Funktionen 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 [Bestätigung von IAM-Ressourcen](#) in Vorlagen. CloudFormation
8. Wählen Sie auf der Seite „Überprüfen“ die Option Änderungssatz erstellen aus.
9. CloudFormation erstellt den Änderungssatz und führt Validierungsprüfungen durch. Überprüfen Sie die Validierungsergebnisse auf der Registerkarte „Bereitstellungsvalidierung“.
10. Wenn die Überprüfung erfolgreich ist oder Sie mit den Warnungen zufrieden sind, wählen Sie „Änderungssatz ausführen“, um Ihre Änderungen zu implementieren.
11. Wenn die Überprüfung fehlschlägt, beheben Sie die Probleme und erstellen Sie einen neuen Änderungssatz, um Ihre Bereitstellung erneut zu validieren.

Validieren Sie eine Stack-Bereitstellung (AWS CLI)

Zu den AWS CLI Befehlen für die Validierung vor der Bereitstellung gehören:

- [create-change-set](#) automatische Validierung während der Erstellung des Änderungssatzes.
- [describe-change-set](#) um den Status des Änderungssatzes zu überprüfen
- [describe-events](#) zur Überprüfung der Validierungsergebnisse

Verwenden Sie das folgende Verfahren, um Ihre Stack-Bereitstellung mithilfe von zu validieren. AWS CLI

Um eine Vorlage vor der Bereitstellung zu validieren

1. Verwenden Sie den [Befehl: create-change-set](#)

```
aws cloudformation create-change-set \  
  --stack-name MyStack \  
  --change-set-name MyChangeSet \  
  --change-set-type "CREATE" \  
  --template-body file://updated-template.yaml
```

Der Befehl gibt sowohl den Change-Set-ARN als auch den Stack-ARN zurück.

2. Verwenden Sie den Befehl [describe-events](#) mit dem ARN des Änderungssatzes oder dem Namen des Änderungssatzes, um den Validierungsstatus und die Ergebnisse zu überprüfen.

```
aws cloudformation describe-events \  
  --change-set-name "arn:aws:cloudformation:us-east-1:123456789012:changeSet/MyChangeSet/94498df5-1afb-43b1-9869-9f82b2d877ac"
```

Beispiel für die Ausgabe eines Validierungsfehlers:

```
{  
  "OperationEvents": [  
    {  
      "EventId": "9b5c9a29-4704-4ad0-8082-afb49418d55b",  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/  
c3908380-b357-11f0-a97f-0ad08f35df65",  
      "OperationId": "f558b823-e1e3-4de3-a222-e6b930ddcad4",  
      "OperationType": "CREATE_CHANGESET",
```

```

    "OperationStatus": "FAILED",
    "EventType": "STACK_EVENT",
    "Timestamp": "2025-10-27T17:10:02.923Z",
    "StartTime": "2025-10-27T17:09:57.537Z",
    "EndTime": "2025-10-27T17:10:02.923Z"
  },
  {
    "EventId": "2d8c3262-3468-4283-82fb-6e780e9e4f1d",
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/c3908380-b357-11f0-a97f-0ad08f35df65",
    "OperationId": "f558b823-e1e3-4de3-a222-e6b930ddcad4",
    "OperationType": "CREATE_CHANGESET",
    "EventType": "VALIDATION_ERROR",
    "LogicalResourceId": "NotificationBucket",
    "PhysicalResourceId": "",
    "ResourceType": "AWS::S3::Bucket",
    "Timestamp": "2025-10-27T17:10:02.461Z",
    "ValidationFailureMode": "FAIL",
    "ValidationName": "PROPERTY_VALIDATION",
    "ValidationStatus": "FAILED",
    "ValidationStatusReason": "#/NotificationConfiguration/QueueConfigurations/0: required key [Event] not found",
    "ValidationPath": "/Resources/NotificationBucket/Properties/NotificationConfiguration/QueueConfigurations/0"
  }
]
}

```

3. Beheben Sie alle Validierungsfehler, indem Sie Ihre Vorlage aktualisieren und dann einen neuen Änderungssatz erstellen.
4. Sobald die Validierung bestanden ist, führen Sie den Änderungssatz aus:

```

aws cloudformation execute-change-set \
  --change-set-name MyChangeSet \
  --stack-name MyStack

```

Arten der Validierung

Die Validierung vor der Bereitstellung umfasst die folgenden Arten von Prüfungen:

- **Überprüfung der Eigenschaftssyntax** — Überprüft Ressourceneigenschaften anhand von AWS Ressourcenschemas. Es sucht nach erforderlichen Eigenschaften und gültigen Eigenschaftswerten und identifiziert veraltete oder nicht unterstützte Eigenschaftskombinationen.
- **Konflikterkennung bei Ressourcennamen** — Überprüft, ob Namenskonflikte mit vorhandenen Ressourcen bestehen. AWS Es überprüft, ob Ressourcennamen die AWS Benennungsanforderungen erfüllen, und identifiziert potenzielle Konflikte, bevor die Bereitstellung versucht wird.
- **Überprüfung des leeren S3-Buckets** — Warnt beim Versuch, S3-Buckets zu löschen, die Objekte enthalten. Es bietet Objektzählungen, um die Auswirkungen von Löschungen besser einschätzen zu können und häufige Fehler beim Löschen von S3-Dateien zu verhindern.

Jeder Überprüfungstyp enthält spezifische Fehlermeldungen und eine Fehlerposition in der Vorlage, damit Sie Probleme schnell lösen können.

Begrenzte Ressourcen

Die folgenden Ressourcentypen werden für die Validierung vor der Bereitstellung nicht unterstützt:

- `AWS::ApiGatewayV2::ApiGatewayManagedOverrides`
- `AWS::ApiGatewayV2::Stage`
- `AWS::AppMesh::GatewayRoute`
- `AWS::AppMesh::Mesh`
- `AWS::AppMesh::Route`
- `AWS::AppMesh::VirtualGateway`
- `AWS::AppMesh::VirtualNode`
- `AWS::AppMesh::VirtualRouter`
- `AWS::AppMesh::VirtualService`
- `AWS::AppStream::Fleet`
- `AWS::AppStream::Stack`
- `AWS::AppStream::StackFleetAssociation`
- `AWS::AppStream::StackUserAssociation`
- `AWS::AppStream::User`
- `AWS::AppSync::ApiCache`

- `AWS::AppSync::ApiKey`
- `AWS::AppSync::GraphQLSchema`
- `AWS::AutoScalingPlans::ScalingPlan`
- `AWS::Budgets::Budget`
- `AWS::CertificateManager::Certificate`
- `AWS::Cloud9::EnvironmentEC2`
- `AWS::CloudFormation::CustomResource`
- `AWS::CloudFormation::Macro`
- `AWS::CloudFormation::WaitCondition`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CloudFront::StreamingDistribution`
- `AWS::CloudWatch::AnomalyDetector`
- `AWS::CloudWatch::InsightRule`
- `AWS::CodeBuild::Project`
- `AWS::CodeBuild::ReportGroup`
- `AWS::CodeBuild::SourceCredential`
- `AWS::CodeCommit::Repository`
- `AWS::CodeDeploy::DeploymentGroup`
- `AWS::CodeStar::GitHubRepository`
- `AWS::Config::ConfigurationRecorder`
- `AWS::Config::DeliveryChannel`
- `AWS::Config::OrganizationConfigRule`
- `AWS::Config::RemediationConfiguration`
- `AWS::DAX::Cluster`
- `AWS::DAX::ParameterGroup`
- `AWS::DAX::SubnetGroup`
- `AWS::DirectoryService::MicrosoftAD`
- `AWS::DLM::LifecyclePolicy`
- `AWS::DMS::Certificate`

- `AWS::DMS::Endpoint`
- `AWS::DMS::EventSubscription`
- `AWS::DMS::ReplicationInstance`
- `AWS::DMS::ReplicationSubnetGroup`
- `AWS::DMS::ReplicationTask`
- `AWS::DocDB::DBCluster`
- `AWS::DocDB::DBClusterParameterGroup`
- `AWS::DocDB::DBInstance`
- `AWS::DocDB::DBSubnetGroup`
- `AWS::DocDB::EventSubscription`
- `AWS::EC2::ClientVpnAuthorizationRule`
- `AWS::EC2::ClientVpnEndpoint`
- `AWS::EC2::ClientVpnRoute`
- `AWS::EC2::ClientVpnTargetNetworkAssociation`
- `AWS::EC2::NetworkInterfacePermission`
- `AWS::ElastiCache::CacheCluster`
- `AWS::ElastiCache::ReplicationGroup`
- `AWS::ElastiCache::SecurityGroup`
- `AWS::ElastiCache::SecurityGroupIngress`
- `AWS::ElasticLoadBalancing::LoadBalancer`
- `AWS::ElasticLoadBalancingV2::ListenerCertificate`
- `AWS::Elasticsearch::Domain`
- `AWS::EMR::Cluster`
- `AWS::EMR::InstanceFleetConfig`
- `AWS::EMR::InstanceGroupConfig`
- `AWS::FSx::FileSystem`
- `AWS::FSx::Snapshot`
- `AWS::FSx::StorageVirtualMachine`
- `AWS::FSx::Volume`

- `AWS::Glue::Classifier`
- `AWS::Glue::Connection`
- `AWS::Glue::CustomEntityType`
- `AWS::Glue::DataCatalogEncryptionSettings`
- `AWS::Glue::DataQualityRuleset`
- `AWS::Glue::DevEndpoint`
- `AWS::Glue::MLTransform`
- `AWS::Glue::Partition`
- `AWS::Glue::SecurityConfiguration`
- `AWS::Glue::Table`
- `AWS::Glue::TableOptimizer`
- `AWS::Glue::Workflow`
- `AWS::Greengrass::ConnectorDefinition`
- `AWS::Greengrass::ConnectorDefinitionVersion`
- `AWS::Greengrass::CoreDefinition`
- `AWS::Greengrass::CoreDefinitionVersion`
- `AWS::Greengrass::DeviceDefinition`
- `AWS::Greengrass::DeviceDefinitionVersion`
- `AWS::Greengrass::FunctionDefinition`
- `AWS::Greengrass::FunctionDefinitionVersion`
- `AWS::Greengrass::Group`
- `AWS::Greengrass::GroupVersion`
- `AWS::Greengrass::LoggerDefinition`
- `AWS::Greengrass::LoggerDefinitionVersion`
- `AWS::Greengrass::ResourceDefinition`
- `AWS::Greengrass::ResourceDefinitionVersion`
- `AWS::Greengrass::SubscriptionDefinition`
- `AWS::Greengrass::SubscriptionDefinitionVersion`
- `AWS::IAM::AccessKey`

- `AWS::IAM::UserToGroupAddition`
- `AWS::IoT::PolicyPrincipalAttachment`
- `AWS::IoT::ThingPrincipalAttachment`
- `AWS::IoTThingsGraph::FlowTemplate`
- `AWS::KinesisAnalytics::Application`
- `AWS::KinesisAnalytics::ApplicationOutput`
- `AWS::KinesisAnalytics::ApplicationReferenceDataSource`
- `AWS::KinesisAnalyticsV2::ApplicationCloudWatchLoggingOption`
- `AWS::KinesisAnalyticsV2::ApplicationOutput`
- `AWS::KinesisAnalyticsV2::ApplicationReferenceDataSource`
- `AWS::LakeFormation::DataLakeSettings`
- `AWS::LakeFormation::Permissions`
- `AWS::LakeFormation::Resource`
- `AWS::ManagedBlockchain::Member`
- `AWS::ManagedBlockchain::Node`
- `AWS::MediaConvert::JobTemplate`
- `AWS::MediaConvert::Preset`
- `AWS::MediaConvert::Queue`
- `AWS::MediaLive::Channel`
- `AWS::MediaLive::Input`
- `AWS::MediaLive::InputSecurityGroup`
- `AWS::MediaStore::Container`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`

- `AWS::Pinpoint::ADMChannel`
- `AWS::Pinpoint::APNSChannel`
- `AWS::Pinpoint::APNSSandboxChannel`
- `AWS::Pinpoint::APNSVoipChannel`
- `AWS::Pinpoint::APNSVoipSandboxChannel`
- `AWS::Pinpoint::App`
- `AWS::Pinpoint::ApplicationSettings`
- `AWS::Pinpoint::BaiduChannel`
- `AWS::Pinpoint::Campaign`
- `AWS::Pinpoint::EmailChannel`
- `AWS::Pinpoint::EmailTemplate`
- `AWS::Pinpoint::EventStream`
- `AWS::Pinpoint::GCMChannel`
- `AWS::Pinpoint::PushTemplate`
- `AWS::Pinpoint::Segment`
- `AWS::Pinpoint::SMSChannel`
- `AWS::Pinpoint::SmsTemplate`
- `AWS::Pinpoint::VoiceChannel`
- `AWS::PinpointEmail::ConfigurationSet`
- `AWS::PinpointEmail::ConfigurationSetEventDestination`
- `AWS::PinpointEmail::DedicatedIpPool`
- `AWS::PinpointEmail::Identity`
- `AWS::QLDB::Ledger`
- `AWS::RDS::DBSecurityGroup`
- `AWS::RDS::DBSecurityGroupIngress`
- `AWS::Redshift::ClusterSecurityGroup`
- `AWS::Redshift::ClusterSecurityGroupIngress`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`

- `AWS::SageMaker::CodeRepository`
- `AWS::SageMaker::EndpointConfig`
- `AWS::SageMaker::Model`
- `AWS::SageMaker::NotebookInstance`
- `AWS::SageMaker::NotebookInstanceLifecycleConfig`
- `AWS::SageMaker::Workteam`
- `AWS::SDB::Domain`
- `AWS::ServiceCatalog::AcceptedPortfolioShare`
- `AWS::ServiceCatalog::LaunchRoleConstraint`
- `AWS::ServiceCatalog::Portfolio`
- `AWS::ServiceCatalog::StackSetConstraint`
- `AWS::ServiceDiscovery::HttpNamespace`
- `AWS::ServiceDiscovery::Instance`
- `AWS::ServiceDiscovery::PrivateDnsNamespace`
- `AWS::ServiceDiscovery::PublicDnsNamespace`
- `AWS::ServiceDiscovery::Service`
- `AWS::SES::ReceiptFilter`
- `AWS::SES::ReceiptRule`
- `AWS::SES::ReceiptRuleSet`
- `AWS::SSM::MaintenanceWindow`
- `AWS::SSM::MaintenanceWindowTarget`
- `AWS::SSM::MaintenanceWindowTask`
- `AWS::WAF::ByteMatchSet`
- `AWS::WAF::IPSet`
- `AWS::WAF::Rule`
- `AWS::WAF::SizeConstraintSet`
- `AWS::WAF::SqlInjectionMatchSet`
- `AWS::WAF::WebACL`
- `AWS::WAF::XssMatchSet`

- `AWS::WAFRegional::ByteMatchSet`
- `AWS::WAFRegional::GeoMatchSet`
- `AWS::WAFRegional::IPSet`
- `AWS::WAFRegional::RateBasedRule`
- `AWS::WAFRegional::RegexPatternSet`
- `AWS::WAFRegional::Rule`
- `AWS::WAFRegional::SizeConstraintSet`
- `AWS::WAFRegional::SqlInjectionMatchSet`
- `AWS::WAFRegional::WebACL`
- `AWS::WAFRegional::WebACLAssociation`
- `AWS::WAFRegional::XssMatchSet`
- `AWS::WorkSpaces::Workspace`
- `AWS::AmazonMQ::ConfigurationAssociation`
- `AWS::ApiGateway::DomainNameAccessAssociation`
- `AWS::AppConfig::ExtensionAssociation`
- `AWS::AppStream::ApplicationEntitlementAssociation`
- `AWS::AppStream::ApplicationFleetAssociation`
- `AWS::AppSync::DomainNameApiAssociation`
- `AWS::AppSync::SourceApiAssociation`
- `AWS::CleanRooms::ConfiguredTableAssociation`
- `AWS::CleanRooms::IdNamespaceAssociation`
- `AWS::CodeGuruReviewer::RepositoryAssociation`
- `AWS::Cognito::IdentityPoolRoleAttachment`
- `AWS::Cognito::UserPoolRiskConfigurationAttachment`
- `AWS::Cognito::UserPoolUICustomizationAttachment`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Connect::IntegrationAssociation`
- `AWS::Deadline::QueueFleetAssociation`
- `AWS::Deadline::QueueLimitAssociation`

- `AWS::EC2::EIPAssociation`
- `AWS::EC2::EnclaveCertificateIamRoleAssociation`
- `AWS::EC2::GatewayRouteTableAssociation`
- `AWS::EC2::IPAMResourceDiscoveryAssociation`
- `AWS::EC2::IpPoolRouteTableAssociation`
- `AWS::EC2::LocalGatewayRouteTableVPCAssociation`
- `AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation`
- `AWS::EC2::NetworkInterfaceAttachment`
- `AWS::EC2::RouteServerAssociation`
- `AWS::EC2::SecurityGroupVpcAssociation`
- `AWS::EC2::SubnetNetworkAclAssociation`
- `AWS::EC2::SubnetRouteTableAssociation`
- `AWS::EC2::TransitGatewayAttachment`
- `AWS::EC2::TransitGatewayMulticastDomainAssociation`
- `AWS::EC2::TransitGatewayPeeringAttachment`
- `AWS::EC2::TransitGatewayRouteTableAssociation`
- `AWS::EC2::TransitGatewayVpcAttachment`
- `AWS::EC2::VPCDHCPOptionsAssociation`
- `AWS::EC2::VPCGatewayAttachment`
- `AWS::EC2::VolumeAttachment`
- `AWS::ECS::ClusterCapacityProviderAssociations`
- `AWS::EKS::PodIdentityAssociation`
- `AWS::FSx::DataRepositoryAssociation`
- `AWS::FSx::S3AccessPointAttachment`
- `AWS::GlobalAccelerator::CrossAccountAttachment`
- `AWS::LakeFormation::TagAssociation`
- `AWS::NetworkFirewall::VpcEndpointAssociation`
- `AWS::NetworkManager::ConnectAttachment`
- `AWS::NetworkManager::CustomerGatewayAssociation`

- `AWS::NetworkManager::DirectConnectGatewayAttachment`
- `AWS::NetworkManager::LinkAssociation`
- `AWS::NetworkManager::SiteToSiteVpnAttachment`
- `AWS::NetworkManager::TransitGatewayRouteTableAttachment`
- `AWS::NetworkManager::VpcAttachment`
- `AWS::Notifications::ChannelAssociation`
- `AWS::Notifications::ManagedNotificationAccountContactAssociation`
- `AWS::Notifications::ManagedNotificationAdditionalChannelAssociation`
- `AWS::Notifications::OrganizationalUnitAssociation`
- `AWS::ResourceExplorer2::DefaultViewAssociation`
- `AWS::Route53Profiles::ProfileAssociation`
- `AWS::Route53Profiles::ProfileResourceAssociation`
- `AWS::Route53Resolver::FirewallRuleGroupAssociation`
- `AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation`
- `AWS::Route53Resolver::ResolverRuleAssociation`
- `AWS::SSM::Association`
- `AWS::SecretsManager::SecretTargetAttachment`
- `AWS::SecurityHub::PolicyAssociation`
- `AWS::ServiceCatalog::PortfolioPrincipalAssociation`
- `AWS::ServiceCatalog::PortfolioProductAssociation`
- `AWS::ServiceCatalog::ServiceActionAssociation`
- `AWS::ServiceCatalog::TagOptionAssociation`
- `AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation`
- `AWS::ServiceCatalogAppRegistry::ResourceAssociation`
- `AWS::VpcLattice::ServiceNetworkResourceAssociation`
- `AWS::VpcLattice::ServiceNetworkServiceAssociation`
- `AWS::VpcLattice::ServiceNetworkVpcAssociation`
- `AWS::WAFv2::WebACLAssociation`
- `AWS::Wisdom::AssistantAssociation`

- `AWS::WorkspacesInstances::VolumeAssociation`
- `AWS::IAM::Policy`
- `AWS::SNS::TopicPolicy`
- `AWS::SQS::QueuePolicy`
- `AWS::EC2::NetworkAclEntry`
- `AWS::EC2::VPNGatewayRoutePropagation`
- `AWS::CloudFormation::Stack`
- `AWS::CloudWatch::MetricStream`
- `AWS::WorkSpaces::ConnectionAlias`
- `AWS::IoT::ProvisioningTemplate`
- `AWS::MediaPackage::Channel`
- `AWS::CloudFront::OriginRequestPolicy`
- `AWS::Route53Resolver::ResolverQueryLoggingConfig`
- `AWS::NetworkManager::TransitGatewayRegistration`
- `AWS::ImageBuilder::Image`
- `AWS::Config::ConformancePack`
- `AWS::S3::AccessPoint`
- `AWS::CodeStarConnections::Connection`
- `AWS::CloudFront::CachePolicy`
- `AWS::FMS::NotificationChannel`
- `AWS::ImageBuilder::InfrastructureConfiguration`
- `AWS::Detective::Graph`
- `AWS::EC2::CarrierGateway`
- `AWS::CloudWatch::CompositeAlarm`
- `AWS::CodeArtifact::Repository`
- `AWS::GroundStation::DataflowEndpointGroup`
- `AWS::ElasticLoadBalancingV2::Listener`
- `AWS::ImageBuilder::ImageRecipe`
- `AWS::NetworkManager::Device`
- `AWS::Kendra::DataSource`

- `AWS::Timestream::Database`
- `AWS::CodeGuruProfiler::ProfilingGroup`
- `AWS::Lambda::EventSourceMapping`
- `AWS::ECR::Repository`
- `AWS::WAFv2::IPSet`
- `AWS::GameLift::Alias`
- `AWS::IoTSiteWise::Asset`
- `AWS::OpsWorksCM::Server`
- `AWS::IoT::Authorizer`
- `AWS::WAFv2::RuleGroup`
- `AWS::NetworkManager::Site`
- `AWS::ResourceGroups::Group`
- `AWS::MediaPackage::PackagingConfiguration`
- `AWS::ImageBuilder::ImagePipeline`
- `AWS::ECS::TaskDefinition`
- `AWS::Macie::CustomDataIdentifier`
- `AWS::MediaPackage::OriginEndpoint`
- `AWS::Logs::LogGroup`
- `AWS::CodeArtifact::Domain`
- `AWS::Kendra::Faq`
- `AWS::ECS::TaskSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::ECS::Cluster`
- `AWS::SSO::Assignment`
- `AWS::GlobalAccelerator::Listener`
- `AWS::ServiceCatalog::CloudFormationProvisionedProduct`
- `AWS::RDS::DBProxy`
- `AWS::EC2::FlowLog`
- `AWS::ImageBuilder::Component`
- `AWS::CloudFront::RealtimeLogConfig`

- `AWS::NetworkManager::GlobalNetwork`
- `AWS::RDS::DBProxyTargetGroup`
- `AWS::WAFv2::WebACL`
- `AWS::IVS::StreamKey`
- `AWS::IVS::PlaybackKeyPair`
- `AWS::Macie::Session`
- `AWS::Route53::HealthCheck`
- `AWS::Synthetics::Canary`
- `AWS::Lambda::CodeSigningConfig`
- `AWS::EFS::AccessPoint`
- `AWS::Timestream::Table`
- `AWS::MediaPackage::PackagingGroup`
- `AWS::ECS::PrimaryTaskSet`
- `AWS::Config::ConfigurationAggregator`
- `AWS::GroundStation::Config`
- `AWS::IoTSiteWise::AssetModel`
- `AWS::SES::ConfigurationSet`
- `AWS::ImageBuilder::DistributionConfiguration`
- `AWS::Config::OrganizationConformancePack`
- `AWS::EC2::LocalGatewayRoute`
- `AWS::KMS::Key`
- `AWS::Detective::MemberInvitation`
- `AWS::EKS::FargateProfile`
- `AWS::MediaPackage::Asset`
- `AWS::GlobalAccelerator::EndpointGroup`
- `AWS::Macie::FindingsFilter`
- `AWS::IoT::Certificate`
- `AWS::SageMaker::MonitoringSchedule`
- `AWS::IVS::Channel`
- `AWS::Kendra::Index`

- `AWS::EventSchemas::RegistryPolicy`
- `AWS::KinesisFirehose::DeliveryStream`
- `AWS::GlobalAccelerator::Accelerator`
- `AWS::EC2::PrefixList`
- `AWS::GameLift::GameServerGroup`
- `AWS::NetworkManager::Link`
- `AWS::EFS::FileSystem`
- `AWS::Route53::HostedZone`
- `AWS::GroundStation::MissionProfile`
- `AWS::KMS::Alias`
- `AWS::FMS::Policy`
- `AWS::SSO::PermissionSet`
- `AWS::StepFunctions::StateMachine`
- `AWS::QLDB::Stream`
- `AWS::IoTSiteWise::Gateway`
- `AWS::ECS::Service`
- `AWS::ECS::CapacityProvider`
- `AWS::EC2::SecurityGroup`
- `AWS::EC2::SecurityGroupIngress`
- `AWS::EC2::SecurityGroupEgress`
- `AWS::EC2::EC2Fleet`
- `AWS::IAM::Group`
- `AWS::IAM::Role`
- `AWS::IAM::User`
- `AWS::ApiGateway::GatewayResponse`
- `AWS::S3::BucketPolicy`
- `AWS::SNS::Topic`
- `AWS::SNS::Subscription`
- `AWS::RDS::DBInstance`
- `AWS::RDS::DBParameterGroup`

- `AWS::RDS::DBCluster`
- `AWS::RDS::DBClusterParameterGroup`
- `AWS::RDS::DBSubnetGroup`
- `AWS::RDS::EventSubscription`
- `AWS::RDS::GlobalCluster`
- `AWS::RDS::OptionGroup`
- `AWS::Neptune::DBInstance`
- `AWS::Neptune::DBParameterGroup`
- `AWS::Neptune::DBCluster`
- `AWS::Neptune::DBClusterParameterGroup`
- `AWS::Neptune::DBSubnetGroup`
- `AWS::Redshift::Cluster`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::Redshift::ClusterSubnetGroup`
- `AWS::Redshift::EndpointAccess`
- `AWS::Redshift::EndpointAuthorization`
- `AWS::Redshift::EventSubscription`
- `AWS::Redshift::ScheduledAction`
- `AWS::ElastiCache::SubnetGroup`

Stapel 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-Bucket speichern.

Für Ressourceneigenschaften, die Aktualisierungen nicht unterstützen, müssen Sie die aktuellen Werte beibehalten. Verwenden Sie Änderungssätze, 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](#).

Wenn Sie einen Stack aktualisieren, CloudFormation können Ressourcen unterbrochen oder aktualisierte Ressourcen ersetzt werden, je nachdem, welche Eigenschaften Sie aktualisieren. Weitere Informationen zum Verhalten von Ressourcenaktualisierungen finden Sie unter [Verstehen des Aktualisierungsverhaltens von Stack-Ressourcen](#).

So aktualisieren Sie einen -Stack (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie auf der Seite Stapel den laufenden Stapel, den Sie aktualisieren möchten.
4. Klicken Sie im Detailbereich des Stacks auf Update (Aktualisieren).
5. Wenn Sie die Stapelvorlage nicht geändert haben, wählen Sie Vorhandene Vorlage verwenden und wählen Sie dann Weiter.

Wenn Sie die Vorlage geändert haben, wählen Sie Vorhandene Vorlage ersetzen und geben 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

- Die Buckets sind für jeden mit Amazon S3-Berechtigungen in Ihrem AWS-Konto zugänglich.
- 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 [Einstellen des standardmäßigen serverseitigen Verschlüsselungsverhaltens für Amazon S3-Buckets](#) im Amazon Simple Storage Service User Guide.

- 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 Versionskontrolle haben, können Sie eine bestimmte Version der Vorlage angeben, indem Sie `?versionId=version-id` an die URL anhängen. Weitere Informationen finden Sie unter [Arbeiten mit Objekten in einem versionierungsfähigen Bucket](#) im Amazon Simple Storage Service User Guide.

Wenn Syntaxprobleme entdeckt werden, gibt die Konsole Fehlermeldungen aus, mit deren Hilfe Sie die Vorlage korrigieren können.

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 über die Verwendung von NoEcho zur Maskierung sensibler Informationen sowie über die Verwendung dynamischer Parameter zur Verwaltung von Geheimnissen finden Sie in der bewährten Vorgehensweise [Keine Anmeldeinformationen in Vorlagen einbetten](#).

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 [Stapeloptionen konfigurieren](#).

8. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie für Fähigkeitendie 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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).
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 dadurch 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 Stapel hat jetzt den Status UPDATE_IN_PROGRESS. Nachdem CloudFormation die Aktualisierung des Stacks erfolgreich abgeschlossen wurde, wird der Stack-Status auf gesetztUPDATE_COMPLETE.

Schlägt die Stack-Aktualisierung fehl CloudFormation, werden die Änderungen automatisch rückgängig gemacht und der Stack-Status wird auf gesetztUPDATE_ROLLBACK_COMPLETE.

 Note

Sie können eine Aktualisierung abbrechen, während sie sich im Status UPDATE_IN_PROGRESS befindet. Weitere Informationen finden Sie unter [Eine Stapelaktualisierung abbrechen](#).

So aktualisieren Sie einen Stack über die Befehlszeile

Verwenden Sie einen der folgenden Befehle:

- [update-stack](#) (AWS CLI)
- [Update-CFNStack](#) (AWS Tools for Windows PowerShell)

Beispiele für die Verwendung der Befehlszeile zur Aktualisierung eines Stacks finden Sie unter [Beispiele für CloudFormation Stack-Operationsbefehle für AWS CLI und PowerShell](#).

Eine Stapelaktualisierung 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

- [So brechen Sie eine Stack-Aktualisierung ab \(Konsole\)](#)
- [So brechen Sie eine Stapelaktualisierung ab \(AWS CLI\)](#)

So brechen Sie eine Stack-Aktualisierung ab (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die 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, der gerade aktualisiert wird. Sein Status muss UPDATE_IN_PROGRESS sein.
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.

So brechen Sie eine Stapelaktualisierung ab (AWS CLI)

Verwenden Sie den Befehl [cancel-update-stack](#), um eine Aktualisierung zu stornieren. Weitere Informationen finden Sie unter [Eine Stapelaktualisierung abbrechen](#).

Einen Stack aus der CloudFormation Konsole löschen

Wenn Sie die Ressourcen in einem Stapel nicht mehr benötigen, können Sie den gesamten Stapel 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](#).

So löschen Sie einen Stapel (Konsole)

1. Melden Sie sich bei <https://console.aws.amazon.com/cloudformation> an AWS-Managementkonsole und öffnen Sie die 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 Stapel den Stapel, 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.

So erzwingen Sie das Löschen eines Stapels (Konsole)

Eine Stapellöschung kann fehlschlagen, weil eine Ressource im Stapel nicht gelöscht werden kann. Schlägt beispielsweise CloudFormation das Löschen einer Ressource fehl, von der auch ein anderer Stack abhängt. Alle Ressourcen, die nicht gelöscht worden sind bleiben erhalten, bis Sie den Stack erfolgreich löschen können. Wenn der Löschvorgang 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 in der Detailansicht des Stapels Löschen wiederholen.
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 der erzwungenen Stapellöschung beibehalten möchten.
 - Löschen dieses gesamten Stapels erzwingen: Diese Option behält alle Ressourcen bei, die nicht gelöscht werden konnten, und behält die Abhängigkeiten von diesen Ressourcen bei.
4. Wählen Sie Löschen um den Löschvorgang mit Ihren Auswahlen zu beginnen.

So überprüfen Sie zurückgehaltene Ressourcen (Konsole)

Nach dem Löschen des Stapels können Sie die Ressourcen, die erhalten geblieben sind, in der Konsole einsehen.

1. Wählen Sie in der Stapelliste den Filterstatus und wählen Sie Gelöscht.
2. Wählen Sie den gelöschten Stapel.
3. Wählen Sie die Registerkarte Resources (Ressourcen) aus.
4. Alle einbehaltenen Ressourcen zeigen den DELETE_SKIPPED Status.
5. Wählen Sie die zurückbehaltene Ressource, die Sie überprüfen möchten.

So löschen Sie einen Stapel über die Befehlszeile

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 Stapels finden Sie unter [Beispiele für CloudFormation Stack-Operationsbefehle für AWS CLI und PowerShell](#).

Zugehörige Ressourcen

Hilfe bei der Behebung von Stacklöschungsfehlern finden Sie im Thema [Stack-Fehler löschen-Fehlerbehebung](#).

Informationen zum Schutz von Stapeln 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-Managementkonsole und öffnen Sie die 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 Stapel aus der Dropdown-Liste Filterstatus die Option Gelöscht .

CloudFormation listet alle Ihre Stapel mit dem Status auf. DELETE_COMPLETE

Weitere Informationen finden Sie auch unter

- [Einen Stack aus der CloudFormation Konsole löschen](#)
- [Stack-Informationen von der CloudFormation Konsole aus anzeigen](#)

Fortschritt des Stapels 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](#)
- [Stack-Ereignisse nach Vorgang anzeigen](#)
- [Sehen Sie sich einen Zeitplan für eine CloudFormation Stack-Bereitstellung an](#)
- [Ereignisse bei der CloudFormation Stackerstellung verstehen](#)
- [Überwachen Sie den Fortschritt einer Stack-Aktualisierung](#)
- [Rollback einer Aktualisierung fortsetzen](#)
- [Ermitteln Sie die Ursache eines Stapelausfalls](#)
- [Wählen Sie aus, wie mit Fehlern bei der Bereitstellung von Ressourcen umgegangen werden soll](#)

CloudFormation Stack-Ereignisse anzeigen

Sie können Stack-Ereignisse anzeigen, um den Fortschritt und den 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 abläuft.

Themen

- [Stack-Ereignisse anzeigen \(Konsole\)](#)
- [Stack-Ereignisse anzeigen \(AWS CLI\)](#)
- [Stack-Statuscodes](#)

Stack-Ereignisse anzeigen (Konsole)

So zeigen Sie Stack-Ereignisse an

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms den AWS-Region , in dem Sie den Stack 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-Erstellungseignisse, um die Abfolge der Ereignisse nachzuvollziehen, die zur Erstellung des Stacks (oder zu einem Fehler im Fall von Stack-Debugging) geführt haben.

Während Ihr Stapel erstellt wird, wird er auf der Seite Stapel mit dem Status `CREATE_IN_PROGRESS` aufgeführt. 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 einer Stack-Aktualisierung](#).

Stack-Ereignisse anzeigen (AWS CLI)

Alternativ können Sie den [describe-stack-events](#) Befehl verwenden, während der Stack erstellt wird, um Ereignisse so anzuzeigen, wie sie gemeldet wurden.

Der folgende describe-stack-events-Befehl beschreibt die *my-stack*-Stackereignisse.

```
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	Description
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.
ResourceType	Typ der Ressource.
Timestamp	Uhrzeit, zu der das Ereignis aufgetreten ist.
ResourceStatus	Der Status der Ressource, der einer der folgenden Statuscod es 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

Feld	Description
	Der Status <code>DELETE_SKIPPED</code> gilt für Ressourcen, deren Richtlinienattribut zum Löschen "retain" (beibehalten) lautet.
<code>DetailedStatus</code>	Der detaillierte Status des Stapels. Wenn <code>CONFIGURATION_COMPLETE</code> vorhanden ist, ist die Konfigurationsphase der Stapelressourcen abgeschlossen und die Stabilisierung der Ressourcen ist im Gange.
<code>ResourceStatusReason</code>	Weitere Informationen zum Status.

Stack-Statuscodes

Die folgende Tabelle beschreibt die Stack-Statuscodes:

Stapelstatus und optionaler detaillierter Status	Description
<code>CREATE_COMPLETE</code>	Erfolgreiche Erstellung eines oder mehrerer Stacks.
<code>CREATE_IN_PROGRESS</code>	Laufende Erstellung eines oder mehrerer Stacks.
<code>CREATE_FAILED</code>	Fehlgeschlagene Erstellung eines oder mehrerer Stacks. Untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen. Mögliche Gründe für eine fehlgeschlagene Erstellung sind unzureichende Berechtigungen für die Arbeit mit allen Ressourcen im Stack, von einem AWS Dienst abgelehnte Parameterwerte oder ein Timeout bei der Ressourcenerstellung.
<code>DELETE_COMPLETE</code>	Erfolgreiche Löschung eines oder mehrerer Stacks. Gelöschte Stacks bleiben 90 Tage lang erhalten und anzeigbar.
<code>DELETE_FAILED</code>	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

Stapelstatus und optionaler detaillierter Status	Description
	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 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.

Stapelstatus und optionaler detaillierter Status	Description
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.
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.

Stapelstatus und optionaler detaillierter Status	Description
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 <code>resource import</code> unterstützen.
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 Vorlagenkonfiguration zurückgesetzt.

Stack-Ereignisse nach Vorgang anzeigen

Sie können Stack-Ereignisse nach Vorgängen gruppiert anzeigen, um die Reihenfolge und den Umfang der an Ihrem Stack vorgenommenen Änderungen besser zu verstehen. Die

vorgangsbasierte Gruppierung hilft Ihnen dabei, verwandte Ereignisse gemeinsam nachzuverfolgen, sodass Sie den Fortschritt leichter überwachen und Probleme beheben können, wenn sie auftreten.

Jedem Stack-Vorgang (Erstellen, Aktualisieren, Löschen, Rollback) wird eine eindeutige Vorgangs-ID zugewiesen, die alle zugehörigen Ereignisse gruppiert. Auf diese Weise können Sie sich auf bestimmte Operationen konzentrieren und die Hauptursache von Fehlern schnell identifizieren.

Themen

- [Voraussetzungen](#)
- [Stack-Ereignisse nach Vorgang anzeigen \(Konsole\)](#)
- [Stack-Ereignisse nach Vorgang \(AWS CLI\) anzeigen](#)
- [Stack-Statuscodes](#)

Voraussetzungen

Um die `DescribeEvents` API verwenden zu können, benötigen Sie die erforderlichen IAM-Berechtigungen für `DescribeEvents`.

Stack-Ereignisse nach Vorgang anzeigen (Konsole)

Um Stack-Ereignisse nach Vorgängen gruppiert anzuzeigen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die 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 den Stack erstellt haben.
3. Wählen Sie auf der Seite Stacks 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. Ereignisse werden automatisch nach der Operation-ID gruppiert. Jeder Vorgang wird als erweiterbarer Abschnitt angezeigt, in dem der Vorgangstyp, der Status und der Zeitstempel angezeigt werden.
6. Klicken Sie auf eine Vorgangs-ID, um eine detaillierte Ansicht zu öffnen, in der nur die Ereignisse angezeigt werden, die sich auf diesen bestimmten Vorgang beziehen.

7. Aktivieren Sie in der Vorgangsdetailansicht das Kontrollkästchen Nur Fehler anzeigen, um nur fehlgeschlagene Ereignisse für die Ursachenanalyse anzuzeigen.

CloudFormation aktualisiert die Stack-Ereignisse automatisch jede Minute. Das Badge „Neue Ereignisse verfügbar“ wird angezeigt, wenn neue Stack-Ereignisse auftreten. Wählen Sie das Aktualisierungssymbol, um diese Ereignisse in die Liste zu laden.

Durch die Anzeige von Stack-Ereignissen, die nach Vorgängen gruppiert sind, können Sie die Reihenfolge der Ereignisse für jeden Vorgang nachvollziehen und schnell erkennen, welcher spezifische Vorgang Probleme verursacht hat (wenn Sie Ihren Stack debuggen).

Während Ihr Stack-Vorgang ausgeführt wird, wird er mit dem Status `CREATE_IN_PROGRESS`, `UPDATE_IN_PROGRESS`, oder `DELETE_IN_PROGRESS` aufgeführt. Nachdem Ihr Vorgang erfolgreich abgeschlossen wurde, ändert sich sein Status in `CREATE_COMPLETE`, `UPDATE_COMPLETE`, oder `DELETE_COMPLETE`.

Weitere Informationen erhalten Sie unter [Ereignisse bei der CloudFormation Stackerstellung verstehen](#) und [Überwachen Sie den Fortschritt einer Stack-Aktualisierung](#).

Stack-Ereignisse nach Vorgang (AWS CLI) anzeigen

Sie können den `describe-events` Befehl mit Operations-ID-Filterung verwenden, um Ereignisse für bestimmte Operationen anzuzeigen.

Letzte Operation abrufen IDs

Die über die Describe-Stacks-API verfügbare Stacks-Beschreibung enthält jetzt LastOperations Informationen über die letzten Operationen IDs und deren Typen. Auf diese Weise können Sie schnell erkennen, welche Operationen stattgefunden haben und welchen aktuellen Status sie haben, ohne die Ereignisprotokolle analysieren zu müssen.

```
aws cloudformation describe-stacks --stack-name MyStack
```

Die folgende Beispielantwort zeigt, dass der letzte Vorgang ein Rollback nach einem fehlgeschlagenen Aktualisierungsvorgang war.

```
{
  "Stacks": [
    {
```

```

    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/07580010-bb79-11f0-8f6c-0289bb5c804f",
    "StackName": "MyStack",
    "Description": "A simple CloudFormation template to create an S3 bucket.",

    "CreationTime": "2025-11-07T01:28:13.778000+00:00",
    "LastUpdatedTime": "2025-11-07T01:43:39.838000+00:00",
    "RollbackConfiguration": {},
    "StackStatus": "UPDATE_ROLLBACK_COMPLETE",
    "DisableRollback": false,
    "NotificationARNs": [],
    "Tags": [],
    "EnableTerminationProtection": false,
    "DriftInformation": {
        "StackDriftStatus": "NOT_CHECKED"
    },
    "LastOperations": [
        {
            "OperationType": "ROLLBACK",
            "OperationId": "d0f12313-7bdb-414d-a879-828a99b36f29"
        },
        {
            "OperationType": "UPDATE_STACK",
            "OperationId": "1c211b5a-4538-4dc9-bfed-e07734371e57"
        }
    ]
}
]
}
}

```

Filtert Ereignisse nach der Vorgangs-ID

Der folgende `describe-events` Befehl beschreibt Ereignisse für eine bestimmte Vorgangs-ID:

```

aws cloudformation describe-events \
  --operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57

```

Verwenden Sie den `--filter FailedEvents=true` Parameter, um nur fehlgeschlagene Ereignisse zur Problembehandlung anzuzeigen:

```

aws cloudformation describe-events \
  --operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57 \

```

```
--filter FailedEvents=true
```

Mit der neuen Funktion zum Filtern von Vorgangs-IDs können Sie sich auf bestimmte Operationen und die damit verbundenen Ereignisse konzentrieren. Dies ist besonders nützlich für:

- **Behebung bestimmter Fehler:** Isolieren Sie die Ereignisse eines fehlgeschlagenen Vorgangs, um zu verstehen, was schief gelaufen ist.
- **Überwachung lang andauernder Operationen:** Verfolgen Sie den Fortschritt komplexer Updates oder umfangreicher Stack-Bereitstellungen.
- **Änderungen überprüfen:** Überprüfen Sie alle Ereignisse, die mit einem bestimmten Aktualisierungsvorgang verbunden sind.
- **Ursachenanalyse:** Verwenden Sie den Fehlerfilter, um schnell die Ursache von Bereitstellungsproblemen zu ermitteln.

Stack-Statuscodes

In der folgenden Tabelle werden die Felder beschrieben, die vom `describe-events` Befehl zurückgegeben werden, wenn die Operations-ID-Filterung verwendet wird:

Feld	Description
EventId	Ereignis-Kennung.
OperationId	Eindeutiger Bezeichner für den Vorgang, der dieses Ereignis generiert hat.
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	Description
ResourceType	Typ der Ressource.
Timestamp	Uhrzeit, zu der das Ereignis aufgetreten ist.
ResourceStatus	Der Status der Ressource (CREATE_COMPLETE UPDATE_FAILED , usw.).
DetailedStatus	Der detaillierte Status des Stapels. Wenn CONFIGURATION_COMPLETE vorhanden ist, ist die Konfigurationsphase der Stapelressourcen abgeschlossen und die Stabilisierung der Ressourcen ist im Gange.
ResourceStatusReason	Weitere Informationen zum Status.

Sehen Sie sich einen Zeitplan für eine CloudFormation Stack-Bereitstellung an

Das Diagramm der Stack-Bereitstellungszeitleiste bietet eine visuelle Darstellung einer Stack-Bereitstellungszeitleiste. Diese Ansicht zeigt die Bereitstellungsstatus für den Stack und jede seiner Ressourcen sowie die Zeitpunkte, zu denen sich jeder Status geändert hat. Der Status der Stapelbereitstellung wird durch eine entsprechende Farbe dargestellt.

Themen

- [Das Diagramm der Zeitleiste für die Stack-Bereitstellung verstehen](#)
- [Anzeigen des Diagramms der Zeitachse für die Stack-Bereitstellung \(Konsole\)](#)

Das Diagramm der Zeitleiste für die Stack-Bereitstellung verstehen

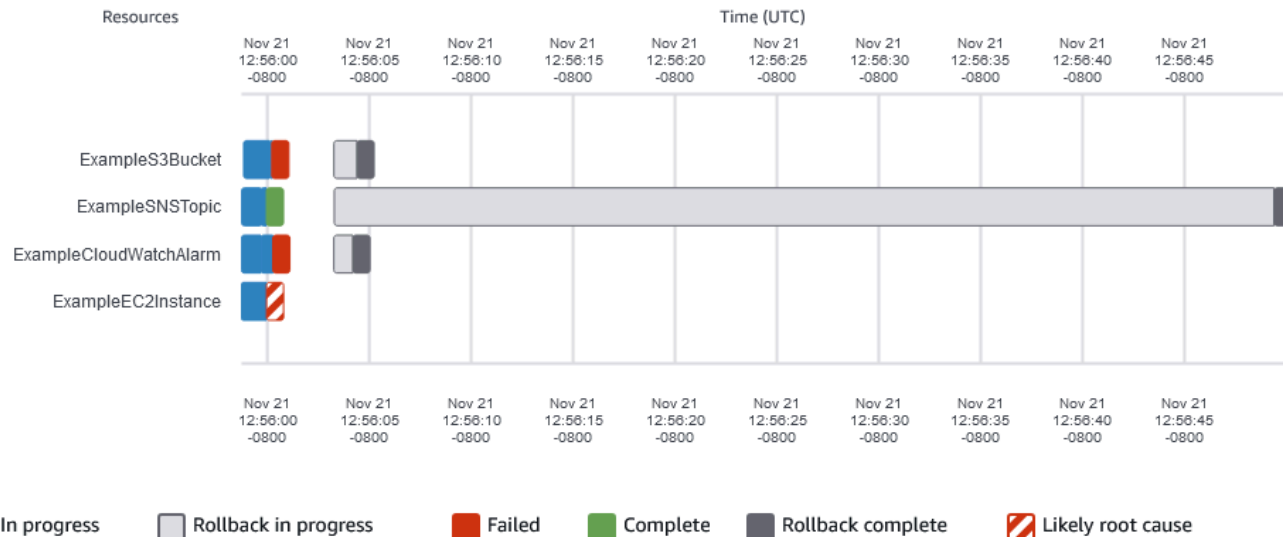
Die folgende Abbildung zeigt das Zeitdiagramm für eine Stack-Bereitstellung, die aufgrund einer fehlgeschlagenen EC2 Amazon-Instance-Ressource fehlgeschlagen ist.

Table view

Timeline view - new

Latest deployment timeline - new[View root cause](#)

This is a timeline view of your latest stack deployment. If you start a new deployment, this view will reset.

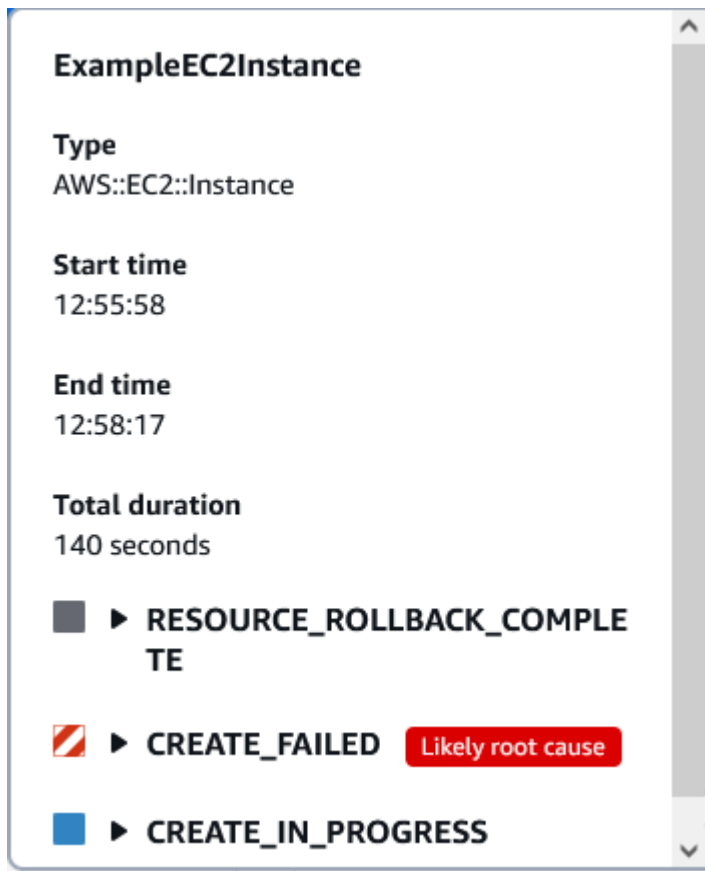


Die Namen der Stack-Ressourcen finden Sie auf der linken Seite des Diagramms, und das Datum und die Uhrzeit im Verhältnis zu den Bereitstellungszeiten finden Sie oben im Diagramm.

Jede Ressource beginnt mit dem Status In Bearbeitung. Die Statusleiste ändert sich bei jeder erfolgreichen Distribution in Complete. Die Statusleiste ändert sich zu Fehlgeschlagen wenn eine Ressource nicht bereitgestellt werden kann. Wenn die Bereitstellung einer Ressource fehlschlägt und auch die Stack-Bereitstellung fehlschlägt, erhält die Ressource, die für den Ausfall der Stack-Bereitstellung verantwortlich ist, den Status Likely root failure.

Nachdem der Stack-Bereitstellungsvorgang fehlggeschlagen ist, beginnt die erfolgreich bereitgestellte Ressource mit dem Rollback und wechselt in den Status Rollback in Bearbeitung. Die Status ändern sich zu Rollback abgeschlossen nachdem die Ressource das Rollback beendet hat.

Wenn Sie die einzelnen Ressourcen auswählen, erhalten Sie detailliertere Informationen über den Zeitplan für die Bereitstellung:



Wenn Sie eine Ressource auswählen, werden der Typ, die Startzeit, die Endzeit und die Gesamtdauer der Bereitstellung angezeigt. In den Dropdown-Menüs unten finden Sie außerdem die Startzeit, Endzeit und Dauer für jeden Einsatzstatus. Wenn die Bereitstellung der Ressource fehlgeschlagen ist, wird ein Fehlergrund angegeben.

Weitere Informationen zum Stack-Status finden Sie unter [Stack-Statuscodes](#).

Anzeigen des Diagramms der Zeitachse für die Stack-Bereitstellung (Konsole)

So zeigen Sie ein Zeitdiagramm zur Stack-Bereitstellung an:

1. Öffnen Sie die 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 Zeitachsendiagramm um das Zeitachsendiagramm für Ihren Stapel anzuzeigen.

Ereignisse bei der CloudFormation Stackerstellung verstehen

Während der Stack-Bereitstellung finden mehrere Ereignisse statt, um die in der Stack-Vorlage definierten Ressourcen zu erstellen, zu konfigurieren und zu validieren. Wenn Sie diese Ereignisse verstehen, können Sie den Prozess der Stack-Erstellung optimieren und die Bereitstellung rationalisieren.

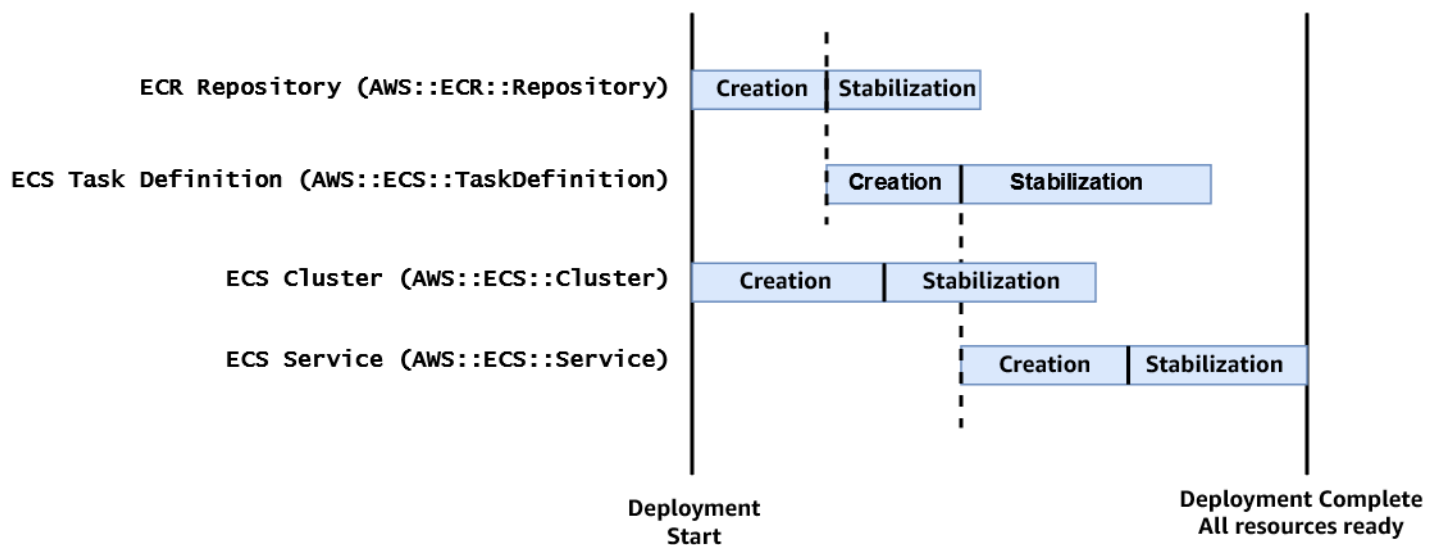
- Ressourcenerstellungsereignisse - Wenn jede Ressource den Erstellungsprozess beginnt, wird ein Status von `CREATE_IN_PROGRESS` Ereignis gesetzt. Dieses Ereignis zeigt an, dass die Ressource gerade bereitgestellt wird.
- Eventuelle Konsistenzprüfung - Ein beträchtlicher Teil der Zeit für die Stackerstellung wird für eine eventuelle Konsistenzprüfung der vom Stack erstellten Ressourcen verwendet. 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 Phase der eventuellen Konsistenzprüfung der Bereitstellung abgeschlossen hat, wird ein Ereignis Detaillierter Status von `CONFIGURATION_COMPLETE` gesetzt.
- Ereignis Ressourcenerstellung abgeschlossen - Nachdem die Ressource erstellt und wie angegeben konfiguriert wurde und die Konfiguration mit den Angaben in der Vorlage übereinstimmt, wird das Ereignis Status von `CREATE_COMPLETE` gesetzt.

Sie können das Ereignis `CONFIGURATION_COMPLETE` nutzen, um Ihren Stack-Erstellungsprozess in Szenarien zu rationalisieren, in denen eine Konsistenzprüfung der Ressourcen nicht erforderlich ist, wie z.B. bei der Validierung einer Vorproduktions-Stack-Konfiguration oder bei der stapelübergreifenden Bereitstellung. Sie können dieses Ereignis auf verschiedene Weise nutzen. Sie können es beispielsweise als visuelles Signal verwenden, um das Warten auf das Ende der Konsistenzprüfung von Ressourcen oder Stapeln zu überspringen. Oder Sie könnten damit einen automatisierten Mechanismus mit kontinuierlicher Integration und kontinuierlicher Bereitstellung (CI/CD) schaffen, um zusätzliche Aktionen auszulösen.

⚠ Important

Die Nutzung des Ereignisses `CONFIGURATION_COMPLETE` beschleunigt zwar die Erstellung von Stapeln, aber Sie sollten sich der Nachteile bewusst sein, die damit verbunden sind. Erstens wird sie nur für eine Untergruppe von Ressourcentypen unterstützt, die die Drifterkennung unterstützen. Eine Liste der Ressourcentypen, die die Drifterkennung unterstützen, finden Sie unter [Unterstützung für Ressourcentypen](#). Dieser Ansatz ist möglicherweise nicht für alle Szenarien geeignet, insbesondere wenn Ressourcen gründliche Konsistenzprüfungen erfordern, um die volle Betriebsbereitschaft in der gesamten Cloud-Umgebung zu gewährleisten (z. B. in Produktionsumgebungen). Wir empfehlen Ihnen, Ihre Einsatzanforderungen und die Wichtigkeit der Konsistenzprüfungen für jede Ressource sorgfältig zu prüfen. Verwenden Sie das Ereignis `CONFIGURATION_COMPLETE`, um die Bereitstellungsgeschwindigkeit zu optimieren, ohne die Integrität und Zuverlässigkeit Ihrer Infrastruktur zu beeinträchtigen.

Da das Ereignis `CONFIGURATION_COMPLETE` nicht garantiert gesetzt wird, sollten alle Szenarien, die es verwenden, darauf vorbereitet sein, ein Ereignis `CREATE_COMPLETE` zu behandeln, wenn kein Ereignis `CONFIGURATION_COMPLETE` gesetzt wurde.



Wenn die Stack-Bereitstellung beginnt, starten sowohl die `AWS::ECR::Repository` - als auch die `AWS::ECS::Cluster` -Ressourcen den Erstellungsprozess (`ResourceStatus = CREATE_IN_PROGRESS`). Wenn der Ressourcentyp `AWS::ECR::Repository` mit der eventuellen Konsistenzprüfung begonnen hat (`DetailedStatus = CONFIGURATION_COMPLETE`), kann die Ressource `AWS::ECS::TaskDefinition` den Erstellungsprozess starten. Sobald die Ressource

AWS::ECS::TaskDefinition mit der Konsistenzprüfung beginnt, beginnt die Ressource AWS::ECS::Service mit dem Erstellungsprozess.

CREATE_IN_PROGRESS- und **CREATE_COMPLETE**-Ereignisse

- [Stack]: 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-ClusterCREATE_IN_PROGRESS, CONFIGURATION_COMPLETE
- [Ressource]: ECS-AufgabendefinitionCREATE_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
- [Stack]: CREATE_IN_PROGRESS, CONFIGURATION_COMPLETE
- [Ressource]: ECS-Aufgabendefinition CREATE_COMPLETE
- [Ressource]: ECS-Dienst CREATE_COMPLETE
- [Stack]: CREATE_COMPLETE

Überwachen Sie den Fortschritt einer Stack-Aktualisierung

Sie können den Fortschritt der Aktualisierung eines Stacks überwachen, indem Sie die Ereignisse des Stacks anzeigen. Die Registerkarte Ereignisse des Stapels zeigt jeden wichtigen Schritt bei der Erstellung und Aktualisierung des Stapels an, sortiert nach dem Zeitpunkt des jeweiligen Ereignisses, wobei die neuesten Ereignisse oben stehen. Weitere Informationen finden Sie unter [Fortschritt des Stapels 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 Beginn der Stack-Aktualisierung wird mit einem UPDATE_IN_PROGRESS-Ereignis für den Stack markiert:

```
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::DBInstance](#)-Ressource mit dem Namen MyDB würde zum Beispiel zu 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 Ihre Aktualisierungsvorlage beispielsweise eine Eigenschaftsänderung angibt, die von der Ressource nicht unterstützt wird, wie beispielsweise die Verringerung der Größe von `AllocatedStorage` für eine [AWS::RDS::DBInstance](#)-Ressource, würden Sie Ereignisse wie diese sehen:

```
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 Konfiguration 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`-Status bezeichnet. Möglicherweise haben Sie einen Stack, der mit dem Rollback auf eine alte 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 Stapel im Zustand `UPDATE_ROLLBACK_FAILED` kann nicht aktualisiert werden, aber er kann in einen funktionierenden Zustand (`UPDATE_ROLLBACK_COMPLETE`) zurückversetzt werden. Nachdem Sie den Stapel auf seine ursprünglichen Einstellungen zurückgesetzt haben, können Sie versuchen, ihn erneut 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 der an AWS-Managementkonsole und öffnen Sie die 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 aktualisieren möchten, wählen Sie Stapel-Aktionen und dann Update-Rollback fortsetzen.

Wenn keine der Lösungen [Behebung von Fehlern](#) funktioniert hat, können Sie die erweiterte Option verwenden, um die Ressourcen zu überspringen, CloudFormation für die ein Rollback nicht erfolgreich durchgeführt werden konnte. Sie müssen die logische [Reihenfolge IDs der Ressourcen, die Sie überspringen möchten, nachschlagen](#) und eingeben. Geben Sie nur die Ressourcen an, die während der `UpdateRollback`-Aktualisierung in den `UPDATE_FAILED`-Zustand übergegangen sind und nicht während der Vorwärtsaktualisierung.

 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, 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, den Sie weiter zurückrollen möchten.

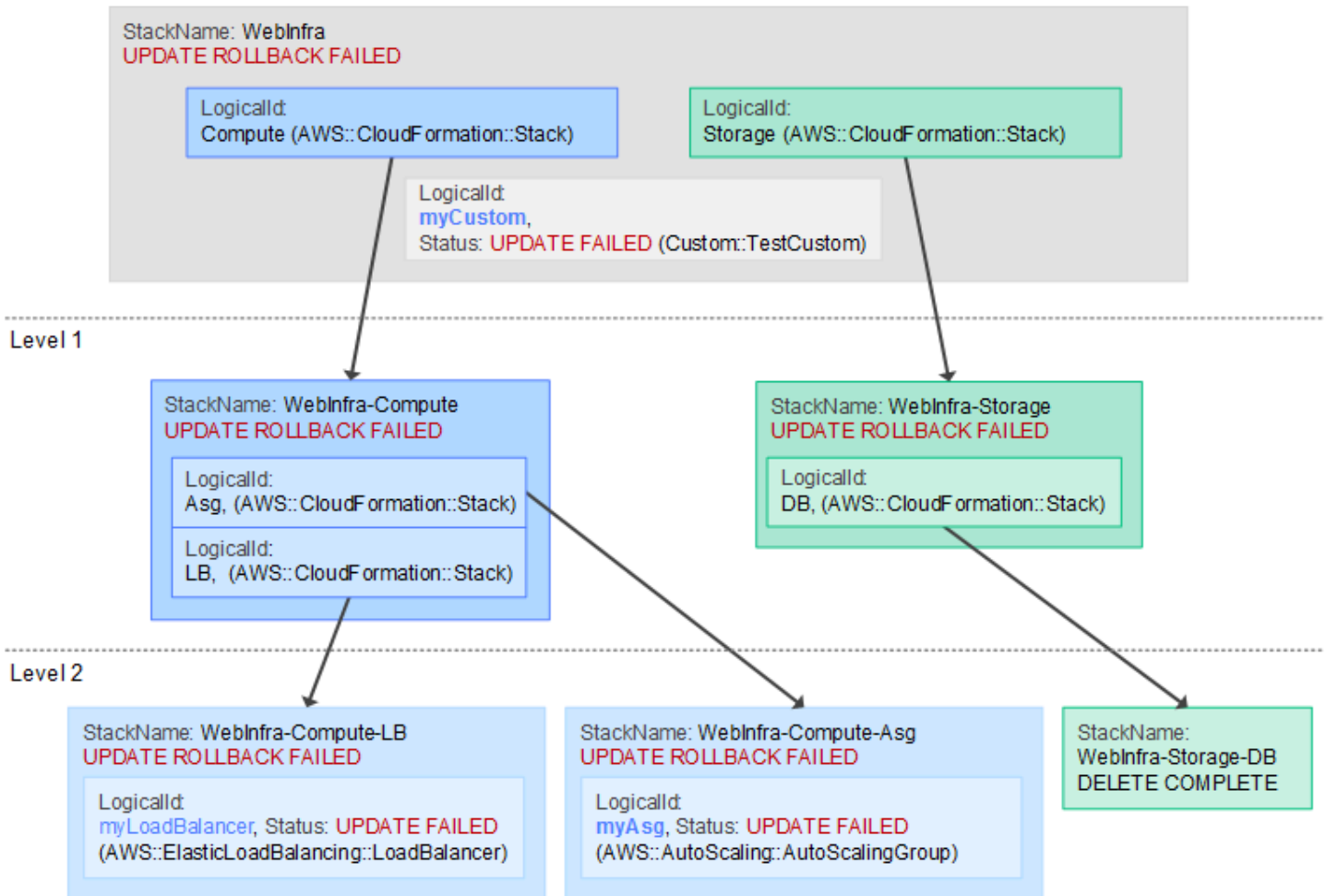
Weiteres Rollback von fehlgeschlagenen verschachtelten Stack-Updates

Wenn Sie mehrere ineinander verschachtelte Stapel haben, müssen Sie möglicherweise Ressourcen über mehrere verschachtelte Ebenen hinweg überspringen, um die gesamte Stapelhierarchie wieder in einen funktionierenden Zustand zu bringen.

Sie haben zum Beispiel einen StammStack namens WebInfra, der zwei kleinere Stack enthält: WebInfra-Compute und WebInfra-Storage. Diese beiden Stapel haben auch ihre eigenen verschachtelten Stapel innerhalb von ihnen.

Wenn während einer Aktualisierung etwas schief geht und der Aktualisierungsprozess fehlschlägt, kann die gesamte Stapelhierarchie in den Zustand `UPDATE_ROLLBACK_FAILED` geraten, wie im folgenden Diagramm dargestellt.

Root



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 mit dem Befehl `continue-update-rollback` in einen funktionsfähigen Zustand zu bringen, müssen Sie die Option `--resources-to-skip` verwenden, um Ressourcen zu überspringen, die beim Rollback fehlgeschlagen sind.

Das folgende `continue-update-rollback`-Beispiel setzt einen Rollback-Vorgang nach einer zuvor fehlgeschlagenen Stack-Aktualisierung fort. In diesem Beispiel umfasst die Option `--resources-to-skip` 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 Stapeln enthalten sind, müssen Sie jedoch sowohl den Namen des verschachtelten Stapels als auch seine logische ID angeben, getrennt durch einen Punkt. 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
```

So finden Sie den Stacknamen eines verschachtelten Stacks

Sie können ihn über die Stack-ID oder den Amazon Resource Name (ARN) des Child-Stacks finden.

Das folgende ARN-Beispiel bezieht sich auf einen Stapel mit dem Namen `WebInfra-Storage-Z2VKC706XKXT`.

```
arn:aws:cloudformation:us-east-1:123456789012:stack/WebInfra-Storage-Z2VKC706XKXT/
ea9e7f90-54f7-11e6-a032-028f3d2330bd
```

So finden Sie die logische ID eines verschachtelten Stapels

Die logische ID eines untergeordneten Stacks ist in der Vorlagendefinition seines übergeordneten Stacks enthalten. Im Diagramm ist das `LogicalId` des `WebInfra-Storage-DB` untergeordneten Stapels `DB` in seinem übergeordneten `WebInfra-Storage`.

In der CloudFormation Konsole finden Sie die logische ID auch in der Spalte Logische ID für die Stack-Ressource auf der Registerkarte Ressourcen oder der Registerkarte Ereignisse. Weitere Informationen finden Sie unter [Stack-Informationen von der CloudFormation Konsole aus anzeigen](#).

Ermitteln Sie die Ursache eines Stapelausfalls

Wenn Ihre Stack-Erstellung fehlschlägt, CloudFormation kann es Ihnen helfen, das Ereignis zu ermitteln, das wahrscheinlich die Hauptursache für den Stack-Ausfall ist. Je nach Szenario und Ihren Berechtigungen können AWS CloudTrail Ereignisse weitere Details zur Ursache liefern, falls der angegebene Statusgrund in Ereignisse nicht klar ist.

So ermitteln Sie die Ursache für einen Stack-Fehler

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie auf der Seite Stapel den ausgefallenen Stapel.
3. Wählen Sie die Registerkarte Ereignisse .
4. Wählen Sie Detect root cause aus. 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 Fehler Status mit der Kennzeichnung Wahrscheinliche Ursache aus, um mehr über die Ursache des Fehlers zu erfahren. Je nach Szenario und Ihren Berechtigungen können Sie möglicherweise ein detailliertes CloudTrail Ereignis überprüfen. Die folgenden Ergebnisse sind möglich, wenn Sie den Statuswählen
 - 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 Stapel unterstützen keine Ursachenermittlung.

Wählen Sie aus, wie mit Fehlern 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 Status UPDATE_FAILED oder CREATE_FAILED 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 Änderungssatz-Vorgänge verfügbar.

- Wenn Sie bei der Stackerstellung 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.
- Bei Aktualisierungs- und Change-Set-Vorgängen wird durch die Auswahl von Erfolgreich bereitgestellte Ressourcen beibehalten der Status erfolgreicher Ressourcen beibehalten, während fehlgeschlagene Ressourcen auf ihren 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.

Themen

- [Überblick über Optionen für Stack-Ausfälle](#)
- [Erforderliche Bedingungen für die Unterbrechung 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, 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, Ressourcen bereitzustellen, die zuvor nicht erfolgreich bereitgestellt werden konnten. Sie beheben Probleme, indem Sie Retry-, Update- oder Rollback-Vorgänge einsenden. Wenn Sie beispielsweise eine Amazon-EC2-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 Bereitstellungs Vorgang 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-) Berechtigung.
- **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 die Unterbrechung des Stack-Rollbacks

Um zu CloudFormation zu 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 beibehaltenwählen. Dies weist darauf hin, dass die erfolgreich erstellten Ressourcen CloudFormation nicht gelöscht werden sollen, auch wenn der gesamte Stack-Vorgang fehlschlägt.
2. Die Stack-Operation muss fehlgeschlagen sein, d. h. der Stack-Status ist entweder CREATE_FAILED oder 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 <https://console.aws.amazon.com/cloudformation> an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole.
2. Wählen Sie auf der Seite Stack oben rechts Stack erstellen und dann Mit neuen Ressourcen (Standard).
3. Für Voraussetzung - Vorlage vorbereitenwählen Sie Vorhandene Vorlage auswählen.

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 /cloudformation an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole. https://console.aws.amazon.com](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:
 - Vorhandene Vorlage verwenden

- 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-Managementkonsole und öffnen Sie die 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 Zustand des Stacks mit dem Befehl `describe-stacks`.

```
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

So bewahren Sie erfolgreich bereitgestellte Ressourcen während einer Stack-Update-Operation

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 Zustand des Stacks mit den Befehlen `describe-stacks` oder `describe-stack-events`.

```
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 Stapel mit dem Status `CREATE_FAILED` oder `UPDATE_FAILED` initiieren, aber nicht für einen Status `UPDATE_ROLLBACK_FAILED`.

So können Sie erfolgreich bereitgestellte Ressourcen während eines Änderungssatz-Vorgangs beibehalten

Geben Sie die `--disable-rollback` Option während eines [execute-change-set](#) Vorgangs 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 mit dem Befehl `describe-stacks` oder `describe-stack-events`.

```
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-abcdefghijkl",
      "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 myteststack \
  --use-previous-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"
}
```

5. Beschreiben Sie den Zustand des Stacks mit den Befehlen `describe-stacks` oder `describe-stack-events`.

```
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
  }
]
```

Einen Stapel zurückrollen

Sie können den Befehl [rollback-stack](#) verwenden, um einen Stack mit einem CREATE_FAILED oder UPDATE_FAILED -Status auf seinen letzten stabilen Zustand zurückzusetzen.

Der folgende Befehl rollback-stack rollt den angegebenen Stapel zurück.

```
aws cloudformation rollback-stack --stack-name myteststack
```

Der Befehl gibt die folgende Ausgabe zurück.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/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 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 werdenbreaching. 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-Auslöser 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.

Rollback-Auslöser bei der Erstellung oder Aktualisierung von Stapeln hinzufügen

So fügen Sie Rollback-Auslöser bei der Erstellung oder Aktualisierung von Stapeln hinzu (Konsole)

1. Wenn Sie einen Stapel erstellen oder aktualisieren, erweitern Sie auf der Seite Stapeloptionen konfigurieren unter Erweiterte Optionenden 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 Stapels.
5. Wenn Sie bereit sind, wählen Sie Senden um den Stapel zu erstellen oder zu aktualisieren.

So fügen Sie Rollback-Auslöser bei der Erstellung oder Aktualisierung von Stapeln hinzu (AWS CLI)

Verwenden Sie den Befehl [create-stack](#) oder [update-stack](#) mit der Option `--rollback-configuration`.

Der folgende update-stack Befehl wird beispielsweise *MyCompositeAlarm* als Rollback-Trigger mit einem Überwachungszeitraum von 5 Minuten festgelegt:

```
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
```

Rollback-Auslöser zu einem Änderungssatz hinzufügen

So fügen Sie Rollback-Auslöser zu einem Änderungssatz hinzu (Konsole)

1. Wenn Sie einen Änderungssatz erstellen, erweitern Sie auf der Seite Stack-Optionen konfigurieren unter Erweiterte Optionenden 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 fertig sind, wählen Sie Änderungsset erstellen um das Änderungsset zu erstellen.

So fügen Sie Rollback-Auslöser zu einem Änderungssatz (AWS CLI) hinzu

Verwenden Sie den Befehl [create-change-set](#) mit der Option `--rollback-configuration`.

Rollback-Auslöser für einen Stapel anzeigen

Informationen zum Anzeigen von Rollback-Auslösern für einen Stack finden Sie im Abschnitt Rollback-Konfiguration.

1. Wählen Sie auf der Seite Stapel den Stapel, den Sie anzeigen möchten, aus der Liste auf der linken Seite.

2. Erweitern Sie auf der Registerkarte Stack-Info den Abschnitt Rollback-Konfiguration um die Rollback-Auslöser anzuzeigen.

Rollback-Auslöser 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 Stapel den Stapel, den Sie anzeigen möchten, aus der Liste auf der linken Seite.
2. Wählen Sie die Registerkarte Änderungssätze und wählen Sie dann den Änderungssatz, den Sie anzeigen möchten.
3. Wählen Sie die Registerkarte Eingabe und sehen Sie sich den Abschnitt Rollback-Konfiguration an.

Erkennen Sie nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen mit Drift Detection

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 Amazon EC2 EC2-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 Aktualisierungs- oder Löschvorgänge des 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 unter [Unterstützung 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 zur Art der Ressourcenbereitstellung finden Sie unter dem Parameter ProvisioningType der Aktion [DescribeType](#) in der AWS CloudFormation API-Referenz und

dem Befehl [DescribeType](#) in der AWS CLI Befehlsreferenz. Weitere Informationen über private Ressourcen finden Sie unter [Erweiterungen 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 einer Drift auf einem Stack CloudFormation wird keine Drift auf verschachtelten Stacks erkannt, die zu diesem Stack gehören. Weitere Informationen finden Sie unter [Aufteilung einer Vorlage in wiederverwendbare Teile mit verschachtelten Stapeln](#). 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)	Description
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 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	Description
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.

Abweichungsstatus	Description
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	Description
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	Description
ADD	Ein Wert wurde zu einer Ressourceneigenschaft hinzugefügt, die ein Array- oder ein Listen-Datentyp ist.
REMOVE	Die Eigenschaft wurde aus der aktuellen Ressourcenkonfiguration 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
<code>AWS::SNS::Topic</code>	<code>AWS::SNS::Subscription</code>
<code>AWS::IAM::User</code>	<code>AWS::IAM::UserToGroupAddition</code>
<code>AWS::IAM::Group</code>	<code>AWS::IAM::ManagedPolicy</code>
<code>AWS::IAM::Role</code>	<code>AWS::IAM::Policy</code>
<code>AWS::IAM::User</code>	<code>AWS::IAM::RolePolicy</code>
<code>AWS::ElasticLoadBalancingV2::Listener</code>	<code>AWS::ElasticLoadBalancingV2::ListenerCertificate</code>
<code>AWS::EC2::SecurityGroup</code>	<code>AWS::EC2::SecurityGroupEgress</code> <code>AWS::EC2::SecurityGroupIngress</code>

- 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 [Code](#)Eigenschaftstyp der [AWS::Lambda::Function](#)Ressource 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 LoginProfile](#)Ergebnisse 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.

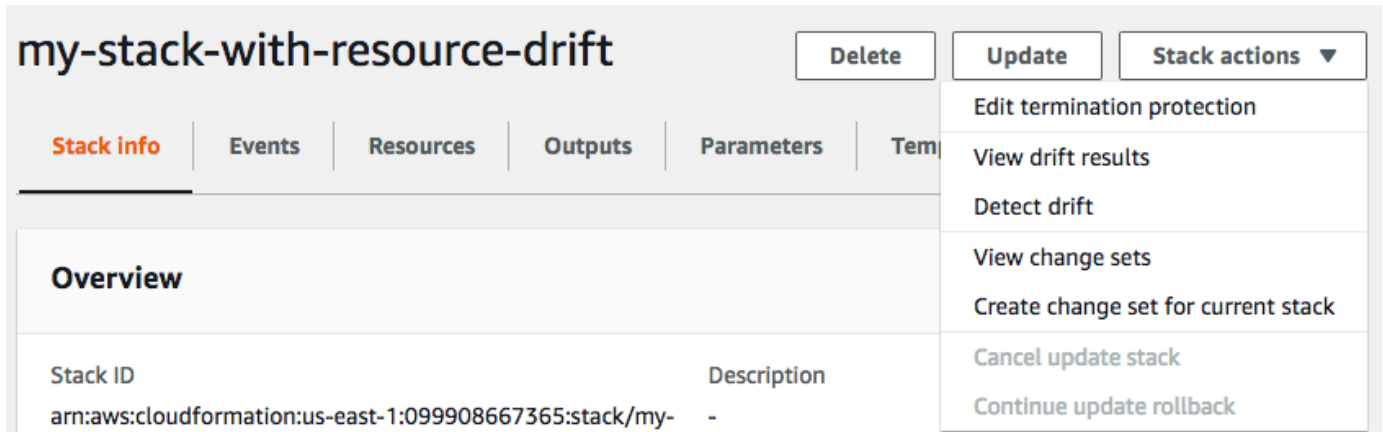
Ändern Sie die erwartete Konfiguration der Ressourceneigenschaft zu 1 024 MB und führen Sie dann die Abweichungserkennung aus, um diesen Fehler zu vermeiden.

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 den AWS-Managementkonsole

1. Öffnen Sie die 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.

CloudFormation > Stacks: my-stack-with-resource-drift > Drifts

Drifts

 **Detect stack drift**

Stack drift status

Drift detection enables you to detect whether a stack's actual configuration differs, or has drifted, from its template configuration. [Learn more.](#)

Drift status	Last drift check time
 DRIFTED	Fri, 24 May 2019 16:08:12 GMT

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.

Differences (4) < 1 >

☒	Property	Change	Expected value	Current value
☒	AttributeDefinitions.1	ADD	-	{"AttributeName":"stuff","AttributeType":"S"}
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	6
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	5	4
☒	Tags	REMOVE	[{"Key":"test","Value":"test"}]	-

Details

Expected

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}
```

Actual

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}
```

Um Drift auf einem gesamten Stack zu erkennen, verwenden Sie den AWS CLI

Important

Überprüfen Sie den Zeitpunkt der letzten Driftprüfung für den Stapel und stellen Sie sicher, dass er vor dem Zeitstempel liegt, der in den Ergebnissen der Ressourcendrift angezeigt wird, um die Verwendung veralteter Daten zu verhindern.

Verwenden Sie die folgenden AWS CLI Befehle AWS CLI, um Drift auf einem gesamten Stack mithilfe von zu erkennen:

- `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.

Das folgende Beispiel verwendet die Option `--stack-resource-drift-status-filters`, um Stack-Drift-Informationen für die Ressourcen abzufragen, 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-494PBHC076H4\\\"}\",
      "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-494PBHC076H4",
      "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-494PBHC076H4\\\"}",
    "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 Stapel hat den Driftstatus `IN_SYNC`. Sie haben eine Drift-Erkennung für eine oder mehrere Ressourcen in diesem Stack CloudFormation durchgeführt und dabei CloudFormation festgestellt, dass eine oder mehrere dieser Ressourcen driftet sind. CloudFormation aktualisiert den Status der Stack-Drift auf `DRIFTED`. Nehmen Sie umgekehrt an, Sie haben einen Stapel mit einem Driftstatus von `DRIFTED` aufgrund einer einzigen gedrifteten Ressource. Wenn Sie diese Ressource auf ihre erwarteten Eigenschaftswerte zurücksetzen und dann erneut eine Drift bei der Ressource feststellen, CloudFormation werden sowohl der Ressourcen-Drift-Status als auch der Stack-Drift-

Status auf aktualisiert, IN_SYNC ohne dass Sie die Drift erneut auf dem gesamten Stack feststellen müssen.

Um Abweichungen bei einer einzelnen Ressource zu erkennen, verwenden Sie AWS-Managementkonsole

1. Öffnen Sie die 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.

Differences (4) < 1 >

☒	Property	Change	Expected value	Current value
☒	AttributeDefinitions.1	ADD	-	{"AttributeName": "stuff", "AttributeType": "S"}
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	6
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	5	4
☒	Tags	REMOVE	[{"Key": "test", "Value": "test"}]	-

Details

Expected

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}
```

Actual

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}
```

Um Abweichungen bei einer einzelnen Ressource zu erkennen, verwenden Sie AWS CLI

• **⚠ Important**

Überprüfen Sie die Zeit der letzten Driftprüfung für die Stapelressource und stellen Sie sicher, dass sie vor dem Zeitstempel liegt, der in den Ergebnissen der Ressourcendrift angezeigt wird, um die Verwendung veralteter Daten zu verhindern.

Verwenden Sie den `detect-stack-resource-drift` Befehl AWS CLI, um Abweichungen bei einer einzelnen Ressource mithilfe von zu erkennen. Geben Sie die logische ID der Ressource sowie den Stack an, in dem sie enthalten ist.

Das folgende Beispiel führt eine Drifterkennungsoperation für eine bestimmte Stapelressource `my-drifted-resource` aus. 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-494PBHC076H4\"}\",
    "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-494PBHC076H4",
    "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-494PBHC076H4\"}\",
    "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 das Ressourcenimportfeature 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 unter [Manuelles Importieren von AWS Ressourcen in einen CloudFormation Stack](#). Eine Liste der Ressourcen, die den Import unterstützen, finden Sie unter [Unterstützung 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

```

In diesem Beispiel nehmen wir an, 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 Informationen über Drift Detect finden Sie unter [Erkennen Sie nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen mit Drift Detection](#).

Differences (3)					< 1 >
☒	Property	Change	Expected value	Current value	
☒	BillingMode	NOT_EQUAL	PROVISIONED	PAY_PER_REQUEST	
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	0	
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	1	0	
Details					
Expected			Actual		
<pre> { "AttributeDefinitions": [{ "AttributeName": "key", "AttributeType": "S" }], "BillingMode": "PROVISIONED", "KeySchema": [{ "AttributeName": "key", "KeyType": "HASH" }], "ProvisionedThroughput": { "ReadCapacityUnits": 5, "WriteCapacityUnits": 1 }, "TableName": "Games" } </pre>			<pre> { "AttributeDefinitions": [{ "AttributeName": "key", "AttributeType": "S" }], "BillingMode": "PAY_PER_REQUEST", "KeySchema": [{ "AttributeName": "key", "KeyType": "HASH" }], "ProvisionedThroughput": { "ReadCapacityUnits": 0, "WriteCapacityUnits": 0 }, "TableName": "Games" } </pre>		

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 Abweichungen bei einem Importvorgang mithilfe der 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-Managementkonsole und öffnen Sie die 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 [DeletionPolicy](#) für die Ressourcen an, die Sie aus Ihrem Stack entfernen möchten. Im folgenden Beispiel haben wir der GameTable Ressource [DeletionPolicy](#) das Attribut 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) `GameTable` 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
```

```

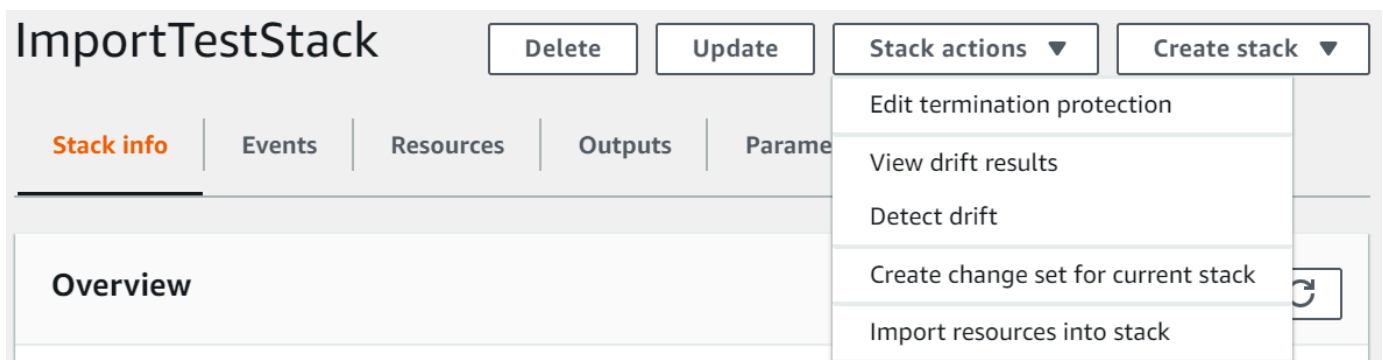
Description: Import test
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      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 [Identifikatoren für Ressourcen](#).
 - 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 `GameTable`-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 [laC-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 manuell in der Vorlage 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 dem Stack-Refactoring können Sie Ressourcen zwischen Stacks verschieben, monolithische Stacks in kleinere Komponenten aufteilen oder mehrere Stacks zu einem einzigen konsolidieren.

Die Ressourcenimportfunktion ermöglicht nicht nur die CloudFormation Verwaltung vorhandener Ressourcen, sondern kann auch in den folgenden Szenarien nützlich sein:

- Verschieben von Ressourcen zwischen Stackn- Sie können Ressourcen von einem Stack in einen anderen importieren und so Ihre Infrastruktur nach Bedarf umstrukturieren.
- Verschachtelung bestehender Stacks- Sie können einen bestehenden 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 ein [DeletionPolicy Attribut](#) haben.
- Identifikatoren für die Ressourcen, die Sie importieren, mit denen Sie die logischen Daten 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

- [Identifikatoren für Ressourcen](#)
- [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](#)

Identifikatoren für Ressourcen

Sie geben zwei Werte an, um jede Ressource, die Sie importieren, zu identifizieren.

- 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 Importprozess einleiten, in dem Sie die Bezeichner-Eigenschaften für die zu importierenden Ressourcen finden. Für einige Ressourcentypen gibt es möglicherweise mehrere Möglichkeiten, sie zu identifizieren, und Sie können in den Dropdown-Listen auswählen, welche Eigenschaft Sie verwenden möchten.

Alternativ können Sie die Identifier-Eigenschaften für die Ressourcen, die Sie importieren, abrufen, indem Sie den [get-template-summary](#) CLI-Befehl aufrufen und die S3-URL der Stack-Vorlage als Wert für die `--template-url` Option angeben.

- Ein Bezeichnerwert. Dies ist der tatsächliche Immobilienwert der Ressource. Beispielsweise könnte der tatsächliche Wert für die `BucketName`-Eigenschaft `MyS3Bucket` sein.

Sie können den Wert der Eigenschaft `identifier` in 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. Erforderliche Eigenschaften für jeden Ressourcentyp sind in der Referenz [AWS Ressource 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

Diese Tabelle beschreibt die verschiedenen Statustypen, die mit dem Feature Ressourcenimport verwendet werden.

Status des Importvorgangs	Description
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 [Richtlinien-Bedingungsschlüssel für CloudFormation](#).
- Die CloudFormation Stack-Limits gelten beim Import von Ressourcen. Weitere Informationen zu Grenzwerten finden Sie unter [CloudFormation Kontingente verstehen](#).

Weitere Ressourcen

Um die Stapeldrift mit einem Ressourcenimport zu beheben, siehe [Beheben 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 unter [Erzeugen 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 in dem neuen Stapel haben möchten. Speichern Sie die Vorlage lokal oder in einem Amazon S3-Bucket.
- Geben Sie für jede Ressource, die Sie importieren möchten, Folgendes an:

- die Eigenschaften und Eigenschaftswerte, die die aktuelle Konfiguration der Ressource definieren.
- den eindeutigen Bezeichner für die Ressource, z. B. den Ressourcennamen. Weitere Informationen finden Sie unter [Identifikatoren für Ressourcen](#).
- the [DeletionPolicy attribute](#).

Themen

- [-Beispielvorlage](#)
- [Erstellen Sie einen Stack aus vorhandenen Ressourcen mit dem AWS-Managementkonsole](#)
- [Erstellen Sie einen Stapel 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 `GameTable` sind die Ziele des Imports.

Note

Diese Vorlage ist nur als Beispiel gedacht. Um es 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"
          }
        ]
      }
    }
  }
}
```

```

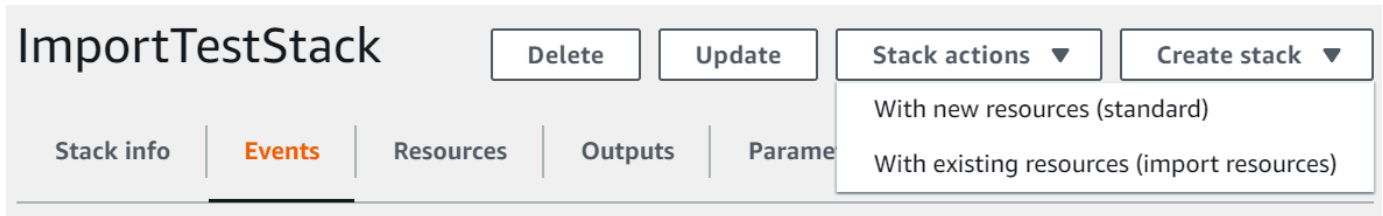
    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  },
  "GamesTable": {
    "Type": "AWS::DynamoDB::Table",
    "DeletionPolicy": "Retain",
    "Properties": {
      "TableName": "Games",
      "AttributeDefinitions": [
        {
          "AttributeName": "key",
          "AttributeType": "S"
        }
      ],
      "KeySchema": [
        {
          "AttributeName": "key",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
      }
    }
  }
}

```

Erstellen Sie einen Stack aus vorhandenen Ressourcen mit dem AWS-Managementkonsole

1. Melden Sie sich bei <https://console.aws.amazon.com/cloudformation> an AWS-Managementkonsole und öffnen Sie die 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 [Identifikatoren für Ressourcen](#).
 - 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.

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

Q Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

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 Drift finden Sie unter [Drift 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 Sie einen Stapel aus vorhandenen Ressourcen mithilfe der AWS CLI

1. Um zu erfahren, welche Eigenschaften die einzelnen Ressourcentypen in der Vorlage identifizieren, führen Sie den Befehl `get-template-summary` aus und geben dabei die S3-URL der Vorlage an. Beispielsweise kann die `AWS::DynamoDB::Table`-Ressource mit der `TableName`-Eigenschaft identifiziert werden. Für die `GameTable`-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 [Identifikatoren für Ressourcen](#).

2. Stellen Sie eine Liste der tatsächlichen Ressourcen aus Ihrer Vorlage und ihrer eindeutigen Bezeichner im folgenden JSON-String-Format zusammen.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":  
{"TableName":"Games"}},  
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifier":  
{"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 Befehl `create-change-set` und ersetzen den Platzhaltertext. Geben Sie für die Option `--change-set-type` einen Wert von **IMPORT** an. Ersetzen Sie bei der Option `--resources-to-import` die Beispiel-JSON-Zeichenfolge durch die tatsächliche JSON-Zeichenfolge, die Sie gerade erstellt haben.

```
aws cloudformation create-change-set \  
  --stack-name TargetStack --change-set-name ImportChangeSet \  
  --resources-to-import '[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}, {"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifier":{"TableName":"Service"}}]'
```



```
--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 kein Inline-YAML. Die Anforderungen für das Escapen von Anführungszeichen in der JSON-Zeichenfolge variieren je nach Ihrem Terminal. Weitere Informationen finden Sie unter [Anführungszeichen in Zeichenketten verwenden](#) im AWS Command Line Interface -Benutzerhandbuch.

Alternativ können Sie auch eine Datei-URL als Eingabe für die Option --resources-to-import verwenden, wie im folgenden Beispiel gezeigt.

```
--resources-to-import file:///ResourcesToImport.txt
```

- Überprüfen Sie den Änderungssatz, um sicherzustellen, dass die richtigen Ressourcen importiert werden.

```
aws cloudformation describe-change-set \  
--change-set-name ImportChangeSet --stack-name TargetStack
```

- Um den Änderungssatz zu initiieren und die Ressourcen zu importieren, verwenden Sie den folgenden execute-change-set-Befehl und ersetzen den Platzhaltertext. Nach erfolgreichem Abschluss des Vorgangs (IMPORT_COMPLETE) werden die Ressourcen erfolgreich importiert.

```
aws cloudformation execute-change-set \  
--change-set-name ImportChangeSet --stack-name TargetStack
```

- (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 Drift finden Sie unter [Erkennen von Abweichungen bei einzelnen Stack-Ressourcen](#).

- a. Führt die Drifterkennung für den angegebenen Stapel aus.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Wenn dieser Befehl erfolgreich ist, erhalten Sie die folgende Beispielausgabe.

```
{ "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. Zeigen Sie Drift-Informationen für die Ressourcen an, die im angegebenen Stapel auf Drift geprüft 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 unter [Erzeugen 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-Bucket.

So erhalten Sie eine Kopie der Vorlage eines laufenden Stacks

1. Öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation/>.
 2. Wählen Sie aus der Liste der Stapel den Stapel aus, aus dem Sie die Vorlage abrufen möchten.
 3. Wählen Sie in der Detailansicht des Stapels die Registerkarte Vorlage und wählen Sie dann In die Zwischenablage kopieren.
 4. Fügen Sie den Code in einen Texteditor ein und fügen Sie weitere Ressourcen zur Vorlage hinzu.
- Geben Sie für jede Ressource, die Sie importieren möchten, Folgendes an:
 - die Eigenschaften und Eigenschaftswerte, die die aktuelle Konfiguration der Ressource definieren.
 - den eindeutigen Bezeichner für die Ressource, z. B. den Ressourcennamen. Weitere Informationen finden Sie unter [Identifikatoren für Ressourcen](#).
 - the [DeletionPolicy attribute](#).

Themen

- [-Beispielvorlage](#)
- [Importieren Sie eine vorhandene Ressource in einen Stack mit dem AWS-Managementkonsole](#)
- [Importieren Sie eine vorhandene Ressource in einen Stack mithilfe der AWS CLI](#)

-Beispielvorlage

In dieser Schritt-für-Schritt-Anleitung gehen wir davon aus, dass Sie die folgende Beispielvorlage mit dem Namen `TemplateToImport.json` verwenden, die zwei DynamoDB-Tabellen angibt. `ServiceTable` ist derzeit Teil des Stacks und `GamesTable` ist die Tabelle, die Sie importieren möchten.

Note

Diese Vorlage ist nur als Beispiel gedacht. Um es 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",
```

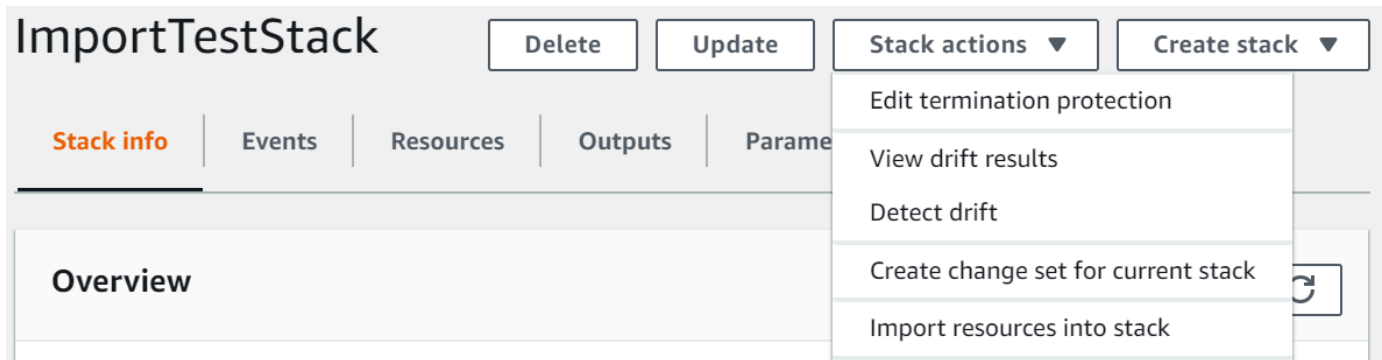
```
"Properties": {
  "TableName": "Games",
  "AttributeDefinitions": [
    {
      "AttributeName": "key",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "key",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 1
  }
}
}
```

Importieren Sie eine vorhandene Ressource in einen Stack mit dem AWS-Managementkonsole

 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-Managementkonsole und öffnen Sie die 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 [Identifikatoren für Ressourcen](#).
 - 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 `GameTable`-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 [Stapeloptionen konfigurieren](#).

Die Seite Ereignisse für den Stack wird angezeigt.

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

Q Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

- (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 Drift finden Sie unter [Drift auf einem ganzen CloudFormation Stack erkennen](#).
- (Optional) Wenn die importierten Ressourcen nicht mit den erwarteten Vorlagenkonfigurationen übereinstimmen, korrigieren Sie entweder die Vorlagenkonfigurationen oder aktualisieren Sie die Ressourcen direkt. Weitere Informationen über den Import getriebener Ressourcen finden Sie unter [Beheben von Abweichungen mit einer Importoperation](#).

Importieren Sie eine vorhandene Ressource in einen Stack mithilfe der AWS CLI

- Um zu erfahren, welche Eigenschaften die einzelnen Ressourcentypen in der Vorlage identifizieren, führen Sie den Befehl `get-template-summary` aus und geben dabei die S3-URL der Vorlage an. Beispielsweise kann die `AWS::DynamoDB::Table`-Ressource mit der `TableName`-Eigenschaft identifiziert werden. Für die `GameTable`-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 [Identifikatoren für Ressourcen](#).

2. Stellen Sie eine Liste der zu importierenden Ressourcen und ihrer eindeutigen Bezeichner im folgenden JSON-String-Format zusammen.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}]
```

Alternativ können Sie die JSON-formatierten Parameter in einer Konfigurationsdatei angeben.

Um beispielsweise zu importieren `GamesTable`, könnten 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 Befehl `create-change-set` und ersetzen den Platzhaltertext. Geben Sie für die Option `--change-set-type` einen Wert von **IMPORT** an. Ersetzen Sie bei der Option `--resources-to-import` die Beispiel-JSON-Zeichenfolge 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","ResourceIdentifier":{"TableName":"Games"}}]'
```

Note

`--resources-to-import` unterstützt kein Inline-YAML. Die Anforderungen für das Escapen von Anführungszeichen in der JSON-Zeichenfolge variieren je nach Ihrem

Terminal. Weitere Informationen finden Sie unter [Anführungszeichen in Zeichenketten verwenden](#) im AWS Command Line Interface -Benutzerhandbuch.

Alternativ können Sie auch eine Datei-URL als Eingabe für die Option `--resources-to-import` 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. Um den Änderungssatz zu initiieren und die Ressourcen zu importieren, verwenden Sie den folgenden `execute-change-set`-Befehl und ersetzen den Platzhaltertext. Alle Tags auf Stack-Ebene werden zu diesem Zeitpunkt auf importierte Ressourcen angewendet. Weitere Informationen finden Sie unter [Stapeloptionen konfigurieren](#). 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 Drift finden Sie unter [Drift auf einem ganzen CloudFormation Stack erkennen](#).

- a. Führt die Drifterkennung für den angegebenen Stapel aus.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Wenn dieser Befehl erfolgreich ist, erhalten Sie die folgende Beispielausgabe.

```
{ "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. Zeigen Sie Drift-Informationen für die Ressourcen an, die im angegebenen Stapel auf Drift geprüft 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 über den Import getriebener Ressourcen finden Sie unter [Beheben von Abweichungen mit einer Importoperation](#).

Verschieben von Ressourcen zwischen Stacks

Mithilfe des `resource import`-Features 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 Thema [AWS Ressourcen in einen CloudFormation Stack importieren](#) durchzulesen.

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.

Refaktorisieren Sie einen Stack mit dem AWS-Managementkonsole

1. Geben Sie in der Quellvorlage a Retain [DeletionPolicy](#) für 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 `GameTable`.

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,
```

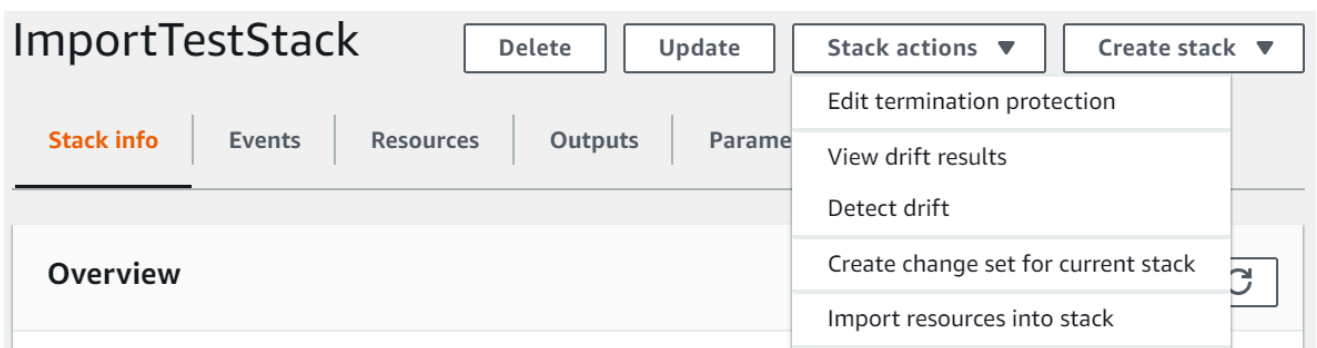
```

    "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 GameTable 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 GameTable). Weitere Informationen finden Sie unter [Identifikatoren für Ressourcen](#).
 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, *GameTables*.

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.

ImportTestStack Delete Update Stack actions ▼ Create stack ▼

Stack info **Events** Resources Outputs Parameters Template Change sets

Events 🔄 ⚙️

🔍 Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	✅ IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	✅ UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	🔄 UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	✅ IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	🔄 IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	🔄 IMPORT_IN_PROGRESS	User initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	🔄 REVIEW_IN_PROGRESS	User initiated

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.

Refaktorisieren Sie einen Stack mit dem AWS CLI

1. Geben Sie in der Quellvorlage a Retain [DeletionPolicy](#) für 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 `GameTable`.

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 zu importierenden Ressourcen und ihrer eindeutigen Bezeichner im folgenden JSON-String-Format zusammen. Weitere Informationen finden Sie unter [Identifikatoren für Ressourcen](#).

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"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"
    }
  }
]
```

- Um einen Änderungssatz zu erstellen, verwenden Sie den folgenden Befehl `create-change-set` und ersetzen den Platzhaltertext. Geben Sie für die Option `--change-set-type` einen Wert von **IMPORT** an. Ersetzen Sie bei der Option `--resources-to-import` die Beispiel-JSON-Zeichenfolge 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", "ResourceIdentifier": {"TableName": "Games"}}]'"
```

Note

`--resources-to-import` unterstützt kein Inline-YAML. Die Anforderungen für das Escapen von Anführungszeichen in der JSON-Zeichenfolge variieren je nach Ihrem Terminal. Weitere Informationen finden Sie unter [Anführungszeichen in Zeichenketten verwenden](#) im AWS Command Line Interface -Benutzerhandbuch.

Alternativ können Sie auch eine Datei-URL als Eingabe für die Option `--resources-to-import` verwenden, wie im folgenden Beispiel gezeigt.

```
--resources-to-import file://ResourcesToImport.txt
```

- 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
```

- Um den Änderungssatz zu initiieren und die Ressource zu importieren, verwenden Sie den folgenden `execute-change-set`-Befehl und ersetzen den Platzhaltertext. 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 das `resource import`-Feature, 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 [Aufteilung einer Vorlage in wiederverwendbare Teile mit verschachtelten Stapeln](#).

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 Thema [Manuelles Importieren von AWS Ressourcen in einen CloudFormation Stack](#) durchzulesen.

Validierung des Imports von verschachtelten Stacks

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 Sie einen vorhandenen Stack mit dem AWS-Managementkonsole

1. Fügen Sie die `AWS::CloudFormation::Stack` Ressource mit einem zur übergeordneten Stack-Vorlage hinzu Retain [DeletionPolicy](#). Im folgenden Beispiel ist die übergeordnete Stapelvorlage `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

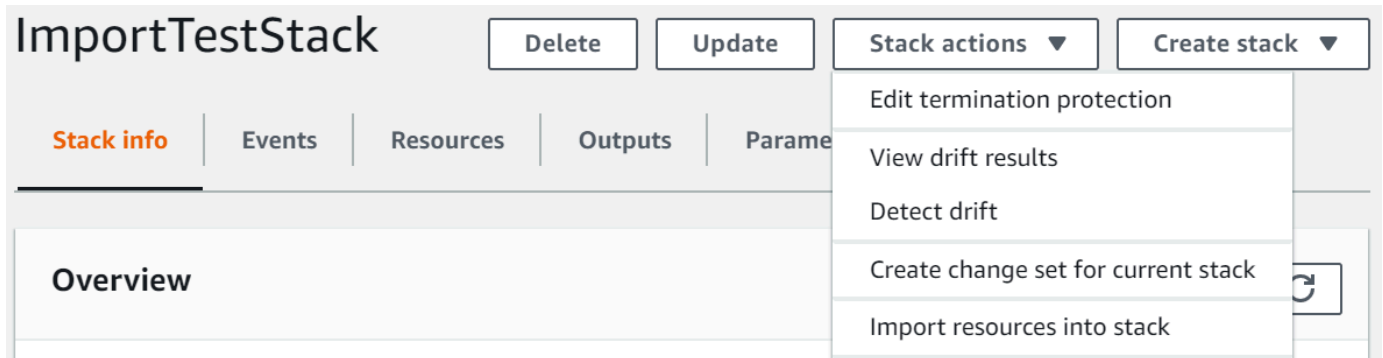
```

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. Öffnen Sie die 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 Identifikatorwert den ARN des Stapels ein, den Sie importieren. Beispiel, *arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10*.

Resources to import (1)

The following logical IDs are new to your template. Provide the identifier values for the resource you are importing.

NestedStack

AWS::CloudFormation::Stack

Identifier property

StackId

Identifier value

Enter StackId

c. Wählen Sie Weiter aus.

- 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.

- Vergewissern Sie sich auf der *MyParentStack* 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.
- Der Bereich Ereignisse der Seite Stack-Details für Ihren übergeordneten Stack wird angezeigt.

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

 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 wurde.

Verschachteln Sie einen vorhandenen Stapel 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. Stellen Sie eine JSON-Zeichenkette wie im folgenden Beispiel zusammen, 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 es durch den ARN des Stacks, den Sie importieren möchten.

```
[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack","ResourceIdentifier":{"StackId":"arn:aws:cloudformation:us-east-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10"}}]
```

Alternativ können Sie die Parameter auch in einer Konfigurationsdatei angeben.

Um beispielsweise zu importieren *MyNestedStack*, könnten 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
```

- Um einen Änderungssatz zu erstellen, verwenden Sie den folgenden Befehl `create-change-set` und ersetzen den Platzhaltertext. Geben Sie für die Option `--change-set-type` einen Wert von **IMPORT** an. Ersetzen Sie bei der Option `--resources-to-import` die Beispiel-JSON-Zeichenfolge 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 kein Inline-YAML. Die Anforderungen für das Escapen von Anführungszeichen in der JSON-Zeichenfolge variieren je nach Ihrem Terminal. Weitere Informationen finden Sie unter [Anführungszeichen in Zeichenketten verwenden](#) im AWS Command Line Interface -Benutzerhandbuch.

Alternativ können Sie auch eine Datei-URL als Eingabe für die Option `--resources-to-import` verwenden, wie im folgenden Beispiel gezeigt.

```
--resources-to-import file://ResourcesToImport.txt
```

Wenn dieser Befehl erfolgreich ist, erhalten Sie die folgende Beispielausgabe.

```
{  
  "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"  
}
```

- Überprüfen Sie den Änderungssatz, um sicherzustellen, dass der richtige Stack importiert wird.

```
aws cloudformation describe-change-set --change-set-name ImportChangeSet
```

5. Um den Änderungssatz zu initiieren und den Stack in den übergeordneten QuellStack zu importieren, verwenden Sie den folgenden `execute-change-set`-Befehl und ersetzen den Platzhaltertext. 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 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. Bei verschachtelten Stapeln erstellen Sie den Änderungssatz vom Stammstapel aus.

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 Voraussetzungen erfüllen:

- Die Ressource muss einen statischen benutzerdefinierten Namen haben, der in Ihrer Vorlage definiert ist. Dynamische Namen (mit `!Ref` oder anderen Funktionen) werden derzeit nicht unterstützt.

- Die Ressource muss einen DeletionPolicy von Retain oder RetainExceptOnCreate haben.
- 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 Schreibschutz werden nicht unterstützt. Um herauszufinden, wie die primäre ID für einen Typ lautet, suchen Sie im Ressourcenschema nach der Eigenschaft `primaryIdentifier`. Weitere Informationen zu dieser Eigenschaft finden Sie unter [primaryIdentifier](#).

Example Beispiel Auto-Import

Das folgende Beispiel verwendet einen Änderungssatz `CreateChangeSet`, um einen Stack namens `my-stack` auf der Grundlage einer Vorlagendatei `template.yaml` zu erstellen, und importiert automatisch die passenden Ressourcen.

```
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
- Überprüfen Sie, ob 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.

Machen Sie einen Importvorgang rückgängig mit dem AWS-Managementkonsole

1. Geben Sie a Retain [DeletionPolicy](#) für 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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#). 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.

Machen Sie einen Importvorgang rückgängig, indem Sie AWS CLI

1. Geben Sie a Retain [DeletionPolicy](#) für 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 (GameTable) und die zugehörigen Parameter und Ausgaben aus dem Stack zu löschen.

```
aws cloudformation update-stack --stack-name MyStack
```

Stapel-Refactoring

Mit Stack-Refactoring können Sie Ressourcen in Ihren CloudFormation Stacks reorganisieren und gleichzeitig die vorhandenen Ressourceneigenschaften und -daten beibehalten. Sie können Ressourcen zwischen Stapeln verschieben, große Stapel in kleinere aufteilen oder mehrere Stapel zu einem kombinieren.

Themen

- [So funktioniert das Stack-Refactoring](#)
- [Überlegungen zum Stack-Refactoring](#)
- [Voraussetzungen](#)
- [Stacks umgestalten \(Konsole\)](#)
- [Stacks umgestalten \(AWS CLI\)](#)
- [Begrenzte Ressourcen](#)

So funktioniert das Stack-Refactoring

Das Refactoring von Stacks umfasst diese Phasen:

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 Ihren Refactor- Definieren Sie, wie die Ressourcen organisiert werden sollen. Berücksichtigen Sie Ihre Abhängigkeiten, Namenskonventionen und operativen Grenzen. Diese können sich auf die spätere Validierung auswirken. CloudFormation
3. Bestimmen Sie die ZielStack- Entscheiden Sie, in welche Stack Sie die Ressourcen umstrukturieren wollen. Sie können Ressourcen zwischen mindestens 2 Stacks (mithilfe der Konsole) und maximal 5 Stacks (mithilfe der AWS CLI) verschieben. Ressourcen können zwischen verschachtelten Stapeln verschoben werden.
4. Aktualisieren Sie Ihre Vorlagen — Passen Sie Ihre CloudFormation Vorlagen an die geplante Änderung an, z. B. das Verschieben von Ressourcendefinitionen zwischen Vorlagen. IDs Während dieses Vorgangs können Sie Logical umbenennen.
5. Erstellen Sie den Stack-Refactor- Geben Sie eine Liste von Stack-Namen und-Vorlagen an, die Sie refactorieren möchten.
6. Überprüfen Sie die Auswirkungen des Refactorings und lösen Sie etwaige Konflikte — CloudFormation validiert die von Ihnen bereitgestellten Vorlagen und prüft stapelübergreifende Abhängigkeiten, Ressourcentypen mit Tag-Aktualisierungsproblemen und Konflikte bei logischen IDs von Ressourcen.

Wenn die Validierung erfolgreich ist, CloudFormation wird eine Vorschau der Refactor-Aktionen generiert, die während der Ausführung ausgeführt werden.

Wenn die Validierung fehlschlägt, beheben Sie die festgestellten Probleme und versuchen Sie es erneut. Bei Konflikten stellen Sie eine logische ID-Zuordnung der Ressource bereit, die die Quelle und das Ziel der konfligierenden Ressourcen angibt.

7. Führen Sie den Refactor aus- Nachdem Sie sich vergewissert haben, dass die Änderungen mit Ihren Refactoring-Zielen übereinstimmen, schließen Sie den Stack-Refactor ab.
8. Monitor- Verfolgen Sie den Ausführungsstatus, um sicherzustellen, dass der Vorgang erfolgreich abgeschlossen wird.

Überlegungen zum Stack-Refactoring

Wenn Sie Ihre Stacks umstrukturieren, sollten Sie Folgendes beachten:

- Das Stack-Refactoring beschränkt sich auf die Reorganisation vorhandener Ressourcen. Während des Refactorings können Sie keine Ressourcen erstellen oder löschen, Ressourcenkonfigurationen ändern oder neue Parameter, Bedingungen oder Zuordnungen ändern oder hinzufügen. Um diese Änderungen vorzunehmen, aktualisieren Sie zuerst Ihre Stacks und führen Sie dann den Stack-Refactor durch.
- Sie können die gleiche Ressource nicht in mehrere Stapel umwandeln.
- Sie können keine Ressourcen refaktorisieren, die sich auf Pseudoparameter beziehen, deren Werte sich zwischen Quell- und ZielStack unterscheiden, wie beispielsweise `AWS::StackName`.
- CloudFormation unterstützt keine leeren Stacks. Wenn das Refactoring dazu führen würde, dass ein Stack keine Ressourcen mehr hat, müssen Sie diesem Stack zunächst mindestens eine Ressource hinzufügen, bevor Sie ihn ausführen. [create-stack-refactor](#) Dies kann eine einfache Ressource wie `AWS::SNS::Topic` oder `AWS::CloudFormation::WaitCondition` sein.
Beispiel:

```
Resources:
  MySimpleSNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: MySimpleTopic
```

- Stack Refactor unterstützt keine Stacks, die mit Stack-Richtlinien verknüpft sind, unabhängig davon, was die Richtlinien erlauben oder verbieten.

Voraussetzungen

Um Stacks umzugestalten, müssen Sie die überarbeiteten Vorlagen bereits erstellt haben.

Verwenden Sie den Befehl [get-template](#), um die CloudFormation Vorlagen für die Stacks abzurufen, die Sie umgestalten möchten.

```
aws cloudformation get-template --stack-name Stack1
```


Wenn Sie die Vorlagen haben, verwenden Sie die integrierte Entwicklungsumgebung (IDE) Ihrer Wahl, um sie zu aktualisieren und die gewünschte Struktur und Ressourcenorganisation zu verwenden.

Stacks umgestalten (Konsole)

Gehen Sie wie folgt vor, um Stacks mithilfe der Konsole umzugestalten.

Um Stacks umzugestalten

1. [Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com /cloudformation.](https://console.aws.amazon.com/cloudformation)
2. Wählen Sie in der Navigationsleiste oben auf dem Bildschirm aus, AWS-Region wo sich Ihre Stacks befinden.
3. Wählen Sie im Navigationsbereich auf der linken Seite Stack Refactors aus.
4. Wählen Sie auf der Seite Stack-Refactors die Option Stack-Refactor starten aus.
5. Geben Sie unter Beschreibung eine Beschreibung ein, die Ihnen bei der Identifizierung Ihres Stack-Refaktors hilft. Wählen Sie anschließend Weiter.
6. Gehen Sie für Stack 1 wie folgt vor:
 - a. Wählen Sie entweder Vorlage für einen vorhandenen Stack aktualisieren oder Neuen Stack erstellen aus.

Wenn Sie Vorlage für einen vorhandenen Stack aktualisieren ausgewählt haben, wählen Sie einen vorhandenen Stack aus der Liste aus. Oder wählen Sie Enter a stack ARN, um den ARN eines vorhandenen Stacks einzugeben.

Wenn Sie Neuen Stack erstellen ausgewählt haben, geben Sie für Stackname einen Namen für den neuen Stack ein.
 - b. Wählen Sie unter Bestehende Vorlage durch überarbeitete Vorlage ersetzen die Option Amazon S3 S3-URL oder Eine Vorlagendatei hochladen aus, um die gewünschte Vorlage für Stack 1 hochzuladen.
 - c. Wählen Sie Weiter aus.
7. Gehen Sie für Stack 2 wie folgt vor:
 - a. Wählen Sie entweder Vorlage für einen vorhandenen Stack aktualisieren oder Neuen Stack erstellen aus.

Wenn Sie Vorlage für einen vorhandenen Stack aktualisieren ausgewählt haben, wählen Sie einen vorhandenen Stack aus der Liste aus. Oder wählen Sie Enter a stack ARN, um den ARN eines vorhandenen Stacks einzugeben.

Wenn Sie Neuen Stack erstellen ausgewählt haben, geben Sie für Stackname einen Namen für den neuen Stack ein.

- b. Wählen Sie unter Bestehende Vorlage durch überarbeitete Vorlage ersetzen die Option Amazon S3 S3-URL oder Vorlagendatei hochladen aus, um die gewünschte Vorlage für Stack 2 hochzuladen.
 - c. Wählen Sie Weiter aus.
8. Vergewissern Sie sich auf der Seite „Logische Ressourcen-ID-Umbenennungen angeben“, dass Sie CloudFormation wissen, wie Stacks umgestaltet werden, indem Sie alle angezeigten Ressourcen der richtigen logischen Zuordnung zuordnen. IDs Wenn die Logik IDs einer Ressource geändert wurde, müssen Sie im Rahmen des Stack-Refactorings angeben, wie sie umbenannt wurde, indem Sie den Quell-Stack-Namen, die ursprüngliche logische ID, den Ziel-Stack-Namen und die umbenannte logische ID angeben. In einigen Fällen erkennt die CloudFormation Konsole die Ressourcenzuordnung möglicherweise automatisch, und Sie können einfach überprüfen, ob die vorab ausgefüllte Ressourcenzuordnung korrekt ist, bevor Sie fortfahren.
 9. Wählen Sie Weiter aus.
 10. Überprüfen Sie auf der Seite Überprüfen und ausführen alle Ihre Auswahlen aus den vorherigen Schritten und stellen Sie sicher, dass alles korrekt eingerichtet ist.
 11. Wenn Sie bereit sind, die Stacks umzugestalten, wählen Sie Stack-Refactor ausführen.

Stacks umgestalten ()AWS CLI

Zu den AWS CLI Befehlen für das Stack-Refactoring gehören:

- [create-stack-refactor](#) um geplante Änderungen zu validieren und eine Vorschau zu generieren.
- [describe-stack-refactor](#) um den Status und die Details eines Stack-Refactoring-Vorgangs abzurufen.
- [execute-stack-refactor](#) um den validierten Stack-Refactoring-Vorgang abzuschließen.
- [list-stack-refactors](#) um alle Stack-Refactoring-Operationen in Ihrem Konto mit ihrem aktuellen Status und grundlegenden Informationen aufzulisten.

- [list-stack-refactor-actions](#) um eine Vorschau der spezifischen Aktionen anzuzeigen, die während der Refactor-Ausführung auf jedem Stack und jeder Ressource ausgeführt CloudFormation werden.

Verwenden Sie die folgende Prozedur, um einen Stapel unter Verwendung des AWS CLI zu refaktorisieren.

Um Stacks umzustrukturieren

1. Verwenden Sie den [create-stack-refactor](#) Befehl und geben Sie die Stack-Namen und die aktualisierten Vorlagen für die Stacks an, die umgestaltet werden sollen. Fügen Sie die `--enable-stack-creation` Option hinzu, mit der neue Stapel erstellt werden können CloudFormation, falls diese noch nicht existieren.

```
aws cloudformation create-stack-refactor \  
  --stack-definitions \  
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \  
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \  
  --enable-stack-creation
```

Der Befehl gibt ein `StackRefactorId` zurück, das Sie in späteren Schritten verwenden werden.

```
{  
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841"  
}
```

Wenn bei der Vorlagenvalidierung Konflikte festgestellt werden (was Sie im nächsten Schritt bestätigen können), verwenden Sie den [create-stack-refactor](#) Befehl mit der `--resource-mappings` Option.

```
aws cloudformation create-stack-refactor \  
  --stack-definitions \  
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \  
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \  
  --enable-stack-creation \  
  --resource-mappings file://resource-mapping.json
```

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

```
[
  {
    "Source": {
      "StackName": "Stack1",
      "LogicalResourceId": "MySNSTopic"
    },
    "Destination": {
      "StackName": "Stack2",
      "LogicalResourceId": "MyLambdaSNSTopic"
    }
  }
]
```

2. Verwenden Sie den [describe-stack-refactor](#) Befehl, um sicherzustellen, dass dies der Fall ist. Der Status ist `CREATE_COMPLETE`. Damit wird überprüft, ob die Validierung vollständig ist.

```
aws cloudformation describe-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

Beispielausgabe:

```
{
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",
  "StackIds": [
    "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
    "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
  ],
  "ExecutionStatus": "AVAILABLE",
  "Status": "CREATE_COMPLETE"
}
```

3. Verwenden Sie den [list-stack-refactor-actions](#) Befehl, um eine Vorschau der spezifischen Aktionen anzuzeigen, die ausgeführt werden.

```
aws cloudformation list-stack-refactor-actions \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

Beispielausgabe:

```

{
  "StackRefactorActions": [
    {
      "Action": "MOVE",
      "Entity": "RESOURCE",
      "PhysicalResourceId": "MyTestLambdaRole",
      "Description": "No configuration changes detected.",
      "Detection": "AUTO",
      "TagResources": [],
      "UntagResources": [],
      "ResourceMapping": {
        "Source": {
          "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
          "LogicalResourceId": "MyLambdaRole"
        },
        "Destination": {
          "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
          "LogicalResourceId": "MyLambdaRole"
        }
      }
    },
    {
      "Action": "MOVE",
      "Entity": "RESOURCE",
      "PhysicalResourceId": "MyTestFunction",
      "Description": "Resource configuration changes will be validated during
refactor execution.",
      "Detection": "AUTO",
      "TagResources": [
        {
          "Key": "aws:cloudformation:stack-name",
          "Value": "Stack2"
        },
        {
          "Key": "aws:cloudformation:logical-id",
          "Value": "MyFunction"
        },
        {
          "Key": "aws:cloudformation:stack-id",
          "Value": "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
        }
      ]
    }
  ]
}

```

```

    }
  ],
  "UntagResources": [
    "aws:cloudformation:stack-name",
    "aws:cloudformation:logical-id",
    "aws:cloudformation:stack-id"
  ],
  "ResourceMapping": {
    "Source": {
      "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
      "LogicalResourceId": "MyFunction"
    },
    "Destination": {
      "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
      "LogicalResourceId": "MyFunction"
    }
  }
}
]
}

```

4. Nachdem Sie Ihre Änderungen überprüft und bestätigt haben, verwenden Sie den [execute-stack-refactor](#) Befehl, um den Stack-Refactoring-Vorgang abzuschließen.

```

aws cloudformation execute-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841

```

5. Verwenden Sie den [describe-stack-refactor](#) Befehl, um den Ausführungsstatus zu überwachen.

```

aws cloudformation describe-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841

```

Beispielausgabe:

```

{
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",
  "StackIds": [
    "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",

```

```

    "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
  ],
  "ExecutionStatus": "SUCCEEDED",
  "Status": "COMPLETE"
}

```

Begrenzte Ressourcen

- Stack Refactoring unterstützt nur Ressourcentypen mit einem provisioningType von FULLY_MUTABLE was Sie mit dem Befehl [describe-type](#) überprüfen können.
- CloudFormation überprüft bei der Erstellung des Refactorings die Eignung von Ressourcen und meldet alle nicht unterstützten Ressourcen in der Ausgabe des Befehls. [describe-stack-refactor](#)
- Die folgenden Ressourcen werden für das Stack-Refactoring nicht unterstützt:
 - 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::CertificateManager::Certificate
 - AWS::CloudFormation::CustomResource
 - AWS::CloudFormation::Macro

- `AWS::CloudFormation::WaitCondition`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CodeDeploy::DeploymentGroup`
- `AWS::CodePipeline::CustomActionType`
- `AWS::Cognito::UserPoolRiskConfigurationAttachment`
- `AWS::Cognito::UserPoolUICustomizationAttachment`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Config::ConfigRule`
- `AWS::Config::ConfigurationRecorder`
- `AWS::Config::DeliveryChannel`
- `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::DirectoryService::MicrosoftAD`
- `AWS::DynamoDB::GlobalTable`
- `AWS::EC2::CustomerGateway`
- `AWS::EC2::EIP`
- `AWS::EC2::LaunchTemplate`
- `AWS::EC2::NetworkInterfacePermission`
- `AWS::EC2::PlacementGroup`
- `AWS::EC2::SpotFleet`

- AWS::EC2::VPCDHCPOptionsAssociation
- AWS::EC2::VolumeAttachment
- AWS::EMR::Cluster
- AWS::EMR::InstanceFleetConfig
- AWS::EMR::InstanceGroupConfig
- AWS::ElastiCache::CacheCluster
- AWS::ElastiCache::ReplicationGroup
- AWS::ElastiCache::SecurityGroup
- AWS::ElastiCache::SecurityGroupIngress
- AWS::ElasticBeanstalk::ConfigurationTemplate
- AWS::ElasticLoadBalancing::LoadBalancer
- AWS::ElasticLoadBalancingV2::ListenerCertificate
- AWS::Elasticsearch::Domain
- AWS::FIS::ExperimentTemplate
- AWS::Glue::Schema
- AWS::GuardDuty::IPSet
- AWS::GuardDuty::PublishingDestination
- AWS::GuardDuty::ThreatIntelSet
- AWS::IAM::AccessKey
- AWS::IAM::UserToGroupAddition
- AWS::ImageBuilder::Component
- AWS::IoT::PolicyPrincipalAttachment
- AWS::IoT::ThingPrincipalAttachment
- AWS::IoTFleetWise::Campaign
- AWS::IoTWireless::WirelessDeviceImportTask
- AWS::Lambda::EventInvokeConfig
- AWS::Lex::BotVersion
- AWS::M2::Application
- AWS::MSK::Configuration
- AWS::MSK::ServerlessCluster

- `AWS::Maester::DocumentType`
- `AWS::MediaTailor::Channel`
- `AWS::NeptuneGraph::PrivateGraphEndpoint`
- `AWS::Omics::AnnotationStore`
- `AWS::Omics::ReferenceStore`
- `AWS::Omics::SequenceStore`
- `AWS::OpenSearchServerless::Collection`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`
- `AWS::PCAConnectorAD::Connector`
- `AWS::PCAConnectorAD::DirectoryRegistration`
- `AWS::PCAConnectorAD::Template`
- `AWS::PCAConnectorAD::TemplateGroupAccessControlEntry`
- `AWS::Panorama::PackageVersion`
- `AWS::QuickSight::Theme`
- `AWS::RDS::DBSecurityGroup`
- `AWS::RDS::DBSecurityGroupIngress`
- `AWS::Redshift::ClusterSecurityGroup`
- `AWS::Redshift::ClusterSecurityGroupIngress`
- `AWS::RefactorSpaces::Environment`
- `AWS::RefactorSpaces::Route`
- `AWS::RefactorSpaces::Service`
- `AWS::RoboMaker::RobotApplication`
- `AWS::RoboMaker::SimulationApplication`
- `AWS::Route53::RecordSet`

- `AWS::Route53::RecordSetGroup`
- `AWS::SDB::Domain`
- `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::WAF::ByteMatchSet`
- `AWS::WAF::IPSet`
- `AWS::WAF::Rule`
- `AWS::WAF::SizeConstraintSet`
- `AWS::WAF::SqlInjectionMatchSet`
- `AWS::WAF::WebACL`
- `AWS::WAF::XssMatchSet`
- `AWS::WAFv2::IPSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::WAFv2::RuleGroup`
- `AWS::WAFv2::WebACL`
- `AWS::WorkSpaces::Workspace`

Unterstützung für Ressourcentypen

In der folgenden Tabelle sind AWS Ressourcentypen aufgeführt, die derzeit Import-, Drifterkennungs- und IaC-Generatoroperationen (Infrastructure as Code) unterstützen. Jeder Ressourcentypname verweist auf das entsprechende Referenzthema im [CloudFormation Template Reference Guide](#).

Eine Ressource, die den Ressourcenimport unterstützt, könnte auch den Auto-Import unterstützen. Weitere Informationen finden Sie unter [AWS Ressourcen in einen CloudFormation Stack importieren](#).

Dies ist keine erschöpfende 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 unter [Ressourcentypen, die Cloud Control API unterstützen](#) im Cloud-Control-API-Benutzerhandbuch. 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 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 für Sie freigegeben wurden](#).

Beachten Sie, dass der IaC-Generator nur AWS Ressourcen unterstützt, die mit der Cloud Control API in Ihrer Region kompatibel sind.

Für den programmgesteuerten Zugriff auf Informationen über öffentliche und private bereitstellbare Ressourcentypen können Sie die Funktion AWS -Cloud-Control- API verwenden. Weitere Informationen finden Sie unter [Feststellen, ob ein Ressourcentyp Cloud-Control-API unterstützt](#) im Cloud-Control-API-Benutzerhandbuch.

Um mit dem Import, der Drifterkennung oder dem IaC-Generator zu beginnen, finden Sie hier einige nützliche Themen zum Nachlesen:

- [AWS Ressourcen in einen CloudFormation Stack importieren](#)
- [Erkennen Sie nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen mit Drift Detection](#)
- [Erzeugen Sie Vorlagen aus vorhandenen Ressourcen mit dem IaC-Generator](#)

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
<u>AWS::ACMPCA::Certificate</u>	 Ja	 Nein	 Nein
<u>AWS::ACMPCA::CertificateAuthority</u>	 Ja	 Ja	 Ja
<u>AWS::ACMPCA::CertificateAuthorityActivation</u>	 Ja	 Nein	 Nein
<u>AWS::ACMPCA::Permission</u>	 Ja	 Ja	 Nein
<u>AWS::AIOps::InvestigationGroup</u>	 Ja	 Ja	 Nein
<u>AWS::APS::AnomalyDetector</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::APS::ResourcePolicy</u>	 Ja	 Ja	 Nein
<u>AWS::APS::RuleGroupsNamespace</u>	 Ja	 Ja	 Ja
<u>AWS::APS::Scraper</u>	 Ja	 Ja	 Nein
<u>AWS::APS::Workspace</u>	 Ja	 Ja	 Ja
<u>AWS::ARCRegionSwitch::Plan</u>	 Ja	 Ja	 Nein
<u>AWS::ARCZonalShift::AutoshiftObserverNotificationStatus</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ARCZonalShift::ZonalAutoshiftConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::AccessAnalyzer::Analyzer</u>	 Ja	 Ja	 Ja
<u>AWS::AmazonMQ::Broker</u>	 Ja	 Ja	 Nein
<u>AWS::AmazonMQ::Configuration</u>	 Ja	 Ja	 Nein
<u>AWS::Amplify::App</u>	 Ja	 Ja	 Ja
<u>AWS::Amplify::Branch</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Amplify::Domain</u>	 Ja	 Ja	 Ja
<u>AWS::AmplifyUIBuilder::Component</u>	 Ja	 Ja	 Ja
<u>AWS::AmplifyUIBuilder::Form</u>	 Ja	 Ja	 Ja
<u>AWS::AmplifyUIBuilder::Theme</u>	 Ja	 Ja	 Ja
<u>AWS::ApiGateway::Account</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::ApiKey</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::ApiGateway::Authorizer</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::BasePathMapping</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::BasePathMappingV2</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::ClientCertificate</u>	 Ja	 Ja	 Ja
<u>AWS::ApiGateway::Deployment</u>	 Ja	 Ja	 Ja
<u>AWS::ApiGateway::DocumentationPart</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ApiGateway::DocumentationVersion</u>	 Ja	 Ja	 Ja
<u>AWS::ApiGateway::DomainName</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::DomainNameAccessAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::DomainNameV2</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::GatewayResponse</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::Method</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ApiGateway::Model</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::RequestValidator</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::Resource</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGateway::RestApi</u>	 Ja	 Ja	 Ja
<u>AWS::ApiGateway::Stage</u>	 Ja	 Nein	 Ja
<u>AWS::ApiGateway::UsagePlan</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::ApiGateway::UsagePlanKey</u>	 Ja	 Nein	 Ja
<u>AWS::ApiGateway::VpcLink</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGatewayV2::Api</u>	 Ja	 Ja	 Ja
<u>AWS::ApiGatewayV2::ApiMapping</u>	 Ja	 Ja	 Ja
<u>AWS::ApiGatewayV2::Authorizer</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGatewayV2::Deployment</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ApiGatewayV2::DomainName</u>	 Ja	 Ja	 Ja
<u>AWS::ApiGatewayV2::Integration</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGatewayV2::IntegrationResponse</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGatewayV2::Model</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGatewayV2::Route</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGatewayV2::RouteResponse</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ApiGatewayV2::RoutingRule</u>	 Ja	 Ja	 Nein
<u>AWS::ApiGatewayV2::VpcLink</u>	 Ja	 Ja	 Ja
<u>AWS::AppConfig::Application</u>	 Ja	 Ja	 Ja
<u>AWS::AppConfig::ConfigurationProfile</u>	 Ja	 Ja	 Nein
<u>AWS::AppConfig::Deployment</u>	 Ja	 Ja	 Nein
<u>AWS::AppConfig::DeploymentStrategy</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::AppConfig::Environment</u>	 Ja	 Ja	 Nein
<u>AWS::AppConfig::Extension</u>	 Ja	 Ja	 Nein
<u>AWS::AppConfig::ExtensionAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::AppConfig::HostedConfigurationVersion</u>	 Ja	 Ja	 Nein
<u>AWS::AppFlow::Connector</u>	 Ja	 Ja	 Ja
<u>AWS::AppFlow::ConnectorProfile</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::AppFlow::Flow</u>	 Ja	 Ja	 Ja
<u>AWS::AppIntegrations::Application</u>	 Ja	 Ja	 Nein
<u>AWS::AppIntegrations::DataIntegration</u>	 Ja	 Ja	 Ja
<u>AWS::AppIntegrations::EventIntegration</u>	 Ja	 Ja	 Ja
<u>AWS::AppRunner::AutoScalingConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::AppRunner::ObservabilityConfiguration</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::AppRunner::Service</u>	 Ja	 Ja	 Nein
<u>AWS::AppRunner::VpcConnector</u>	 Ja	 Ja	 Nein
<u>AWS::AppRunner::VpcIngressConnection</u>	 Ja	 Ja	 Nein
<u>AWS::AppStream::AppBlock</u>	 Ja	 Ja	 Nein
<u>AWS::AppStream::AppBlockBuilder</u>	 Ja	 Ja	 Nein
<u>AWS::AppStream::Application</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::AppStream::ApplicationEntitlementAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::AppStream::ApplicationFleetAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::AppStream::DirectoryConfig</u>	 Ja	 Ja	 Ja
<u>AWS::AppStream::Entitlement</u>	 Ja	 Ja	 Nein
<u>AWS::AppStream::ImageBuilder</u>	 Ja	 Ja	 Ja
<u>AWS::AppSync::Api</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::AppSync::ChannelNamespace</u>	 Ja	 Ja	 Nein
<u>AWS::AppSync::DataSource</u>	 Ja	 Ja	 Nein
<u>AWS::AppSync::DomainName</u>	 Ja	 Ja	 Ja
<u>AWS::AppSync::DomainNameApiAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::AppSync::FunctionConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::AppSync::GraphQLApi</u>	 Nein	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::AppSync::Resolver</u>	 Ja	 Ja	 Nein
<u>AWS::AppSync::SourceApiAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::AppTest::TestCase</u>	 Ja	 Ja	 Nein
<u>AWS::ApplicationAutoScaling::ScalableTarget</u>	 Ja	 Ja	 Ja
<u>AWS::ApplicationAutoScaling::ScalingPolicy</u>	 Ja	 Nein	 Nein
<u>AWS::ApplicationInsights::Application</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ApplicationSignals::Discovery</u>	 Ja	 Ja	 Nein
<u>AWS::ApplicationSignals::GroupingConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::ApplicationSignals::ServiceLevelObjective</u>	 Ja	 Ja	 Nein
<u>AWS::Athena::CapacityReservation</u>	 Ja	 Ja	 Ja
<u>AWS::Athena::DataCatalog</u>	 Ja	 Ja	 Ja
<u>AWS::Athena::NamedQuery</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Athena::PreparedStatement</u>	 Ja	 Ja	 Ja
<u>AWS::Athena::WorkGroup</u>	 Ja	 Ja	 Ja
<u>AWS::AuditManager::Assessment</u>	 Ja	 Nein	 Nein
<u>AWS::AutoScaling::AutoScalingGroup</u>	 Ja	 Nein	 Ja
<u>AWS::AutoScaling::LaunchConfiguratio n</u>	 Ja	 Ja	 Ja
<u>AWS::AutoScaling::LifecycleHook</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::AutoScaling::ScalingPolicy</u>	 Ja	 Ja	 Ja
<u>AWS::AutoScaling::ScheduledAction</u>	 Ja	 Ja	 Ja
<u>AWS::AutoScaling::WarmPool</u>	 Ja	 Ja	 Nein
<u>AWS::B2BI::Capability</u>	 Ja	 Ja	 Nein
<u>AWS::B2BI::Partnership</u>	 Ja	 Ja	 Nein
<u>AWS::B2BI::Profile</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::B2BI::Transformer</u>	 Ja	 Ja	 Nein
<u>AWS::BCMDDataExports::Export</u>	 Ja	 Ja	 Nein
<u>AWS::Backup::BackupPlan</u>	 Ja	 Ja	 Nein
<u>AWS::Backup::BackupSelection</u>	 Ja	 Nein	 Ja
<u>AWS::Backup::BackupVault</u>	 Ja	 Ja	 Nein
<u>AWS::Backup::Framework</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Backup::LogicallyAirGappedBackupVault</u>	 Ja	 Ja	 Nein
<u>AWS::Backup::ReportPlan</u>	 Ja	 Ja	 Ja
<u>AWS::Backup::RestoreTestingPlan</u>	 Ja	 Ja	 Nein
<u>AWS::Backup::RestoreTestingSelection</u>	 Ja	 Ja	 Nein
<u>AWS::BackupGateway::Hypervisor</u>	 Ja	 Ja	 Nein
<u>AWS::Batch::ComputeEnvironment</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Batch::ConsumableResource</u>	 Ja	 Ja	 Nein
<u>AWS::Batch::JobDefinition</u>	 Ja	 Ja	 Nein
<u>AWS::Batch::JobQueue</u>	 Ja	 Ja	 Ja
<u>AWS::Batch::SchedulingPolicy</u>	 Ja	 Ja	 Ja
<u>AWS::Batch::ServiceEnvironment</u>	 Ja	 Ja	 Nein
<u>AWS::Bedrock::Agent</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Bedrock::AgentAlias</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::ApplicationInferenceProfile</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::AutomatedReasoningPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::Bedrock::AutomatedReasoningPolicyVersion</u>	 Ja	 Ja	 Nein
<u>AWS::Bedrock::Blueprint</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::DataAutomationProject</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Bedrock::DataSource</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::Flow</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::FlowAlias</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::FlowVersion</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::Guardrail</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::GuardrailVersion</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Bedrock::IntelligentPromptRouter</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::KnowledgeBase</u>	 Ja	 Nein	 Ja
<u>AWS::Bedrock::Prompt</u>	 Ja	 Ja	 Ja
<u>AWS::Bedrock::PromptVersion</u>	 Ja	 Ja	 Ja
<u>AWS::BedrockAgentCore::BrowserCustom</u>	 Ja	 Ja	 Nein
<u>AWS::BedrockAgentCore::CodeInterpreterCustom</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::BedrockAgentCore::Gateway</u>	 Ja	 Ja	 Nein
<u>AWS::BedrockAgentCore::GatewayTarget</u>	 Ja	 Ja	 Nein
<u>AWS::BedrockAgentCore::Memory</u>	 Ja	 Ja	 Nein
<u>AWS::BedrockAgentCore::Runtime</u>	 Ja	 Ja	 Nein
<u>AWS::BedrockAgentCore::RuntimeEndpoint</u>	 Ja	 Ja	 Nein
<u>AWS::BedrockAgentCore::WorkloadIdentity</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Billing::BillingView</u>	 Ja	 Ja	 Nein
<u>AWS::BillingConductor::BillingGroup</u>	 Ja	 Ja	 Ja
<u>AWS::BillingConductor::CustomLineItem</u>	 Ja	 Ja	 Ja
<u>AWS::BillingConductor::PricingPlan</u>	 Ja	 Ja	 Ja
<u>AWS::BillingConductor::PricingRule</u>	 Ja	 Ja	 Ja
<u>AWS::Budgets::BudgetsAction</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::CE::AnomalyMonitor</u>	 Ja	 Ja	 Nein
<u>AWS::CE::AnomalySubscription</u>	 Ja	 Ja	 Nein
<u>AWS::CE::CostCategory</u>	 Ja	 Ja	 Ja
<u>AWS::CUR::ReportDefinition</u>	 Ja	 Ja	 Nein
<u>AWS::Cases::CaseRule</u>	 Ja	 Ja	 Nein
<u>AWS::Cases::Domain</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Cases::Field</u>	 Ja	 Ja	 Nein
<u>AWS::Cases::Layout</u>	 Ja	 Ja	 Nein
<u>AWS::Cases::Template</u>	 Ja	 Ja	 Nein
<u>AWS::Cassandra::Keyspace</u>	 Ja	 Ja	 Nein
<u>AWS::Cassandra::Table</u>	 Ja	 Ja	 Ja
<u>AWS::Cassandra::Type</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::CertificateManager::Account</u>	 Ja	 Ja	 Nein
<u>AWS::Chatbot::CustomAction</u>	 Ja	 Ja	 Nein
<u>AWS::Chatbot::MicrosoftTeamsChannelC onfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::Chatbot::SlackChannelConfigurat ion</u>	 Ja	 Ja	 Ja
<u>AWS::CleanRooms::AnalysisTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::CleanRooms::Collaboration</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::CleanRooms::ConfiguredTable</u>	 Ja	 Ja	 Ja
<u>AWS::CleanRooms::ConfiguredTableAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::CleanRooms::IdMappingTable</u>	 Ja	 Ja	 Nein
<u>AWS::CleanRooms::IdNamespaceAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::CleanRooms::Membership</u>	 Ja	 Ja	 Ja
<u>AWS::CleanRooms::PrivacyBudgetTemplate</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::CleanRoomsML::TrainingDataset</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFormation::GuardHook</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFormation::HookDefaultVersion</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFormation::HookTypeConfig</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFormation::HookVersion</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFormation::LambdaHook</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::CloudFormation::ModuleDefaultVersion</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFormation::ModuleVersion</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFormation::PublicTypeVersion</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFormation::Publisher</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFormation::ResourceDefaultVersion</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFormation::ResourceVersion</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::CloudFormation::Stack</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFormation::StackSet</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFormation::TypeActivation</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFront::AnycastIpList</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFront::CachePolicy</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFront::CloudFrontOriginAccessIdentity</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::CloudFront::ConnectionFunction</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFront::ConnectionGroup</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFront::ContinuousDeploymen tPolicy</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFront::Distribution</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFront::DistributionTenant</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFront::Function</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::CloudFront::KeyGroup</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFront::KeyValueStore</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFront::MonitoringSubscript ion</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFront::OriginAccessControl</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFront::OriginRequestPolicy</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFront::PublicKey</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::CloudFront::RealtimeLogConfig</u>	 Ja	 Ja	 Ja
<u>AWS::CloudFront::ResponseHeadersPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFront::TrustStore</u>	 Ja	 Ja	 Nein
<u>AWS::CloudFront::VpcOrigin</u>	 Ja	 Ja	 Nein
<u>AWS::CloudTrail::Channel</u>	 Ja	 Ja	 Nein
<u>AWS::CloudTrail::Dashboard</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::CloudTrail::EventDataStore</u>	 Ja	 Ja	 Ja
<u>AWS::CloudTrail::ResourcePolicy</u>	 Ja	 Ja	 Nein
<u>AWS::CloudTrail::Trail</u>	 Ja	 Ja	 Ja
<u>AWS::CloudWatch::Alarm</u>	 Ja	 Ja	 Ja
<u>AWS::CloudWatch::CompositeAlarm</u>	 Ja	 Ja	 Ja
<u>AWS::CloudWatch::Dashboard</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::CloudWatch::MetricStream</u>	 Ja	 Ja	 Nein
<u>AWS::CodeArtifact::Domain</u>	 Ja	 Ja	 Ja
<u>AWS::CodeArtifact::PackageGroup</u>	 Ja	 Ja	 Nein
<u>AWS::CodeArtifact::Repository</u>	 Ja	 Ja	 Ja
<u>AWS::CodeBuild::Fleet</u>	 Ja	 Ja	 Nein
<u>AWS::CodeConnections::Connection</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::CodeDeploy::Application</u>	 Ja	 Ja	 Ja
<u>AWS::CodeDeploy::DeploymentConfig</u>	 Ja	 Ja	 Nein
<u>AWS::CodeGuruProfiler::ProfilingGroup</u>	 Ja	 Ja	 Ja
<u>AWS::CodeGuruReviewer::RepositoryAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::CodePipeline::CustomActionType</u>	 Ja	 Ja	 Nein
<u>AWS::CodePipeline::Pipeline</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::CodePipeline::Webhook</u>	 Ja	 Ja	 Nein
<u>AWS::CodeStarConnections::Connection</u>	 Ja	 Ja	 Ja
<u>AWS::CodeStarConnections::Repository Link</u>	 Ja	 Ja	 Nein
<u>AWS::CodeStarConnections::SyncConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::CodeStarNotifications::NotificationRule</u>	 Ja	 Ja	 Ja
<u>AWS::Cognito::IdentityPool</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Cognito::IdentityPoolPrincipalTag</u>	 Ja	 Ja	 Ja
<u>AWS::Cognito::IdentityPoolRoleAttachment</u>	 Ja	 Ja	 Nein
<u>AWS::Cognito::LogDeliveryConfiguration</u>	 Ja	 Nein	 Nein
<u>AWS::Cognito::ManagedLoginBranding</u>	 Ja	 Ja	 Nein
<u>AWS::Cognito::Terms</u>	 Ja	 Ja	 Nein
<u>AWS::Cognito::UserPool</u>	 Ja	 Nein	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Cognito::UserPoolClient</u>	 Ja	 Ja	 Nein
<u>AWS::Cognito::UserPoolDomain</u>	 Ja	 Ja	 Nein
<u>AWS::Cognito::UserPoolGroup</u>	 Ja	 Ja	 Ja
<u>AWS::Cognito::UserPoolIdentityProvider</u>	 Ja	 Ja	 Nein
<u>AWS::Cognito::UserPoolResourceServer</u>	 Ja	 Ja	 Nein
<u>AWS::Cognito::UserPoolRiskConfigurationAttachment</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Cognito::UserPoolUICustomizationAttachment</u>	 Ja	 Ja	 Nein
<u>AWS::Cognito::UserPoolUser</u>	 Ja	 Nein	 Nein
<u>AWS::Cognito::UserPoolUserToGroupAttachment</u>	 Ja	 Ja	 Nein
<u>AWS::Comprehend::DocumentClassifier</u>	 Ja	 Ja	 Ja
<u>AWS::Comprehend::Flywheel</u>	 Ja	 Ja	 Ja
<u>AWS::Config::AggregationAuthorization</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Config::ConfigRule</u>	 Ja	 Ja	 Ja
<u>AWS::Config::ConfigurationAggregator</u>	 Ja	 Ja	 Ja
<u>AWS::Config::ConformancePack</u>	 Ja	 Ja	 Ja
<u>AWS::Config::OrganizationConformancePack</u>	 Ja	 Ja	 Ja
<u>AWS::Config::StoredQuery</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::AgentStatus</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Connect::ApprovedOrigin</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::ContactFlow</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::ContactFlowModule</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::ContactFlowVersion</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::DataTable</u>	 Ja	 Ja	 Nein
<u>AWS::Connect::DataTableAttribute</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Connect::DataTableRecord</u>	 Ja	 Ja	 Nein
<u>AWS::Connect::EmailAddress</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::EvaluationForm</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::HoursOfOperation</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::Instance</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::InstanceStorageConfig</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Connect::IntegrationAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::PhoneNumber</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::PredefinedAttribute</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::Prompt</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::Queue</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::QuickConnect</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Connect::RoutingProfile</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::Rule</u>	 Ja	 Ja	 Nein
<u>AWS::Connect::SecurityKey</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::SecurityProfile</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::TaskTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::TrafficDistributionGroup</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Connect::User</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::UserHierarchyGroup</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::UserHierarchyStructure</u>	 Ja	 Ja	 Nein
<u>AWS::Connect::View</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::ViewVersion</u>	 Ja	 Ja	 Ja
<u>AWS::Connect::Workspace</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ConnectCampaigns::Campaign</u>	 Ja	 Ja	 Ja
<u>AWS::ConnectCampaignsV2::Campaign</u>	 Ja	 Ja	 Ja
<u>AWS::ControlTower::EnabledBaseline</u>	 Ja	 Ja	 Nein
<u>AWS::ControlTower::EnabledControl</u>	 Ja	 Ja	 Ja
<u>AWS::ControlTower::LandingZone</u>	 Ja	 Ja	 Nein
<u>AWS::CustomerProfiles::CalculatedAttributeDefinition</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::CustomerProfiles::Domain</u>	 Ja	 Ja	 Ja
<u>AWS::CustomerProfiles::EventStream</u>	 Ja	 Ja	 Ja
<u>AWS::CustomerProfiles::EventTrigger</u>	 Ja	 Ja	 Nein
<u>AWS::CustomerProfiles::Integration</u>	 Ja	 Ja	 Ja
<u>AWS::CustomerProfiles::ObjectType</u>	 Ja	 Ja	 Ja
<u>AWS::CustomerProfiles::SegmentDefinition</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::DMS::DataMigration</u>	 Ja	 Ja	 Nein
<u>AWS::DMS::DataProvider</u>	 Ja	 Ja	 Nein
<u>AWS::DMS::InstanceProfile</u>	 Ja	 Ja	 Nein
<u>AWS::DMS::MigrationProject</u>	 Ja	 Ja	 Nein
<u>AWS::DMS::ReplicationConfig</u>	 Ja	 Ja	 Ja
<u>AWS::DSQL::Cluster</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::DataBrew::Dataset</u>	 Ja	 Ja	 Ja
<u>AWS::DataBrew::Job</u>	 Ja	 Ja	 Ja
<u>AWS::DataBrew::Project</u>	 Ja	 Ja	 Ja
<u>AWS::DataBrew::Recipe</u>	 Ja	 Ja	 Ja
<u>AWS::DataBrew::Ruleset</u>	 Ja	 Ja	 Ja
<u>AWS::DataBrew::Schedule</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::DataPipeline::Pipeline</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::Agent</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::LocationAzureBlob</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::LocationEFS</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::LocationFSxLustre</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::LocationFSxONTAP</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::DataSync::LocationFSxOpenZFS</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::LocationFSxWindows</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::LocationHDFS</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::LocationNFS</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::LocationObjectStorage</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::LocationS3</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::DataSync::LocationSMB</u>	 Ja	 Ja	 Ja
<u>AWS::DataSync::Task</u>	 Ja	 Ja	 Ja
<u>AWS::DataZone::Connection</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::DataSource</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::Domain</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::DomainUnit</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::DataZone::Environment</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::EnvironmentActions</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::EnvironmentBlueprintConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::EnvironmentProfile</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::FormType</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::GroupProfile</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::DataZone::Owner</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::PolicyGrant</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::Project</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::ProjectMembership</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::ProjectProfile</u>	 Ja	 Ja	 Nein
<u>AWS::DataZone::SubscriptionTarget</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::DataZone::UserProfile</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::Farm</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::Fleet</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::LicenseEndpoint</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::Limit</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::MeteredProduct</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Deadline::Monitor</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::Queue</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::QueueEnvironment</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::QueueFleetAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::QueueLimitAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::Deadline::StorageProfile</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Detective::Graph</u>	 Ja	 Ja	 Ja
<u>AWS::Detective::MemberInvitation</u>	 Ja	 Ja	 Ja
<u>AWS::Detective::OrganizationAdmin</u>	 Ja	 Ja	 Ja
<u>AWS::DevOpsAgent::AgentSpace</u>	 Ja	 Ja	 Nein
<u>AWS::DevOpsAgent::Association</u>	 Ja	 Ja	 Nein
<u>AWS::DevOpsGuru::LogAnomalyDetection Integration</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::DevOpsGuru::NotificationChannel</u>	 Ja	 Ja	 Ja
<u>AWS::DevOpsGuru::ResourceCollection</u>	 Ja	 Ja	 Ja
<u>AWS::DeviceFarm::DevicePool</u>	 Ja	 Ja	 Nein
<u>AWS::DeviceFarm::InstanceProfile</u>	 Ja	 Ja	 Nein
<u>AWS::DeviceFarm::NetworkProfile</u>	 Ja	 Ja	 Nein
<u>AWS::DeviceFarm::Project</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::DeviceFarm::TestGridProject</u>	 Ja	 Ja	 Nein
<u>AWS::DeviceFarm::VPCEConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::DirectoryService::SimpleAD</u>	 Ja	 Ja	 Ja
<u>AWS::DocDBElastic::Cluster</u>	 Ja	 Ja	 Ja
<u>AWS::DynamoDB::GlobalTable</u>	 Ja	 Ja	 Nein
<u>AWS::DynamoDB::Table</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::EC2::CapacityManagerDataExport</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::CapacityReservation</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::CapacityReservationFleet</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::CarrierGateway</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::CustomerGateway</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::DHCPOptions</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::EC2Fleet</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::EIP</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::EIPAssociation</u>	 Ja	 Nein	 Ja
<u>AWS::EC2::EgressOnlyInternetGateway</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::EnclaveCertificateIamRoleAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::FlowLog</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
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
AWS::EC2::IPAMPoolCidr	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::EC2::IPAMResourceDiscovery</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::IPAMResourceDiscoveryAssoc iation</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::IPAMScope</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::Instance</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::InstanceConnectEndpoint</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::InternetGateway</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::EC2::IpPoolRouteTableAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::KeyPair</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::LaunchTemplate</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::LocalGatewayRoute</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::LocalGatewayRouteTable</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::LocalGatewayRouteTableVPCAssociation</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::LocalGatewayVirtualInterface</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::LocalGatewayVirtualInterfaceGroup</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::NatGateway</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::NetworkAcl</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::NetworkInsightsAccessScope</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::EC2::NetworkInsightsAccessScope Analysis</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::NetworkInsightsAnalysis</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::NetworkInsightsPath</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::NetworkInterface</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::NetworkInterfaceAttachment</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::NetworkPerformanceMetricSu bscription</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::PlacementGroup</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::PrefixList</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::Route</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::RouteServer</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::RouteServerAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::RouteServerEndpoint</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::RouteServerPeer</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::RouteServerPropagation</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::RouteTable</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::SecurityGroup</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::SecurityGroupEgress</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::SecurityGroupIngress</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::EC2::SecurityGroupVpcAssociatio n</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::SnapshotBlockPublicAccess</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::SpotFleet</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::Subnet</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::SubnetCidrBlock</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::SubnetNetworkAclAssociatio n</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::SubnetRouteTableAssociation</u>	 Ja	 Nein	 Ja
<u>AWS::EC2::TrafficMirrorFilter</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TrafficMirrorFilterRule</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TrafficMirrorSession</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TrafficMirrorTarget</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TransitGateway</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::TransitGatewayAttachment</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::TransitGatewayConnect</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::TransitGatewayConnectPeer</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TransitGatewayMeteringPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TransitGatewayMeteringPolicyEntry</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TransitGatewayMulticastDomain</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::EC2::TransitGatewayMulticastDomainAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::TransitGatewayMulticastGroupMember</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::TransitGatewayMulticastGroupSource</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::TransitGatewayPeeringAttachment</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::TransitGatewayRoute</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TransitGatewayRouteTable</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::TransitGatewayRouteTableAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TransitGatewayRouteTablePropagation</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::TransitGatewayVpcAttachment</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VPC</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VPCLockPublicAccessExclusion</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::VPCLockPublicAccessOptions</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::VPCCidrBlock</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::VPCDHCPOptionsAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VPCEncryptionControl</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::VPCEndpoint</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VPCEndpointConnectionNotification</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VPCEndpointService</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::VPCEndpointServicePermissions</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::VPCGatewayAttachment</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::VPCPeeringConnection</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VPNConcentrator</u>	 Ja	 Ja	 Nein
<u>AWS::EC2::VPNConnection</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VPNConnectionRoute</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::EC2::VPNGateway</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VerifiedAccessEndpoint</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VerifiedAccessGroup</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VerifiedAccessInstance</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::VerifiedAccessTrustProvide r</u>	 Ja	 Ja	 Ja
<u>AWS::EC2::Volume</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EC2::VolumeAttachment</u>	 Ja	 Ja	 Ja
<u>AWS::ECR::PublicRepository</u>	 Ja	 Ja	 Ja
<u>AWS::ECR::PullThroughCacheRule</u>	 Ja	 Ja	 Ja
<u>AWS::ECR::PullTimeUpdateExclusion</u>	 Ja	 Ja	 Nein
<u>AWS::ECR::RegistryPolicy</u>	 Ja	 Ja	 Ja
<u>AWS::ECR::RegistryScanningConfiguration</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ECR::ReplicationConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::ECR::Repository</u>	 Ja	 Ja	 Ja
<u>AWS::ECR::RepositoryCreationTemplate</u>	 Ja	 Ja	 Nein
<u>AWS::ECR::SigningConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::ECS::CapacityProvider</u>	 Ja	 Ja	 Nein
<u>AWS::ECS::Cluster</u>	 Ja	 Nein	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ECS::ClusterCapacityProviderAssociations</u>	 Ja	 Nein	 Ja
<u>AWS::ECS::ExpressGatewayService</u>	 Ja	 Ja	 Nein
<u>AWS::ECS::PrimaryTaskSet</u>	 Ja	 Ja	 Nein
<u>AWS::ECS::Service</u>	 Ja	 Ja	 Ja
<u>AWS::ECS::TaskDefinition</u>	 Ja	 Ja	 Ja
<u>AWS::ECS::TaskSet</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::EFS::AccessPoint</u>	 Ja	 Ja	 Ja
<u>AWS::EFS::FileSystem</u>	 Ja	 Ja	 Ja
<u>AWS::EFS::MountTarget</u>	 Ja	 Ja	 Ja
<u>AWS::EKS::AccessEntry</u>	 Ja	 Ja	 Nein
<u>AWS::EKS::Addon</u>	 Ja	 Ja	 Ja
<u>AWS::EKS::Capability</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EKS::Cluster</u>	 Ja	 Nein	 Ja
<u>AWS::EKS::FargateProfile</u>	 Ja	 Ja	 Ja
<u>AWS::EKS::IdentityProviderConfig</u>	 Ja	 Ja	 Ja
<u>AWS::EKS::Nodegroup</u>	 Ja	 Nein	 Ja
<u>AWS::EKS::PodIdentityAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::EMR::SecurityConfiguration</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::EMR::Step</u>	 Ja	 Nein	 Nein
<u>AWS::EMR::Studio</u>	 Ja	 Ja	 Ja
<u>AWS::EMR::StudioSessionMapping</u>	 Ja	 Ja	 Ja
<u>AWS::EMR::WALWorkspace</u>	 Ja	 Ja	 Nein
<u>AWS::EMRContainers::VirtualCluster</u>	 Ja	 Ja	 Nein
<u>AWS::EMRServerless::Application</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EVS::Environment</u>	 Ja	 Ja	 Nein
<u>AWS::ElastiCache::GlobalReplicationGroup</u>	 Ja	 Ja	 Nein
<u>AWS::ElastiCache::ParameterGroup</u>	 Ja	 Ja	 Nein
<u>AWS::ElastiCache::ServerlessCache</u>	 Ja	 Ja	 Nein
<u>AWS::ElastiCache::SubnetGroup</u>	 Ja	 Ja	 Ja
<u>AWS::ElastiCache::User</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ElastiCache::UserGroup</u>	 Ja	 Ja	 Ja
<u>AWS::ElasticBeanstalk::Application</u>	 Ja	 Ja	 Ja
<u>AWS::ElasticBeanstalk::ApplicationVersion</u>	 Ja	 Ja	 Ja
<u>AWS::ElasticBeanstalk::ConfigurationTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::ElasticBeanstalk::Environment</u>	 Ja	 Nein	 Nein
<u>AWS::ElasticLoadBalancingV2::Listener</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ElasticLoadBalancingV2::ListenerRule</u>	 Ja	 Nein	 Ja
<u>AWS::ElasticLoadBalancingV2::LoadBalancer</u>	 Ja	 Ja	 Ja
<u>AWS::ElasticLoadBalancingV2::TargetGroup</u>	 Ja	 Ja	 Ja
<u>AWS::ElasticLoadBalancingV2::TrustStore</u>	 Ja	 Ja	 Nein
<u>AWS::ElasticLoadBalancingV2::TrustStoreRevocation</u>	 Ja	 Ja	 Nein
<u>AWS::EntityResolution::IdMappingWorkflow</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EntityResolution::IdNamespace</u>	 Ja	 Ja	 Nein
<u>AWS::EntityResolution::MatchingWorkflow</u>	 Ja	 Ja	 Ja
<u>AWS::EntityResolution::PolicyStatement</u>	 Ja	 Ja	 Nein
<u>AWS::EntityResolution::SchemaMapping</u>	 Ja	 Ja	 Ja
<u>AWS::EventSchemas::Discoverer</u>	 Ja	 Ja	 Nein
<u>AWS::EventSchemas::Registry</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::EventSchemas::RegistryPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::EventSchemas::Schema</u>	 Ja	 Ja	 Nein
<u>AWS::Events::ApiDestination</u>	 Ja	 Ja	 Ja
<u>AWS::Events::Archive</u>	 Ja	 Ja	 Ja
<u>AWS::Events::Connection</u>	 Ja	 Ja	 Nein
<u>AWS::Events::Endpoint</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Events::EventBus</u>	 Ja	 Ja	 Ja
<u>AWS::Events::EventBusPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::Events::Rule</u>	 Ja	 Ja	 Ja
<u>AWS::Evidently::Experiment</u>	 Ja	 Ja	 Nein
<u>AWS::Evidently::Feature</u>	 Ja	 Ja	 Nein
<u>AWS::Evidently::Launch</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Evidently::Project</u>	 Ja	 Ja	 Nein
<u>AWS::Evidently::Segment</u>	 Ja	 Ja	 Nein
<u>AWS::FIS::ExperimentTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::FIS::TargetAccountConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::FMS::NotificationChannel</u>	 Ja	 Ja	 Nein
<u>AWS::FMS::Policy</u>	 Ja	 Nein	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::FMS::ResourceSet</u>	 Ja	 Ja	 Nein
<u>AWS::FSx::DataRepositoryAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::FSx::S3AccessPointAttachment</u>	 Ja	 Ja	 Nein
<u>AWS::FinSpace::Environment</u>	 Ja	 Ja	 Ja
<u>AWS::Forecast::Dataset</u>	 Ja	 Ja	 Nein
<u>AWS::Forecast::DatasetGroup</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::FraudDetector::Detector</u>	 Ja	 Ja	 Ja
<u>AWS::FraudDetector::EntityType</u>	 Ja	 Ja	 Ja
<u>AWS::FraudDetector::EventType</u>	 Ja	 Ja	 Ja
<u>AWS::FraudDetector::Label</u>	 Ja	 Ja	 Ja
<u>AWS::FraudDetector::List</u>	 Ja	 Ja	 Ja
<u>AWS::FraudDetector::Outcome</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::FraudDetector::Variable</u>	 Ja	 Ja	 Ja
<u>AWS::GameLift::Alias</u>	 Ja	 Ja	 Ja
<u>AWS::GameLift::Build</u>	 Ja	 Ja	 Ja
<u>AWS::GameLift::ContainerFleet</u>	 Ja	 Ja	 Nein
<u>AWS::GameLift::ContainerGroupDefinition</u>	 Ja	 Ja	 Nein
<u>AWS::GameLift::Fleet</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::GameLift::GameServerGroup</u>	 Ja	 Ja	 Nein
<u>AWS::GameLift::GameSessionQueue</u>	 Ja	 Ja	 Nein
<u>AWS::GameLift::Location</u>	 Ja	 Ja	 Nein
<u>AWS::GameLift::MatchmakingConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::GameLift::MatchmakingRuleSet</u>	 Ja	 Ja	 Nein
<u>AWS::GameLift::Script</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::GameLiftStreams::Application</u>	 Ja	 Ja	 Nein
<u>AWS::GameLiftStreams::StreamGroup</u>	 Ja	 Ja	 Nein
<u>AWS::GlobalAccelerator::Accelerator</u>	 Ja	 Ja	 Ja
<u>AWS::GlobalAccelerator::CrossAccountAttachment</u>	 Ja	 Ja	 Nein
<u>AWS::GlobalAccelerator::EndpointGroup</u>	 Ja	 Ja	 Ja
<u>AWS::GlobalAccelerator::Listener</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Glue::Crawler</u>	 Ja	 Ja	 Nein
<u>AWS::Glue::Database</u>	 Ja	 Ja	 Nein
<u>AWS::Glue::IdentityCenterConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::Glue::Integration</u>	 Ja	 Ja	 Nein
<u>AWS::Glue::IntegrationResourceProperty</u>	 Ja	 Ja	 Nein
<u>AWS::Glue::Job</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Glue::Registry</u>	 Ja	 Ja	 Ja
<u>AWS::Glue::Schema</u>	 Ja	 Ja	 Ja
<u>AWS::Glue::SchemaVersion</u>	 Ja	 Ja	 Ja
<u>AWS::Glue::SchemaVersionMetadata</u>	 Ja	 Ja	 Ja
<u>AWS::Glue::Trigger</u>	 Ja	 Ja	 Nein
<u>AWS::Glue::UsageProfile</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Grafana::Workspace</u>	 Ja	 Ja	 Ja
<u>AWS::GreengrassV2::ComponentVersion</u>	 Ja	 Ja	 Nein
<u>AWS::GreengrassV2::Deployment</u>	 Ja	 Ja	 Ja
<u>AWS::GroundStation::Config</u>	 Ja	 Ja	 Ja
<u>AWS::GroundStation::DataflowEndpoint Group</u>	 Ja	 Ja	 Ja
<u>AWS::GroundStation::DataflowEndpoint GroupV2</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::GroundStation::MissionProfile</u>	 Ja	 Ja	 Ja
<u>AWS::GuardDuty::Detector</u>	 Ja	 Ja	 Ja
<u>AWS::GuardDuty::Filter</u>	 Ja	 Ja	 Nein
<u>AWS::GuardDuty::IPSet</u>	 Ja	 Ja	 Nein
<u>AWS::GuardDuty::MalwareProtectionPlan</u>	 Ja	 Ja	 Nein
<u>AWS::GuardDuty::Master</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::GuardDuty::Member</u>	 Ja	 Ja	 Nein
<u>AWS::GuardDuty::PublishingDestination</u>	 Ja	 Ja	 Nein
<u>AWS::GuardDuty::ThreatEntitySet</u>	 Ja	 Ja	 Nein
<u>AWS::GuardDuty::ThreatIntelSet</u>	 Ja	 Ja	 Nein
<u>AWS::GuardDuty::TrustedEntitySet</u>	 Ja	 Ja	 Nein
<u>AWS::HealthImaging::Datastore</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::HealthLake::FHIRDatastore</u>	 Ja	 Ja	 Ja
<u>AWS::IAM::Group</u>	 Ja	 Ja	 Ja
<u>AWS::IAM::GroupPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::IAM::InstanceProfile</u>	 Ja	 Ja	 Ja
<u>AWS::IAM::ManagedPolicy</u>	 Ja	 Ja	 Ja
<u>AWS::IAM::OIDCProvider</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::IAM::Role</u>	 Ja	 Ja	 Ja
<u>AWS::IAM::RolePolicy</u>	 Ja	 Ja	 Nein
<u>AWS::IAM::SAMLProvider</u>	 Ja	 Ja	 Ja
<u>AWS::IAM::ServerCertificate</u>	 Ja	 Ja	 Ja
<u>AWS::IAM::ServiceLinkedRole</u>	 Ja	 Ja	 Nein
<u>AWS::IAM::User</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::IAM::UserPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::IAM::VirtualMFADevice</u>	 Ja	 Ja	 Nein
<u>AWS::IVS::Channel</u>	 Ja	 Ja	 Ja
<u>AWS::IVS::EncoderConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::IVS::IngestConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::IVS::PlaybackKeyPair</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::IVS::PlaybackRestrictionPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::IVS::PublicKey</u>	 Ja	 Ja	 Nein
<u>AWS::IVS::RecordingConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::IVS::Stage</u>	 Ja	 Ja	 Nein
<u>AWS::IVS::StorageConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::IVS::StreamKey</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::IVSChat::LoggingConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::IVSChat::Room</u>	 Ja	 Ja	 Ja
<u>AWS::IdentityStore::Group</u>	 Ja	 Ja	 Ja
<u>AWS::IdentityStore::GroupMembership</u>	 Ja	 Ja	 Ja
<u>AWS::ImageBuilder::Component</u>	 Ja	 Ja	 Ja
<u>AWS::ImageBuilder::ContainerRecipe</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ImageBuilder::DistributionConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::ImageBuilder::Image</u>	 Ja	 Ja	 Ja
<u>AWS::ImageBuilder::ImagePipeline</u>	 Ja	 Ja	 Ja
<u>AWS::ImageBuilder::ImageRecipe</u>	 Ja	 Ja	 Ja
<u>AWS::ImageBuilder::InfrastructureConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::ImageBuilder::LifecyclePolicy</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ImageBuilder::Workflow</u>	 Ja	 Ja	 Nein
<u>AWS::Inspector::AssessmentTarget</u>	 Ja	 Ja	 Ja
<u>AWS::Inspector::AssessmentTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::Inspector::ResourceGroup</u>	 Ja	 Ja	 Nein
<u>AWS::InspectorV2::CisScanConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::InspectorV2::CodeSecurityIntegration</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::InspectorV2::CodeSecurityScanConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::InspectorV2::Filter</u>	 Ja	 Ja	 Nein
<u>AWS::InternetMonitor::Monitor</u>	 Ja	 Ja	 Ja
<u>AWS::Invoicing::InvoiceUnit</u>	 Ja	 Ja	 Nein
<u>AWS::IoT::AccountAuditConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::Authorizer</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::IoT::BillingGroup</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::CACertificate</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::Certificate</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::CertificateProvider</u>	 Ja	 Ja	 Nein
<u>AWS::IoT::Command</u>	 Ja	 Ja	 Nein
<u>AWS::IoT::CustomMetric</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::IoT::Dimension</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::DomainConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::IoT::EncryptionConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::IoT::FleetMetric</u>	 Ja	 Ja	 Nein
<u>AWS::IoT::JobTemplate</u>	 Ja	 Ja	 Nein
<u>AWS::IoT::Logging</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::IoT::MitigationAction</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::Policy</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::ProvisioningTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::ResourceSpecificLogging</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::RoleAlias</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::ScheduledAudit</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::IoT::SecurityProfile</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::SoftwarePackage</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::SoftwarePackageVersion</u>	 Ja	 Ja	 Nein
<u>AWS::IoT::Thing</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::ThingGroup</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::ThingType</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::IoT::TopicRule</u>	 Ja	 Ja	 Ja
<u>AWS::IoT::TopicRuleDestination</u>	 Ja	 Ja	 Ja
<u>AWS::IoTAnalytics::Channel</u>	 Ja	 Ja	 Ja
<u>AWS::IoTAnalytics::Dataset</u>	 Ja	 Ja	 Ja
<u>AWS::IoTAnalytics::Datastore</u>	 Ja	 Ja	 Ja
<u>AWS::IoTAnalytics::Pipeline</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::IoTCoreDeviceAdvisor::SuiteDefinition</u>	 Ja	 Ja	 Ja
<u>AWS::IoTEvents::AlarmModel</u>	 Ja	 Ja	 Ja
<u>AWS::IoTEvents::DetectorModel</u>	 Ja	 Ja	 Ja
<u>AWS::IoTEvents::Input</u>	 Ja	 Ja	 Ja
<u>AWS::IoTFleetWise::Campaign</u>	 Ja	 Ja	 Ja
<u>AWS::IoTFleetWise::DecoderManifest</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::IoT FleetWise::Fleet</u>	 Ja	 Ja	 Ja
<u>AWS::IoT FleetWise::ModelManifest</u>	 Ja	 Ja	 Ja
<u>AWS::IoT FleetWise::SignalCatalog</u>	 Ja	 Ja	 Ja
<u>AWS::IoT FleetWise::StateTemplate</u>	 Ja	 Ja	 Nein
<u>AWS::IoT FleetWise::Vehicle</u>	 Ja	 Ja	 Ja
<u>AWS::IoT SiteWise::AccessPolicy</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::IoTSiteWise::Asset</u>	 Ja	 Ja	 Ja
<u>AWS::IoTSiteWise::AssetModel</u>	 Ja	 Ja	 Ja
<u>AWS::IoTSiteWise::ComputationModel</u>	 Ja	 Ja	 Nein
<u>AWS::IoTSiteWise::Dashboard</u>	 Ja	 Ja	 Nein
<u>AWS::IoTSiteWise::Dataset</u>	 Ja	 Ja	 Nein
<u>AWS::IoTSiteWise::Gateway</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::IoTSiteWise::Portal</u>	 Ja	 Ja	 Nein
<u>AWS::IoTSiteWise::Project</u>	 Ja	 Ja	 Nein
<u>AWS::IoTtwinMaker::ComponentType</u>	 Ja	 Ja	 Ja
<u>AWS::IoTtwinMaker::Entity</u>	 Ja	 Ja	 Ja
<u>AWS::IoTtwinMaker::Scene</u>	 Ja	 Ja	 Ja
<u>AWS::IoTtwinMaker::SyncJob</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::IoT TwinMaker::Workspace</u>	 Ja	 Ja	 Ja
<u>AWS::IoT Wireless::Destination</u>	 Ja	 Ja	 Ja
<u>AWS::IoT Wireless::DeviceProfile</u>	 Ja	 Ja	 Ja
<u>AWS::IoT Wireless::FirmwareTask</u>	 Ja	 Ja	 Ja
<u>AWS::IoT Wireless::MulticastGroup</u>	 Ja	 Ja	 Ja
<u>AWS::IoT Wireless::NetworkAnalyzerCon figuration</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::IoTWireless::PartnerAccount</u>	 Ja	 Ja	 Ja
<u>AWS::IoTWireless::ServiceProfile</u>	 Ja	 Ja	 Ja
<u>AWS::IoTWireless::TaskDefinition</u>	 Ja	 Ja	 Ja
<u>AWS::IoTWireless::WirelessDevice</u>	 Ja	 Ja	 Ja
<u>AWS::IoTWireless::WirelessDeviceImportTask</u>	 Ja	 Ja	 Ja
<u>AWS::IoTWireless::WirelessGateway</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::KMS::Alias</u>	 Ja	 Nein	 Ja
<u>AWS::KMS::Key</u>	 Ja	 Nein	 Ja
<u>AWS::KMS::ReplicaKey</u>	 Ja	 Nein	 Ja
<u>AWS::KafkaConnect::Connector</u>	 Ja	 Nein	 Nein
<u>AWS::KafkaConnect::CustomPlugin</u>	 Ja	 Ja	 Nein
<u>AWS::KafkaConnect::WorkerConfiguration</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Kendra::DataSource</u>	 Ja	 Ja	 Ja
<u>AWS::Kendra::FAQ</u>	 Ja	 Ja	 Ja
<u>AWS::Kendra::Index</u>	 Ja	 Ja	 Ja
<u>AWS::KendraRanking::ExecutionPlan</u>	 Ja	 Ja	 Ja
<u>AWS::Kinesis::ResourcePolicy</u>	 Ja	 Ja	 Nein
<u>AWS::Kinesis::Stream</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Kinesis::StreamConsumer</u>	 Ja	 Ja	 Nein
<u>AWS::KinesisAnalyticsV2::Application</u>	 Ja	 Ja	 Nein
<u>AWS::KinesisFirehose::DeliveryStream</u>	 Ja	 Nein	 Nein
<u>AWS::KinesisVideo::SignalingChannel</u>	 Ja	 Ja	 Nein
<u>AWS::KinesisVideo::Stream</u>	 Ja	 Ja	 Nein
<u>AWS::LakeFormation::DataCellsFilter</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::LakeFormation::PrincipalPermissions</u>	 Ja	 Ja	 Nein
<u>AWS::LakeFormation::Tag</u>	 Ja	 Ja	 Nein
<u>AWS::LakeFormation::TagAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::Lambda::Alias</u>	 Ja	 Ja	 Nein
<u>AWS::Lambda::CapacityProvider</u>	 Ja	 Ja	 Nein
<u>AWS::Lambda::CodeSigningConfig</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Lambda::EventInvokeConfig</u>	 Ja	 Ja	 Nein
<u>AWS::Lambda::EventSourceMapping</u>	 Ja	 Ja	 Ja
<u>AWS::Lambda::Function</u>	 Ja	 Ja	 Ja
<u>AWS::Lambda::LayerVersion</u>	 Ja	 Ja	 Nein
<u>AWS::Lambda::LayerVersionPermission</u>	 Ja	 Ja	 Ja
<u>AWS::Lambda::Permission</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Lambda::Url</u>	 Ja	 Ja	 Ja
<u>AWS::Lambda::Version</u>	 Ja	 Ja	 Ja
<u>AWS::LaunchWizard::Deployment</u>	 Ja	 Ja	 Nein
<u>AWS::Lex::Bot</u>	 Ja	 Ja	 Nein
<u>AWS::Lex::BotAlias</u>	 Ja	 Ja	 Nein
<u>AWS::Lex::BotVersion</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Lex::ResourcePolicy</u>	 Ja	 Ja	 Nein
<u>AWS::LicenseManager::Grant</u>	 Ja	 Ja	 Nein
<u>AWS::LicenseManager::License</u>	 Ja	 Ja	 Nein
<u>AWS::Lightsail::Alarm</u>	 Ja	 Ja	 Ja
<u>AWS::Lightsail::Bucket</u>	 Ja	 Ja	 Ja
<u>AWS::Lightsail::Certificate</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Lightsail::Container</u>	 Ja	 Ja	 Ja
<u>AWS::Lightsail::Database</u>	 Ja	 Ja	 Nein
<u>AWS::Lightsail::Disk</u>	 Ja	 Ja	 Ja
<u>AWS::Lightsail::DiskSnapshot</u>	 Ja	 Ja	 Nein
<u>AWS::Lightsail::Distribution</u>	 Ja	 Ja	 Ja
<u>AWS::Lightsail::Domain</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Lightsail::Instance</u>	 Ja	 Ja	 Ja
<u>AWS::Lightsail::InstanceSnapshot</u>	 Ja	 Ja	 Nein
<u>AWS::Lightsail::LoadBalancer</u>	 Ja	 Ja	 Ja
<u>AWS::Lightsail::LoadBalancerTlsCertificate</u>	 Ja	 Ja	 Ja
<u>AWS::Lightsail::StaticIp</u>	 Ja	 Ja	 Ja
<u>AWS::Location::APIKey</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Location::GeofenceCollection</u>	 Ja	 Ja	 Ja
<u>AWS::Location::Map</u>	 Ja	 Ja	 Ja
<u>AWS::Location::PlaceIndex</u>	 Ja	 Ja	 Ja
<u>AWS::Location::RouteCalculator</u>	 Ja	 Ja	 Ja
<u>AWS::Location::Tracker</u>	 Ja	 Ja	 Ja
<u>AWS::Location::TrackerConsumer</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Logs::AccountPolicy</u>	 Ja	 Ja	 Ja
<u>AWS::Logs::Delivery</u>	 Ja	 Ja	 Nein
<u>AWS::Logs::DeliveryDestination</u>	 Ja	 Ja	 Nein
<u>AWS::Logs::DeliverySource</u>	 Ja	 Ja	 Nein
<u>AWS::Logs::Destination</u>	 Ja	 Ja	 Ja
<u>AWS::Logs::Integration</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Logs::LogAnomalyDetector</u>	 Ja	 Ja	 Nein
<u>AWS::Logs::LogGroup</u>	 Ja	 Ja	 Ja
<u>AWS::Logs::LogStream</u>	 Ja	 Ja	 Ja
<u>AWS::Logs::MetricFilter</u>	 Ja	 Ja	 Ja
<u>AWS::Logs::QueryDefinition</u>	 Ja	 Ja	 Nein
<u>AWS::Logs::ResourcePolicy</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Logs::SubscriptionFilter</u>	 Ja	 Ja	 Ja
<u>AWS::Logs::Transformer</u>	 Ja	 Ja	 Nein
<u>AWS::LookoutEquipment::InferenceScheduler</u>	 Ja	 Ja	 Nein
<u>AWS::LookoutVision::Project</u>	 Ja	 Ja	 Ja
<u>AWS::M2::Application</u>	 Ja	 Ja	 Nein
<u>AWS::M2::Deployment</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::M2::Environment</u>	 Ja	 Ja	 Ja
<u>AWS::MPA::ApprovalTeam</u>	 Ja	 Ja	 Nein
<u>AWS::MPA::IdentitySource</u>	 Ja	 Ja	 Nein
<u>AWS::MSK::BatchScramSecret</u>	 Ja	 Ja	 Ja
<u>AWS::MSK::Cluster</u>	 Ja	 Ja	 Ja
<u>AWS::MSK::ClusterPolicy</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::MSK::Configuration</u>	 Ja	 Ja	 Ja
<u>AWS::MSK::Replicator</u>	 Ja	 Ja	 Ja
<u>AWS::MSK::ServerlessCluster</u>	 Ja	 Ja	 Ja
<u>AWS::MSK::VpcConnection</u>	 Ja	 Ja	 Ja
<u>AWS::MWAA::Environment</u>	 Ja	 Ja	 Nein
<u>AWS::Macie::AllowList</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Macie::CustomDataIdentifier</u>	 Ja	 Ja	 Nein
<u>AWS::Macie::FindingsFilter</u>	 Ja	 Ja	 Nein
<u>AWS::Macie::Session</u>	 Ja	 Ja	 Ja
<u>AWS::ManagedBlockchain::Accessor</u>	 Ja	 Ja	 Ja
<u>AWS::MediaConnect::Bridge</u>	 Ja	 Ja	 Ja
<u>AWS::MediaConnect::BridgeOutput</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::MediaConnect::BridgeSource</u>	 Ja	 Ja	 Nein
<u>AWS::MediaConnect::Flow</u>	 Ja	 Ja	 Nein
<u>AWS::MediaConnect::FlowEntitlement</u>	 Ja	 Ja	 Nein
<u>AWS::MediaConnect::FlowOutput</u>	 Ja	 Ja	 Nein
<u>AWS::MediaConnect::FlowSource</u>	 Ja	 Ja	 Nein
<u>AWS::MediaConnect::FlowVpcInterface</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::MediaConnect::Gateway</u>	 Ja	 Ja	 Ja
<u>AWS::MediaConnect::RouterInput</u>	 Ja	 Ja	 Nein
<u>AWS::MediaConnect::RouterNetworkInterface</u>	 Ja	 Ja	 Nein
<u>AWS::MediaConnect::RouterOutput</u>	 Ja	 Ja	 Nein
<u>AWS::MediaLive::ChannelPlacementGroup</u>	 Ja	 Ja	 Nein
<u>AWS::MediaLive::CloudWatchAlarmTemplate</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::MediaLive::CloudWatchAlarmTemplateGroup</u>	 Ja	 Ja	 Nein
<u>AWS::MediaLive::Cluster</u>	 Ja	 Ja	 Nein
<u>AWS::MediaLive::EventBridgeRuleTemplate</u>	 Ja	 Ja	 Nein
<u>AWS::MediaLive::EventBridgeRuleTemplateGroup</u>	 Ja	 Ja	 Nein
<u>AWS::MediaLive::Multiplex</u>	 Ja	 Ja	 Nein
<u>AWS::MediaLive::Multiplexprogram</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::MediaLive::Network</u>	 Ja	 Ja	 Nein
<u>AWS::MediaLive::SdiSource</u>	 Ja	 Ja	 Nein
<u>AWS::MediaLive::SignalMap</u>	 Ja	 Ja	 Nein
<u>AWS::MediaPackage::Asset</u>	 Ja	 Ja	 Ja
<u>AWS::MediaPackage::Channel</u>	 Ja	 Ja	 Ja
<u>AWS::MediaPackage::OriginEndpoint</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::MediaPackage::PackagingConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::MediaPackage::PackagingGroup</u>	 Ja	 Ja	 Ja
<u>AWS::MediaPackageV2::Channel</u>	 Ja	 Ja	 Ja
<u>AWS::MediaPackageV2::ChannelGroup</u>	 Ja	 Ja	 Ja
<u>AWS::MediaPackageV2::ChannelPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::MediaPackageV2::OriginEndpoint</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::MediaPackageV2::OriginEndpointPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::MediaTailor::Channel</u>	 Ja	 Ja	 Nein
<u>AWS::MediaTailor::ChannelPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::MediaTailor::LiveSource</u>	 Ja	 Ja	 Ja
<u>AWS::MediaTailor::PlaybackConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::MediaTailor::SourceLocation</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::MediaTailor::VodSource</u>	 Ja	 Ja	 Ja
<u>AWS::MemoryDB::ACL</u>	 Ja	 Ja	 Ja
<u>AWS::MemoryDB::Cluster</u>	 Ja	 Ja	 Ja
<u>AWS::MemoryDB::MultiRegionCluster</u>	 Ja	 Ja	 Nein
<u>AWS::MemoryDB::ParameterGroup</u>	 Ja	 Ja	 Ja
<u>AWS::MemoryDB::SubnetGroup</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::MemoryDB::User</u>	 Ja	 Ja	 Ja
<u>AWS::Neptune::DBCluster</u>	 Ja	 Ja	 Ja
<u>AWS::Neptune::DBClusterParameterGroup</u>	 Ja	 Ja	 Nein
<u>AWS::Neptune::DBInstance</u>	 Ja	 Ja	 Nein
<u>AWS::Neptune::DBParameterGroup</u>	 Ja	 Ja	 Nein
<u>AWS::Neptune::DBSubnetGroup</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Neptune::EventSubscription</u>	 Ja	 Ja	 Nein
<u>AWS::NeptuneGraph::Graph</u>	 Ja	 Ja	 Nein
<u>AWS::NeptuneGraph::PrivateGraphEndpoint</u>	 Ja	 Ja	 Nein
<u>AWS::NetworkFirewall::Firewall</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkFirewall::FirewallPolicy</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkFirewall::LoggingConfiguration</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::NetworkFirewall::RuleGroup</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkFirewall::TLSInspectionConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::NetworkFirewall::VpcEndpointAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::NetworkManager::ConnectAttachment</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::ConnectPeer</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::CoreNetwork</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::NetworkManager::CoreNetworkPrefixListAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::NetworkManager::CustomerGatewayAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::Device</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::DirectConnectGatewayAttachment</u>	 Ja	 Ja	 Nein
<u>AWS::NetworkManager::GlobalNetwork</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::Link</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::NetworkManager::LinkAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::Site</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::SiteToSiteVpnAttachment</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::TransitGatewayPeering</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::TransitGatewayRegistration</u>	 Ja	 Ja	 Ja
<u>AWS::NetworkManager::TransitGatewayRouteTableAttachment</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
<u>AWS::NetworkManager::VpcAttachment</u>	 Ja	 Ja	 Ja
<u>AWS::Notifications::ChannelAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::Notifications::EventRule</u>	 Ja	 Ja	 Nein
<u>AWS::Notifications::ManagedNotificationAccountContactAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::Notifications::ManagedNotificationAdditionalChannelAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::Notifications::NotificationConfiguration</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Notifications::NotificationHub</u>	 Ja	 Ja	 Nein
<u>AWS::Notifications::OrganizationalUnitAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::NotificationsContacts::EmailContact</u>	 Ja	 Ja	 Nein
<u>AWS::ODB::CloudAutonomousVmCluster</u>	 Ja	 Ja	 Nein
<u>AWS::ODB::CloudExadataInfrastructure</u>	 Ja	 Ja	 Nein
<u>AWS::ODB::CloudVmCluster</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ODB::OdbNetwork</u>	 Ja	 Ja	 Nein
<u>AWS::ODB::OdbPeeringConnection</u>	 Ja	 Ja	 Nein
<u>AWS::OSIS::Pipeline</u>	 Ja	 Ja	 Ja
<u>AWS::Oam::Link</u>	 Ja	 Ja	 Ja
<u>AWS::Oam::Sink</u>	 Ja	 Ja	 Ja
<u>AWS::ObservabilityAdmin::OrganizationCentralizationRule</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ObservabilityAdmin::OrganizationTelemetryRule</u>	 Ja	 Ja	 Nein
<u>AWS::ObservabilityAdmin::S3TableIntegration</u>	 Ja	 Ja	 Nein
<u>AWS::ObservabilityAdmin::TelemetryPipelines</u>	 Ja	 Ja	 Nein
<u>AWS::ObservabilityAdmin::TelemetryRule</u>	 Ja	 Ja	 Nein
<u>AWS::Omics::AnnotationStore</u>	 Ja	 Ja	 Ja
<u>AWS::Omics::ReferenceStore</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Omics::RunGroup</u>	 Ja	 Ja	 Ja
<u>AWS::Omics::SequenceStore</u>	 Ja	 Ja	 Ja
<u>AWS::Omics::VariantStore</u>	 Ja	 Ja	 Ja
<u>AWS::Omics::Workflow</u>	 Ja	 Ja	 Ja
<u>AWS::Omics::WorkflowVersion</u>	 Ja	 Ja	 Nein
<u>AWS::OpenSearchServerless::AccessPolicy</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::OpenSearchServerless::Collection</u>	 Ja	 Ja	 Nein
<u>AWS::OpenSearchServerless::CollectionGroup</u>	 Ja	 Ja	 Nein
<u>AWS::OpenSearchServerless::Index</u>	 Ja	 Ja	 Nein
<u>AWS::OpenSearchServerless::LifecyclePolicy</u>	 Ja	 Ja	 Nein
<u>AWS::OpenSearchServerless::SecurityConfig</u>	 Ja	 Ja	 Ja
<u>AWS::OpenSearchServerless::SecurityPolicy</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::OpenSearchServerless::VpcEndpoint</u>	 Ja	 Ja	 Ja
<u>AWS::OpenSearchService::Application</u>	 Ja	 Ja	 Nein
<u>AWS::OpenSearchService::Domain</u>	 Ja	 Ja	 Nein
<u>AWS::Organizations::Account</u>	 Ja	 Ja	 Ja
<u>AWS::Organizations::Organization</u>	 Ja	 Ja	 Ja
<u>AWS::Organizations::OrganizationalUnit</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Organizations::Policy</u>	 Ja	 Ja	 Ja
<u>AWS::Organizations::ResourcePolicy</u>	 Ja	 Ja	 Ja
<u>AWS::PCAManagement::Connector</u>	 Ja	 Ja	 Ja
<u>AWS::PCAManagement::DirectoryRegist ration</u>	 Ja	 Ja	 Ja
<u>AWS::PCAManagement::ServicePrincipa lName</u>	 Ja	 Ja	 Ja
<u>AWS::PCAManagement::Template</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::PCAConnectorAD::TemplateGroupAccessControlEntry</u>	 Ja	 Ja	 Ja
<u>AWS::PCAConnectorSCEP::Challenge</u>	 Ja	 Ja	 Nein
<u>AWS::PCAConnectorSCEP::Connector</u>	 Ja	 Ja	 Nein
<u>AWS::PCS::Cluster</u>	 Ja	 Ja	 Nein
<u>AWS::PCS::ComputeNodeGroup</u>	 Ja	 Ja	 Nein
<u>AWS::PCS::Queue</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Panorama::ApplicationInstance</u>	 Ja	 Ja	 Ja
<u>AWS::Panorama::Package</u>	 Ja	 Ja	 Ja
<u>AWS::Panorama::PackageVersion</u>	 Ja	 Ja	 Nein
<u>AWS::PaymentCryptography::Alias</u>	 Ja	 Ja	 Nein
<u>AWS::PaymentCryptography::Key</u>	 Ja	 Ja	 Nein
<u>AWS::Personalize::Dataset</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Personalize::DatasetGroup</u>	 Ja	 Ja	 Ja
<u>AWS::Personalize::Schema</u>	 Ja	 Ja	 Ja
<u>AWS::Personalize::Solution</u>	 Ja	 Ja	 Nein
<u>AWS::Pinpoint::InAppTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::Pipes::Pipe</u>	 Ja	 Ja	 Ja
<u>AWS::Proton::EnvironmentAccountConnection</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Proton::EnvironmentTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::Proton::ServiceTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::QBusiness::Application</u>	 Ja	 Ja	 Nein
<u>AWS::QBusiness::DataAccessor</u>	 Ja	 Ja	 Nein
<u>AWS::QBusiness::DataSource</u>	 Ja	 Ja	 Nein
<u>AWS::QBusiness::Index</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::QBusiness::Permission</u>	 Ja	 Ja	 Nein
<u>AWS::QBusiness::Plugin</u>	 Ja	 Ja	 Nein
<u>AWS::QBusiness::Retriever</u>	 Ja	 Ja	 Nein
<u>AWS::QBusiness::WebExperience</u>	 Ja	 Ja	 Nein
<u>AWS::QLDB::Stream</u>	 Ja	 Ja	 Ja
<u>AWS::QuickSight::Analysis</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::QuickSight::CustomPermissions</u>	 Ja	 Ja	 Nein
<u>AWS::QuickSight::Dashboard</u>	 Ja	 Ja	 Nein
<u>AWS::QuickSight::DataSet</u>	 Ja	 Ja	 Ja
<u>AWS::QuickSight::DataSource</u>	 Ja	 Ja	 Ja
<u>AWS::QuickSight::Folder</u>	 Ja	 Ja	 Nein
<u>AWS::QuickSight::RefreshSchedule</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::QuickSight::Template</u>	 Ja	 Ja	 Nein
<u>AWS::QuickSight::Theme</u>	 Ja	 Ja	 Nein
<u>AWS::QuickSight::Topic</u>	 Ja	 Ja	 Ja
<u>AWS::QuickSight::VPCConnection</u>	 Ja	 Ja	 Ja
<u>AWS::RAM::Permission</u>	 Ja	 Ja	 Nein
<u>AWS::RAM::ResourceShare</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
<u>AWS::RDS::CustomDBEngineVersion</u>	 Ja	 Ja	 Nein
<u>AWS::RDS::DBCluster</u>	 Ja	 Ja	 Ja
<u>AWS::RDS::DBClusterParameterGroup</u>	 Ja	 Ja	 Nein
<u>AWS::RDS::DBInstance</u>	 Ja	 Ja	 Ja
<u>AWS::RDS::DBParameterGroup</u>	 Ja	 Ja	 Nein
<u>AWS::RDS::DBProxy</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::RDS::DBProxyEndpoint</u>	 Ja	 Ja	 Ja
<u>AWS::RDS::DBProxyTargetGroup</u>	 Ja	 Ja	 Ja
<u>AWS::RDS::DBShardGroup</u>	 Ja	 Ja	 Nein
<u>AWS::RDS::DBSubnetGroup</u>	 Ja	 Ja	 Ja
<u>AWS::RDS::EventSubscription</u>	 Ja	 Ja	 Ja
<u>AWS::RDS::GlobalCluster</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::RDS::Integration</u>	 Ja	 Ja	 Nein
<u>AWS::RDS::OptionGroup</u>	 Ja	 Ja	 Ja
<u>AWS::RTBFabric::InboundExternalLink</u>	 Ja	 Ja	 Nein
<u>AWS::RTBFabric::Link</u>	 Ja	 Ja	 Nein
<u>AWS::RTBFabric::OutboundExternalLink</u>	 Ja	 Ja	 Nein
<u>AWS::RTBFabric::RequesterGateway</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::RTBFabric::ResponderGateway</u>	 Ja	 Ja	 Nein
<u>AWS::RUM::AppMonitor</u>	 Ja	 Ja	 Ja
<u>AWS::Rbin::Rule</u>	 Ja	 Ja	 Nein
<u>AWS::Redshift::Cluster</u>	 Ja	 Ja	 Ja
<u>AWS::Redshift::ClusterParameterGroup</u>	 Ja	 Ja	 Nein
<u>AWS::Redshift::ClusterSubnetGroup</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Redshift::EndpointAccess</u>	 Ja	 Ja	 Ja
<u>AWS::Redshift::EndpointAuthorization</u>	 Ja	 Ja	 Ja
<u>AWS::Redshift::EventSubscription</u>	 Ja	 Ja	 Nein
<u>AWS::Redshift::Integration</u>	 Ja	 Ja	 Nein
<u>AWS::Redshift::ScheduledAction</u>	 Ja	 Ja	 Ja
<u>AWS::RedshiftServerless::Namespace</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::RedshiftServerless::Snapshot</u>	 Ja	 Ja	 Nein
<u>AWS::RedshiftServerless::Workgroup</u>	 Ja	 Ja	 Nein
<u>AWS::RefactorSpaces::Application</u>	 Ja	 Ja	 Ja
<u>AWS::RefactorSpaces::Environment</u>	 Ja	 Ja	 Ja
<u>AWS::RefactorSpaces::Route</u>	 Ja	 Ja	 Ja
<u>AWS::RefactorSpaces::Service</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Rekognition::Collection</u>	 Ja	 Ja	 Ja
<u>AWS::Rekognition::Project</u>	 Ja	 Ja	 Ja
<u>AWS::Rekognition::StreamProcessor</u>	 Ja	 Ja	 Ja
<u>AWS::ResilienceHub::App</u>	 Ja	 Ja	 Ja
<u>AWS::ResilienceHub::ResiliencyPolicy</u>	 Ja	 Ja	 Ja
<u>AWS::ResourceExplorer2::DefaultViewAs sociation</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ResourceExplorer2::Index</u>	 Ja	 Ja	 Ja
<u>AWS::ResourceExplorer2::View</u>	 Ja	 Ja	 Ja
<u>AWS::ResourceGroups::Group</u>	 Ja	 Ja	 Ja
<u>AWS::ResourceGroups::TagSyncTask</u>	 Ja	 Ja	 Nein
<u>AWS::RolesAnywhere::CRL</u>	 Ja	 Ja	 Ja
<u>AWS::RolesAnywhere::Profile</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::RolesAnywhere::TrustAnchor</u>	 Ja	 Ja	 Nein
<u>AWS::Route53::CidrCollection</u>	 Ja	 Ja	 Ja
<u>AWS::Route53::DNSSEC</u>	 Ja	 Ja	 Ja
<u>AWS::Route53::HealthCheck</u>	 Ja	 Ja	 Ja
<u>AWS::Route53::HostedZone</u>	 Ja	 Ja	 Ja
<u>AWS::Route53::KeySigningKey</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Route53Profiles::Profile</u>	 Ja	 Ja	 Nein
<u>AWS::Route53Profiles::ProfileAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::Route53Profiles::ProfileResourceAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::Route53RecoveryControl::Cluster</u>	 Ja	 Ja	 Ja
<u>AWS::Route53RecoveryControl::ControlPanel</u>	 Ja	 Ja	 Ja
<u>AWS::Route53RecoveryControl::RoutingControl</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
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
AWS::Route53Resolver::FirewallDomainList	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
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
AWS::Route53Resolver::ResolverEndpoint	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Route53Resolver::ResolverQueryLoggingConfig</u>	 Ja	 Ja	 Ja
<u>AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::Route53Resolver::ResolverRule</u>	 Ja	 Ja	 Nein
<u>AWS::Route53Resolver::ResolverRuleAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::S3::AccessGrant</u>	 Ja	 Ja	 Nein
<u>AWS::S3::AccessGrantsInstance</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::S3::AccessGrantsLocation</u>	 Ja	 Ja	 Nein
<u>AWS::S3::AccessPoint</u>	 Ja	 Ja	 Ja
<u>AWS::S3::Bucket</u>	 Ja	 Ja	 Ja
<u>AWS::S3::BucketPolicy</u>	 Ja	 Nein	 Ja
<u>AWS::S3::MultiRegionAccessPoint</u>	 Ja	 Ja	 Ja
<u>AWS::S3::MultiRegionAccessPointPolicy</u>	 Ja	 Nein	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::S3::StorageLens</u>	 Ja	 Ja	 Ja
<u>AWS::S3::StorageLensGroup</u>	 Ja	 Ja	 Nein
<u>AWS::S3Express::AccessPoint</u>	 Ja	 Ja	 Nein
<u>AWS::S3Express::BucketPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::S3Express::DirectoryBucket</u>	 Ja	 Ja	 Nein
<u>AWS::S3ObjectLambda::AccessPoint</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::S3ObjectLambda::AccessPointPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::S3Outposts::AccessPoint</u>	 Ja	 Ja	 Ja
<u>AWS::S3Outposts::Bucket</u>	 Ja	 Ja	 Ja
<u>AWS::S3Outposts::BucketPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::S3Outposts::Endpoint</u>	 Ja	 Ja	 Ja
<u>AWS::S3Tables::Namespace</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::S3Tables::Table</u>	 Ja	 Ja	 Nein
<u>AWS::S3Tables::TableBucket</u>	 Ja	 Ja	 Nein
<u>AWS::S3Tables::TableBucketPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::S3Tables::TablePolicy</u>	 Ja	 Ja	 Nein
<u>AWS::S3Vectors::Index</u>	 Ja	 Ja	 Nein
<u>AWS::S3Vectors::VectorBucket</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::S3Vectors::VectorBucketPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::SES::ConfigurationSet</u>	 Ja	 Ja	 Nein
<u>AWS::SES::ConfigurationSetEventDesti nation</u>	 Nein	 Ja	 Nein
<u>AWS::SES::ContactList</u>	 Ja	 Ja	 Nein
<u>AWS::SES::DedicatedIpPool</u>	 Ja	 Ja	 Ja
<u>AWS::SES::EmailIdentity</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::SES::MailManagerAddonInstance</u>	 Ja	 Ja	 Nein
<u>AWS::SES::MailManagerAddonSubscription</u>	 Ja	 Ja	 Nein
<u>AWS::SES::MailManagerAddressList</u>	 Ja	 Ja	 Nein
<u>AWS::SES::MailManagerArchive</u>	 Ja	 Ja	 Nein
<u>AWS::SES::MailManagerIngressPoint</u>	 Ja	 Ja	 Nein
<u>AWS::SES::MailManagerRelay</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SES::MailManagerRuleSet</u>	 Ja	 Ja	 Nein
<u>AWS::SES::MailManagerTrafficPolicy</u>	 Ja	 Ja	 Nein
<u>AWS::SES::MultiRegionEndpoint</u>	 Ja	 Ja	 Nein
<u>AWS::SES::Template</u>	 Ja	 Ja	 Nein
<u>AWS::SES::Tenant</u>	 Ja	 Ja	 Nein
<u>AWS::SES::VdmAttributes</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SMSVOICE::ConfigurationSet</u>	 Ja	 Ja	 Nein
<u>AWS::SMSVOICE::OptOutList</u>	 Ja	 Ja	 Nein
<u>AWS::SMSVOICE::PhoneNumber</u>	 Ja	 Ja	 Nein
<u>AWS::SMSVOICE::Pool</u>	 Ja	 Ja	 Nein
<u>AWS::SMSVOICE::ProtectConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::SMSVOICE::ResourcePolicy</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::SMSVOICE::SenderId</u>	 Ja	 Ja	 Nein
<u>AWS::SNS::Subscription</u>	 Ja	 Ja	 Nein
<u>AWS::SNS::Topic</u>	 Ja	 Ja	 Ja
<u>AWS::SNS::TopicInlinePolicy</u>	 Ja	 Ja	 Nein
<u>AWS::SQS::Queue</u>	 Ja	 Ja	 Ja
<u>AWS::SQS::QueueInlinePolicy</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SSM::Association</u>	 Ja	 Ja	 Ja
<u>AWS::SSM::Document</u>	 Ja	 Nein	 Nein
<u>AWS::SSM::MaintenanceWindowTarget</u>	 Ja	 Ja	 Nein
<u>AWS::SSM::MaintenanceWindowTask</u>	 Ja	 Ja	 Nein
<u>AWS::SSM::Parameter</u>	 Ja	 Ja	 Nein
<u>AWS::SSM::PatchBaseline</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SSM::ResourceDataSync</u>	 Ja	 Ja	 Nein
<u>AWS::SSM::ResourcePolicy</u>	 Ja	 Ja	 Ja
<u>AWS::SSMContacts::Contact</u>	 Ja	 Ja	 Ja
<u>AWS::SSMContacts::ContactChannel</u>	 Ja	 Ja	 Ja
<u>AWS::SSMContacts::Plan</u>	 Ja	 Ja	 Nein
<u>AWS::SSMContacts::Rotation</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SSMGuiConnect::Preferences</u>	 Ja	 Ja	 Nein
<u>AWS::SSMIncidents::ReplicationSet</u>	 Ja	 Ja	 Ja
<u>AWS::SSMIncidents::ResponsePlan</u>	 Ja	 Ja	 Ja
<u>AWS::SSMQuickSetup::ConfigurationManager</u>	 Ja	 Ja	 Nein
<u>AWS::SSMQuickSetup::LifecycleAutomation</u>	 Ja	 Ja	 Nein
<u>AWS::SSO::Application</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SSO::ApplicationAssignment</u>	 Ja	 Ja	 Nein
<u>AWS::SSO::Assignment</u>	 Ja	 Ja	 Nein
<u>AWS::SSO::Instance</u>	 Ja	 Ja	 Nein
<u>AWS::SSO::InstanceAccessControlAttributeConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::App</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::AppImageConfig</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::SageMaker::Cluster</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::DataQualityJobDefinition</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::Device</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::DeviceFleet</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::Domain</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::Endpoint</u>	 Nein	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::SageMaker::FeatureGroup</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::Image</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::ImageVersion</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::InferenceComponent</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::InferenceExperiment</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::MlflowTrackingServer</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SageMaker::ModelBiasJobDefinition</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::ModelCard</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::ModelExplainabilityJobDefinition</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::ModelPackage</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::ModelPackageGroup</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::ModelQualityJobDefinition</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::SageMaker::MonitoringSchedule</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::PartnerApp</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::Pipeline</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::ProcessingJob</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::Project</u>	 Ja	 Ja	 Ja
<u>AWS::SageMaker::Space</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SageMaker::StudioLifecycleConfig</u>	 Ja	 Ja	 Nein
<u>AWS::SageMaker::UserProfile</u>	 Ja	 Ja	 Nein
<u>AWS::Scheduler::Schedule</u>	 Ja	 Ja	 Ja
<u>AWS::Scheduler::ScheduleGroup</u>	 Ja	 Ja	 Nein
<u>AWS::SecretsManager::ResourcePolicy</u>	 Ja	 Nein	 Nein
<u>AWS::SecretsManager::RotationSchedule</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	IaC-Generator
<u>AWS::SecretsManager::Secret</u>	 Ja	 Ja	 Ja
<u>AWS::SecretsManager::SecretTargetAttachment</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::AggregatorV2</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::AutomationRule</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::AutomationRuleV2</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::ConfigurationPolicy</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SecurityHub::ConnectorV2</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::DelegatedAdmin</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::FindingAggregator</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::Hub</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::HubV2</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::Insight</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::SecurityHub::OrganizationConfig uration</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::PolicyAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::ProductSubscriptio n</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::SecurityControl</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityHub::Standard</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityLake::AwsLogSource</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SecurityLake::DataLake</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityLake::Subscriber</u>	 Ja	 Ja	 Nein
<u>AWS::SecurityLake::SubscriberNotification</u>	 Ja	 Ja	 Nein
<u>AWS::ServiceCatalog::CloudFormationProvisionedProduct</u>	 Ja	 Nein	 Nein
<u>AWS::ServiceCatalog::LaunchNotificationConstraint</u>	 Ja	 Ja	 Nein
<u>AWS::ServiceCatalog::LaunchTemplateConstraint</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::ServiceCatalog::PortfolioPrinci palAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::ServiceCatalog::PortfolioProduc tAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::ServiceCatalog::PortfolioShare</u>	 Ja	 Ja	 Nein
<u>AWS::ServiceCatalog::ResourceUpdateC onstraint</u>	 Ja	 Ja	 Nein
<u>AWS::ServiceCatalog::ServiceAction</u>	 Ja	 Ja	 Ja
<u>AWS::ServiceCatalog::ServiceActionAs sociation</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::ServiceCatalog::TagOption</u>	 Ja	 Ja	 Nein
<u>AWS::ServiceCatalog::TagOptionAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::ServiceCatalogAppRegistry::Application</u>	 Ja	 Ja	 Ja
<u>AWS::ServiceCatalogAppRegistry::AttributeGroup</u>	 Ja	 Ja	 Ja
<u>AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::ServiceCatalogAppRegistry::ResourceAssociation</u>	 Ja	 Nein	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Shield::DRTAccess</u>	 Ja	 Ja	 Nein
<u>AWS::Shield::ProactiveEngagement</u>	 Ja	 Ja	 Nein
<u>AWS::Shield::Protection</u>	 Ja	 Ja	 Nein
<u>AWS::Shield::ProtectionGroup</u>	 Ja	 Ja	 Nein
<u>AWS::Signer::ProfilePermission</u>	 Ja	 Ja	 Nein
<u>AWS::Signer::SigningProfile</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SimSpaceWeaver::Simulation</u>	 Ja	 Ja	 Ja
<u>AWS::StepFunctions::Activity</u>	 Ja	 Ja	 Nein
<u>AWS::StepFunctions::StateMachine</u>	 Ja	 Ja	 Ja
<u>AWS::StepFunctions::StateMachineAlias</u>	 Ja	 Ja	 Ja
<u>AWS::StepFunctions::StateMachineVersion</u>	 Ja	 Ja	 Ja
<u>AWS::SupportApp::AccountAlias</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::SupportApp::SlackChannelConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::SupportApp::SlackWorkspaceConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::Synthetics::Canary</u>	 Ja	 Ja	 Ja
<u>AWS::Synthetics::Group</u>	 Ja	 Ja	 Ja
<u>AWS::SystemsManagerSAP::Application</u>	 Ja	 Ja	 Nein
<u>AWS::Timestream::Database</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::Timestream::InfluxDBInstance</u>	 Ja	 Ja	 Nein
<u>AWS::Timestream::ScheduledQuery</u>	 Ja	 Ja	 Nein
<u>AWS::Timestream::Table</u>	 Ja	 Ja	 Ja
<u>AWS::Transfer::Agreement</u>	 Ja	 Ja	 Ja
<u>AWS::Transfer::Certificate</u>	 Ja	 Ja	 Ja
<u>AWS::Transfer::Connector</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Transfer::Profile</u>	 Ja	 Ja	 Ja
<u>AWS::Transfer::Server</u>	 Ja	 Ja	 Nein
<u>AWS::Transfer::User</u>	 Ja	 Ja	 Nein
<u>AWS::Transfer::WebApp</u>	 Ja	 Ja	 Nein
<u>AWS::Transfer::Workflow</u>	 Ja	 Ja	 Ja
<u>AWS::VerifiedPermissions::IdentitySource</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::VerifiedPermissions::Policy</u>	 Ja	 Ja	 Ja
<u>AWS::VerifiedPermissions::PolicyStore</u>	 Ja	 Ja	 Ja
<u>AWS::VerifiedPermissions::PolicyTemplate</u>	 Ja	 Ja	 Ja
<u>AWS::VoiceID::Domain</u>	 Ja	 Ja	 Nein
<u>AWS::VpcLattice::AccessLogSubscription</u>	 Ja	 Ja	 Ja
<u>AWS::VpcLattice::AuthPolicy</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::VpcLattice::DomainVerification</u>	 Ja	 Ja	 Nein
<u>AWS::VpcLattice::Listener</u>	 Ja	 Ja	 Ja
<u>AWS::VpcLattice::ResourceConfiguration</u>	 Ja	 Ja	 Nein
<u>AWS::VpcLattice::ResourceGateway</u>	 Ja	 Ja	 Nein
<u>AWS::VpcLattice::ResourcePolicy</u>	 Ja	 Ja	 Nein
<u>AWS::VpcLattice::Rule</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::VpcLattice::Service</u>	 Ja	 Ja	 Ja
<u>AWS::VpcLattice::ServiceNetwork</u>	 Ja	 Ja	 Ja
<u>AWS::VpcLattice::ServiceNetworkResourceAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::VpcLattice::ServiceNetworkServiceAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::VpcLattice::ServiceNetworkVpcAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::VpcLattice::TargetGroup</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::WAFv2::IPSet</u>	 Ja	 Ja	 Ja
<u>AWS::WAFv2::LoggingConfiguration</u>	 Ja	 Ja	 Ja
<u>AWS::WAFv2::RegexPatternSet</u>	 Ja	 Ja	 Ja
<u>AWS::WAFv2::RuleGroup</u>	 Ja	 Ja	 Ja
<u>AWS::WAFv2::WebACL</u>	 Ja	 Ja	 Ja
<u>AWS::WAFv2::WebACLAssociation</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Wisdom::AIAgent</u>	 Ja	 Ja	 Nein
<u>AWS::Wisdom::AIAgentVersion</u>	 Ja	 Ja	 Nein
<u>AWS::Wisdom::AIGuardrail</u>	 Ja	 Ja	 Nein
<u>AWS::Wisdom::AIGuardrailVersion</u>	 Ja	 Ja	 Nein
<u>AWS::Wisdom::AIPrompt</u>	 Ja	 Ja	 Nein
<u>AWS::Wisdom::AIPromptVersion</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::Wisdom::Assistant</u>	 Ja	 Ja	 Ja
<u>AWS::Wisdom::AssistantAssociation</u>	 Ja	 Ja	 Ja
<u>AWS::Wisdom::KnowledgeBase</u>	 Ja	 Ja	 Ja
<u>AWS::Wisdom::MessageTemplate</u>	 Ja	 Ja	 Nein
<u>AWS::Wisdom::MessageTemplateVersion</u>	 Ja	 Ja	 Nein
<u>AWS::Wisdom::QuickResponse</u>	 Ja	 Ja	 Nein

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::WorkSpaces::ConnectionAlias</u>	 Ja	 Ja	 Nein
<u>AWS::WorkSpaces::WorkspacesPool</u>	 Ja	 Ja	 Nein
<u>AWS::WorkSpacesThinClient::Environment</u>	 Ja	 Ja	 Nein
<u>AWS::WorkSpacesWeb::BrowserSettings</u>	 Ja	 Ja	 Ja
<u>AWS::WorkSpacesWeb::DataProtectionSettings</u>	 Ja	 Ja	 Nein
<u>AWS::WorkSpacesWeb::IdentityProvider</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichun gen	laC-Gener ator
<u>AWS::WorkSpacesWeb::IpAccessSettings</u>	 Ja	 Ja	 Ja
<u>AWS::WorkSpacesWeb::NetworkSettings</u>	 Ja	 Ja	 Ja
<u>AWS::WorkSpacesWeb::Portal</u>	 Ja	 Ja	 Ja
<u>AWS::WorkSpacesWeb::SessionLogger</u>	 Ja	 Ja	 Nein
<u>AWS::WorkSpacesWeb::TrustStore</u>	 Ja	 Ja	 Ja
<u>AWS::WorkSpacesWeb::UserAccessLoggingSettings</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::WorkSpacesWeb::UserSettings</u>	 Ja	 Ja	 Ja
<u>AWS::WorkSpacesInstances::Volume</u>	 Ja	 Ja	 Nein
<u>AWS::WorkSpacesInstances::VolumeAssociation</u>	 Ja	 Ja	 Nein
<u>AWS::WorkSpacesInstances::WorkspaceInstance</u>	 Ja	 Ja	 Nein
<u>AWS::XRay::Group</u>	 Ja	 Ja	 Ja
<u>AWS::XRay::ResourcePolicy</u>	 Ja	 Ja	 Ja

Ressource	Import	Erkennung von Abweichungen	laC-Generator
<u>AWS::XRay::SamplingRule</u>	 Ja	 Ja	 Nein
<u>AWS::XRay::TransactionSearchConfig</u>	 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 Stack-Namen und die Vorlagenparameter als URL-Abfrageparameter angeben, können Sie eine einzelne Stack erstellen-Seite vorausfüllen und die Stack-Erstellung beschleunigen. Dies vereinfacht die Erstellung von Stapeln, da sowohl die Anzahl der Assistentenseiten als auch die Anzahl der erforderlichen Benutzereingaben reduziert wird. Es optimiert auch die Wiederverwendung von Vorlagen, da Sie mehrere erstellen können, URLs die unterschiedliche Werte für dieselbe Vorlage angeben.

URL-Format

Der Link für das Quick Setup 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:

Vorlage-URL

Erforderlich Der Parameter `templateURL` gibt die URL der Stack-Vorlage an, die sich in einem Amazon S3-Bucket befindet. Um Zugriffsprobleme mit einer vordefinierten S3-URL zu vermeiden, stellen Sie sicher, dass Sie die URL verschlüsseln.

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` (altes 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. Für Parameter in der Stapelvorlage, die nicht vom Typ `NoEcho` sind, verwenden Sie 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` Eigenschaften auf `true` Typen gesetzt sind (in der Regel Benutzernamen und Passwörter). 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 Stapels mit Hilfe eines Schnellzugriffs

Wenn Sie einen Link zur Schnellerstellung öffnen, werden Sie zur CloudFormation Konsole weitergeleitet. Die Konsole öffnet direkt die Seite Schnelles Erstellen von Stapeln und verwendet automatisch die angegebenen Werte für die Parameter.

So erstellen Sie einen Stapel über einen Link zum schnellen Erstellen (Konsole)

1. Bestätigen Sie auf der Seite Stapel schnell erstellen für Vorlage, Vorlagen-URL, dass die Vorlagen-URL korrekt ist.
2. Erweitern Sie den Abschnitt Vorlage ansehen um die Vorlage zu überprüfen.
3. Überprüfen Sie für Stapelnamen den vorausgefüllten Stapelnamen.

4. Lesen Sie den Abschnitt **Parameter** . Überprüfen Sie, ob die vorausgefüllten Parameterwerte korrekt sind. Füllen Sie alle erforderlichen Parameter aus, die nicht in der URL angegeben wurden. Ändern Sie bei Bedarf alle vorausgefüllten Werte.
5. Als nächstes können Sie die folgenden Einstellungen vornehmen:
 - **Tags**- Organisieren Sie Ressourcen mit Schlüssel-Werte-Paaren.
 - **Berechtigungen**- Wählen Sie die IAM-Service-Rolle für Stack-Operationen.
 - **Optionen für Stack-Fehler**- Wählen Sie, ob Sie einen Rollback durchführen (Standard) oder die Ressourcen erhalten wollen.
 - **Stackrichtlinien**- Kontrollieren Sie die Berechtigungen zur Aktualisierung von Ressourcen.
 - **Rollback-Konfiguration** — Konfigurieren Sie CloudWatch ein alarmbasiertes Rollback.
 - **Benachrichtigungsoptionen**- Richten Sie Amazon SNS-Benachrichtigungen für Stack-Ereignisse ein.
 - **Optionen für die Stackerstellung**- Definieren Sie die maximale Erstellungszeit des Stacks und aktivieren Sie den Abbruchschutz, um versehentliche Löschungen zu verhindern.

Weitere Informationen finden Sie unter [Stapeloptionen konfigurieren](#).

6. Für Fähigkeiten, füllen Sie alle erforderlichen Bestätigungen aus. Wenn Ihre Vorlage zum Beispiel IAM-Ressourcen enthält, wählen Sie **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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).
7. (Optional) Sie können einen Änderungssatz erstellen, um eine Vorschau der Konfiguration des Stapels zu erhalten, bevor Sie ihn erstellen. Wählen Sie **Änderungssatz erstellen** und folgen Sie den Anweisungen. Weitere Informationen finden Sie unter [Vorschau der Konfiguration Ihres Stacks](#).
8. Wenn Sie bereit sind, wählen Sie **Stack erstellen** um den Stack zu starten und überwachen dann den Fortschritt der Stack-Erstellung auf der Registerkarte **Ereignisse** . Weitere Informationen finden Sie unter [Fortschritt des Stapels 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 umfassen nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie unter [cloudformation](#) in der AWS CLI Befehlsreferenz.

Die Beispiele in diesem Handbuch verwenden die Konvention eines Backslash (\), um anzuzeigen, dass eine lange Befehlszeile in der nächsten Zeile fortgesetzt wird.

Themen

- [Eine Stapelaktualisierung abbrechen](#)
- [Rollback einer Aktualisierung fortsetzen](#)
- [Erstellen eines Stacks](#)
- [Erstellen Sie einen Stapel, der Transformationen enthält](#)
- [Löschen eines Stacks](#)
- [Beschreiben Sie Stack-Ereignisse](#)
- [Beschreiben Sie eine Stapelressource](#)
- [Beschreiben Sie Stack-Ressourcen](#)
- [Beschreiben Sie Stapel](#)
- [Eine Vorlage erhalten](#)
- [Stack-Ressourcen auflisten](#)
- [Stapel auflisten](#)
- [Einen Stack aktualisieren](#)
- [Validieren Sie Ihre Vorlage](#)
- [Laden Sie lokale Artefakte in einen S3-Bucket hoch mit dem AWS CLI](#)

Eine Stapelaktualisierung abbrechen

Verwenden Sie den Befehl [cancel-update-stack](#), um eine Stack-Aktualisierung abubrechen. Weitere Informationen finden Sie unter [Eine Stapelaktualisierung abbrechen](#).

CLI

AWS CLI

So brechen Sie eine laufende Stapelaktualisierung ab

Der folgende `cancel-update-stack`-Befehl bricht eine Stackaktualisierung auf dem `myteststack`-Stack ab:

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- Einzelheiten zur API finden Sie [CancelUpdateStack](#) in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Bricht eine Aktualisierung auf dem angegebenen Stapel ab.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [CancelUpdateStack AWS -Tools für PowerShell](#) Cmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Bricht eine Aktualisierung auf dem angegebenen Stapel ab.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [CancelUpdateStack AWS -Tools für PowerShell](#) Cmdlet-Referenz (V5).

Rollback einer Aktualisierung fortsetzen

Verwenden Sie den Befehl [continue-update-rollback](#), um ein Update zurückzusetzen. Weitere Informationen finden Sie unter [Rollback einer Aktualisierung fortsetzen](#).

CLI

AWS CLI

Wiederholung eines Update-Rollbacks

Das folgende `continue-update-rollback`-Beispiel setzt einen Rollback-Vorgang nach einer zuvor fehlgeschlagenen Stack-Aktualisierung fort.

```
aws cloudformation continue-update-rollback \  
  --stack-name my-stack
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- Einzelheiten zur API finden Sie unter Befehlsreferenz [ContinueUpdateRollback](#).AWS CLI

PowerShell

Tools für PowerShell V4

Beispiel 1: Setzt Rollback des benannten Stapels fort, der sich im Zustand 'UPDATE_ROLLBACK_FAILED' befinden sollte. Wenn das fortgesetzte Rollback erfolgreich ist, geht der Stack in den Zustand 'UPDATE_ROLLBACK_COMPLETE' über.

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [ContinueUpdateRollback AWS -Tools für PowerShell](#) Cmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Setzt Rollback des benannten Stapels fort, der sich im Zustand 'UPDATE_ROLLBACK_FAILED' befinden sollte. Wenn das fortgesetzte Rollback erfolgreich ist, geht der Stack in den Zustand 'UPDATE_ROLLBACK_COMPLETE' über.

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [ContinueUpdateRollback AWS -Tools für PowerShell](#) Cmdlet-Referenz (V5).

Erstellen eines Stacks

Verwenden Sie den Befehl [create-stack](#), 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, erstellt es den Stack CloudFormation nicht und meldet, dass die Vorlage diesen Parameter nicht enthält.

Die folgenden Beispiele zeigen, wie Sie einen neuen Stapel mit dem angegebenen Namen, der Vorlage und den Eingabeparametern erstellen.

CLI

AWS CLI

Um einen Stapel zu erstellen AWS CloudFormation

Der folgende Befehl `create-stack` erstellt einen Stapel mit dem Namen `myteststack` unter Verwendung der Vorlage `sampletemplate.json`:

```
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 [CreateStack](#) in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Erzeugt einen neuen Stapel 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 Stapels 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 Stapel 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 Stapels fehlschlägt, wird er zurückgerollt.

```
$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 Stapel 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 Stapel 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 PowerShell](#) Cmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Erzeugt einen neuen Stapel 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, " und PV1 '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 Stapels 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 Stapel 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 Stapels fehlschlägt, wird er zurückgerollt.

```
$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 Stapel 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 Stapel 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 PowerShell](#) Cmdlet-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. Bei Amazon S3 muss die URL mit `https://` beginnen. 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 Befehl [deploy](#), um einen Stack zu erstellen, der Transformationen enthält. Wenn Sie einen Stapel aus einer Vorlage erstellen, die Transformationen enthält, müssen Sie ein Änderungsset verwenden. Der Befehl `deploy` kombiniert zwei Schritte (Erstellung eines Änderungssatzes und dessen Ausführung) in einem einzigen Befehl.

AWS CLI

Der folgende Befehl `deploy` erstellt einen Stapel 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 Befehl [delete-stack](#), um einen Stack zu löschen. Weitere Informationen finden Sie unter [Löschen eines Stacks](#).

CLI

AWS CLI

So löschen Sie einen Stack

Das folgende Beispiel `delete-stack` löscht den angegebenen Stapel.

```
aws cloudformation delete-stack \  
  --stack-name my-stack
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- Einzelheiten zur API finden Sie [DeleteStack](#) in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Löscht den angegebenen Stapel.

```
Remove-CFNStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [DeleteStack AWS -Tools für PowerShell](#) Cmdlet-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 PowerShell](#) Cmdlet-Referenz (V5).

Wenn der Löschvorgang fehlschlägt und den Status `DELETE_FAILED` zurückgibt, können Sie den Stapel mit der Option `--deletion-mode` zwangsweise löschen. Dies sind die folgenden Werte, die mit `deletion-mode` verwendet werden können:

- **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

Der folgende `delete-stack`-Befehl erzwingt das Löschen des *myteststack*-Stacks unter Verwendung des `FORCE_DELETE_STACK`-Werts mit dem `deletion-mode`-Parameter:

```
aws cloudformation delete-stack --stack-name myteststack \  
  --deletion-mode FORCE_DELETE_STACK
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Nachdem Sie `FORCE_DELETE_STACK` verwendet haben, können Sie den Befehl `list-stack-resources` verwenden, um die Ressourcen aufzulisten, die beim Löschen des Stapels übersprungen wurden. Die zurückbehaltenen Ressourcen erhalten den Status `DELETE_SKIPPED`. Weitere Informationen finden Sie unter [Stack-Ressourcen auflisten](#).

Beschreiben Sie Stack-Ereignisse

Verwenden Sie den Befehl [describe-stack-events](#), um Stack-Ereignisse zu beschreiben. Weitere Informationen finden Sie unter [Fortschritt des Stapels überwachen](#).

CLI

AWS CLI

Zur Beschreibung von Stapelereignissen

Das folgende Beispiel `describe-stack-events` zeigt die 2 jüngsten Ereignisse für den angegebenen Stapel an.

```
aws cloudformation describe-stack-events \  
  --stack-name my-stack \  
  --max-items 2
```

```

--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": "eyJ0ZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

- Einzelheiten zur API finden Sie unter Befehlsreferenz [DescribeStackEvents](#).AWS CLI

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt alle stapelbezogenen Ereignisse für den angegebenen Stapel zurück.

```
Get-CFNStackEvent -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [DescribeStackEvents AWS -Tools für PowerShell](#) Cmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt alle stapelbezogenen Ereignisse für den angegebenen Stapel zurück.

```
Get-CFNStackEvent -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [DescribeStackEvents AWS -Tools für PowerShell](#) Cmdlet-Referenz (V5).

Beschreiben Sie eine Stapelressource

CLI

AWS CLI

So erhalten Sie Informationen zu einer Stack-Ressource

Das folgende Beispiel `describe-stack-resource` zeigt Details für die Ressource mit dem Namen `MyFunction` im angegebenen Stapel an.

```
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 unter Befehlsreferenz [DescribeStackResource](#).AWS CLI

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 PowerShell](#) Cmdlet-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 PowerShell](#) Cmdlet-Referenz (V5).

Beschreiben Sie Stack-Ressourcen

CLI

AWS CLI

So erhalten Sie Informationen zu einer Stack-Ressource

Das folgende Beispiel `describe-stack-resources` zeigt Details zu den Ressourcen im angegebenen Stapel an.

```
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-HIZXMPLE0M9E",
      "ResourceType": "AWS::IAM::Role",
      "Timestamp": "2019-10-02T04:34:06.350Z",
      "ResourceStatus": "CREATE_COMPLETE",
      "DriftInformation": {
```

```
        "StackResourceDriftStatus": "IN_SYNC"
      }
    }
  ]
}
```

- Einzelheiten zur API finden Sie unter Befehlsreferenz [DescribeStackResources](#).AWS CLI

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 Amazon EC2-Instance zurück, die in der Vorlage, die mit dem angegebenen Stack verbunden ist, durch die logische ID "Ec2Instance" identifiziert wurde.

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Beispiel 3: Gibt die Beschreibung von bis zu 100 Ressourcen zurück, die mit dem Stack verbunden sind, der eine Amazon EC2-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 Amazon EC2-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 also einer Amazon EC2-Instance mit der Instance-ID "i-123456". Je nach Inhalt der Vorlage könnte auch eine andere physische Ressource zur Identifizierung des Stacks verwendet werden, zum Beispiel ein Amazon S3-Bucket.


```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- Einzelheiten zur API finden Sie unter [DescribeStackResources AWS -Tools für PowerShell](#) Cmdlet-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 Amazon EC2-Instance zurück, die in der Vorlage, die mit dem angegebenen Stack verbunden ist, durch die logische ID "Ec2Instance" identifiziert wurde.

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Beispiel 3: Gibt die Beschreibung von bis zu 100 Ressourcen zurück, die mit dem Stack verbunden sind, der eine Amazon EC2-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 Amazon EC2-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 also einer Amazon EC2-Instance mit der Instance-ID "i-123456". Je nach Inhalt der Vorlage könnte auch eine andere physische Ressource zur Identifizierung des Stacks verwendet werden, zum Beispiel ein Amazon S3-Bucket.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- Einzelheiten zur API finden Sie unter [DescribeStackResources AWS -Tools für PowerShell](#) Cmdlet-Referenz (V5).

Beschreiben Sie Stapel

CLI

AWS CLI

Um Stacks zu beschreiben AWS CloudFormation

Der folgende describe-stacks-Befehl zeigt eine Zusammenfassung der 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-jssofilzie2w"
        }
      ],
      "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 [DescribeStacks](#) in 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 [DescribeStacks](#) in der AWS SDK für Go API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt eine Sammlung von Stack-Instances zurück, die alle Stacks des Benutzers beschreiben.

```
Get-CFNStack
```

Beispiel 2: Gibt eine Stack-Instance zurück, die den angegebenen Stack beschreibt

```
Get-CFNStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [DescribeStacks AWS -Tools für PowerShell](#) Cmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt eine Sammlung von Stack-Instances zurück, die alle Stacks des Benutzers beschreiben.

```
Get-CFNStack
```

Beispiel 2: Gibt eine Stack-Instance zurück, die den angegebenen Stack beschreibt

```
Get-CFNStack -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [DescribeStacks AWS -Tools für PowerShell](#) Cmdlet-Referenz (V5).

Der Befehl `describe-stacks` gibt standardmäßig Parameterwerte zurück. Um zu verhindern, dass sensible Parameterwerte wie Passwörter zurückgegeben werden, fügen Sie `TRUE` in Ihre Vorlagen eine `NoEcho` Eigenschaft ein, die auf `gesetzt` ist. CloudFormation

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](#).

Eine Vorlage erhalten

CLI

AWS CLI

Um den Vorlagentext für einen AWS CloudFormation Stapel anzuzeigen

Der folgende Befehl `get-template` zeigt die Vorlage für den Stapel `myteststack` :

```
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 [GetTemplate](#) in der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt die Vorlage zurück, die mit dem angegebenen Stapel verbunden ist.

```
Get-CFNTemplate -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [GetTemplate AWS -Tools für PowerShell](#) Cmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt die Vorlage zurück, die mit dem angegebenen Stapel verbunden ist.

```
Get-CFNTemplate -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [GetTemplate AWS -Tools für PowerShell](#) Cmdlet-Referenz (V5).

Stack-Ressourcen auflisten

CLI

AWS CLI

So listen Sie Ressourcen in einem Stack auf

Der folgende Befehl zeigt die Liste der Ressourcen im angegebenen Stapel 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 unter Befehlsreferenz [ListStackResources](#).AWS CLI

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt Beschreibungen aller Ressourcen zurück, die mit dem angegebenen Stapel verbunden sind.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [ListStackResources AWS -Tools für PowerShell](#) Cmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt Beschreibungen aller Ressourcen zurück, die mit dem angegebenen Stapel verbunden sind.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Einzelheiten zur API finden Sie unter [ListStackResources AWS -Tools für PowerShell](#) Cmdlet-Referenz (V5).

Stapel auflisten

Verwenden Sie den Befehl [list-stacks](#), um Stack aufzulisten. Um nur Stapel mit den angegebenen Statuscodes aufzulisten, fügen Sie die Option `--stack-status-filter` hinzu. Sie können einen oder mehrere Stack-Statuscodes für die Option `--stack-status-filter` angeben. Weitere Informationen finden Sie unter [Stack-Statuscodes](#).

CLI

AWS CLI

Um Stapel aufzulisten AWS CloudFormation

Der folgende `list-stacks`-Befehl zeigt eine Zusammenfassung aller Stack an, die den Status `CREATE_COMPLETE` haben:

```
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 [ListStacks](#) in 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 PowerShell](#) Cmdlet-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 PowerShell](#) Cmdlet-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 [Stapel direkt aktualisieren](#).

CLI

AWS CLI

Um Stacks zu aktualisieren AWS CloudFormation

Der folgende Befehl `update-stack` aktualisiert die Vorlage und die Eingabeparameter für den Stapel `mystack` :

```
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 Befehl `update-stack` aktualisiert nur den Wert des Parameters `SubnetIDs` für den Stapel `mystack` . Wenn Sie keinen Parameterwert angeben, wird der Standardwert verwendet, der in der Vorlage angegeben ist:

```
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
```

Der folgende `update-stack`-Befehl fügt dem `mystack`-Stack zwei Themen für Stackbenachrichtigungen hinzu:

```
aws cloudformation update-stack --stack-name mystack --use-previous-template --  
notification-arns "arn:aws:sns:us-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 [UpdateStack](#) in 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 [UpdateStack](#) Cmdlet-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 [UpdateStack](#) Cmdlet-Referenz (V5).AWS -Tools für PowerShell

Note

Um alle Benachrichtigungen zu entfernen, geben Sie für [] die Option `--notification-arns` an.

Validieren Sie Ihre Vorlage

Verwenden Sie den Befehl [validate-template](#), um Ihre Vorlagendatei auf Syntaxfehler zu überprüfen.

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.

CLI

AWS CLI

Um eine AWS CloudFormation Vorlage zu validieren

Der folgende Befehl `validate-template` validiert die Vorlage `sampletemplate.json` :

```
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 [ValidateTemplate](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Validiert den angegebenen Vorlageninhalt. Die Ausgabe enthält die Fähigkeiten, Beschreibung und Parameter der Vorlage.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Beispiel 2: Validiert die angegebene Vorlage, auf die über eine Amazon S3 URL zugegriffen wird. Die Ausgabe enthält die Fähigkeiten, Beschreibung und Parameter der Vorlage.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template
```

- Einzelheiten zur API finden Sie unter [ValidateTemplate AWS -Tools für PowerShell](#) Cmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Validiert den angegebenen Vorlageninhalt. Die Ausgabe enthält die Fähigkeiten, Beschreibung und Parameter der Vorlage.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```


Beispiel 2: Validiert die angegebene Vorlage, auf die über eine Amazon S3 URL zugegriffen wird. Die Ausgabe enthält die Fähigkeiten, Beschreibung und Parameter der Vorlage.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template
```

- Einzelheiten zur API finden Sie unter [ValidateTemplate AWS -Tools für PowerShell](#) Cmdlet-Referenz (V5).

Im Folgenden sehen Sie eine Beispielantwort, die einen Validierungsfehler erzeugt.

```
{
  "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 keine Sandbox oder keinen Testbereich für 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 von einer CloudFormation Vorlage verwiesen wird, 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 deren Speicherorte zu Ihrer Vorlage hinzuzufügen, können Sie lokale Artefakte in Ihrer Vorlage angeben und den Befehl [package](#) verwenden, um sie schnell hochzuladen.

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 eine Amazon API Gateway OpenAPI Gateway-REST-API-Datei sein.

Wenn Sie den Befehl package 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 dann die .zip-Datei 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 Stapel 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 [package](#) Dokumentation in der [AWS CLI Befehlsreferenz](#).

Voraussetzungen

Bevor Sie beginnen, müssen Sie über einen Amazon S3 Bucket verfügen.

Paketieren und verteilen Sie eine Vorlage mit lokalen Artefakten

Die folgende Vorlage gibt das lokale Artefakt für den Quellcode einer Lambda-Funktion an. Der Quellcode wird im Ordner `/home/user/code/lambdafunction` 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 Befehl [package](#) erstellt und lädt eine `.zip`-Datei des Quellcode-Ordners der Funktion 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 erzeugt eine neue Vorlage unter dem durch `--output-template-file` angegebenen Pfad. Es ersetzt die Artefakt-Referenz durch den Amazon S3-Speicherort, wie unten gezeigt. Die `.zip` Datei wird anhand der MD5 Prüfsumme des Ordnerinhalts benannt und nicht anhand des Ordnersnamens 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 verpackt haben, stellen Sie die verarbeitete Vorlage mit dem Befehl [deploy](#) bereit.

```
aws cloudformation deploy \
  --template-file packaged-template.json \
  --stack-name stack-name
```

Wenn Sie Vorlagen bereitstellen, die größer als 51.200 Bytes sind, verwenden Sie den Befehl `deploy` mit der Option `--s3-bucket`, um Ihre Vorlage in S3 hochzuladen, wie im folgenden Beispiel.

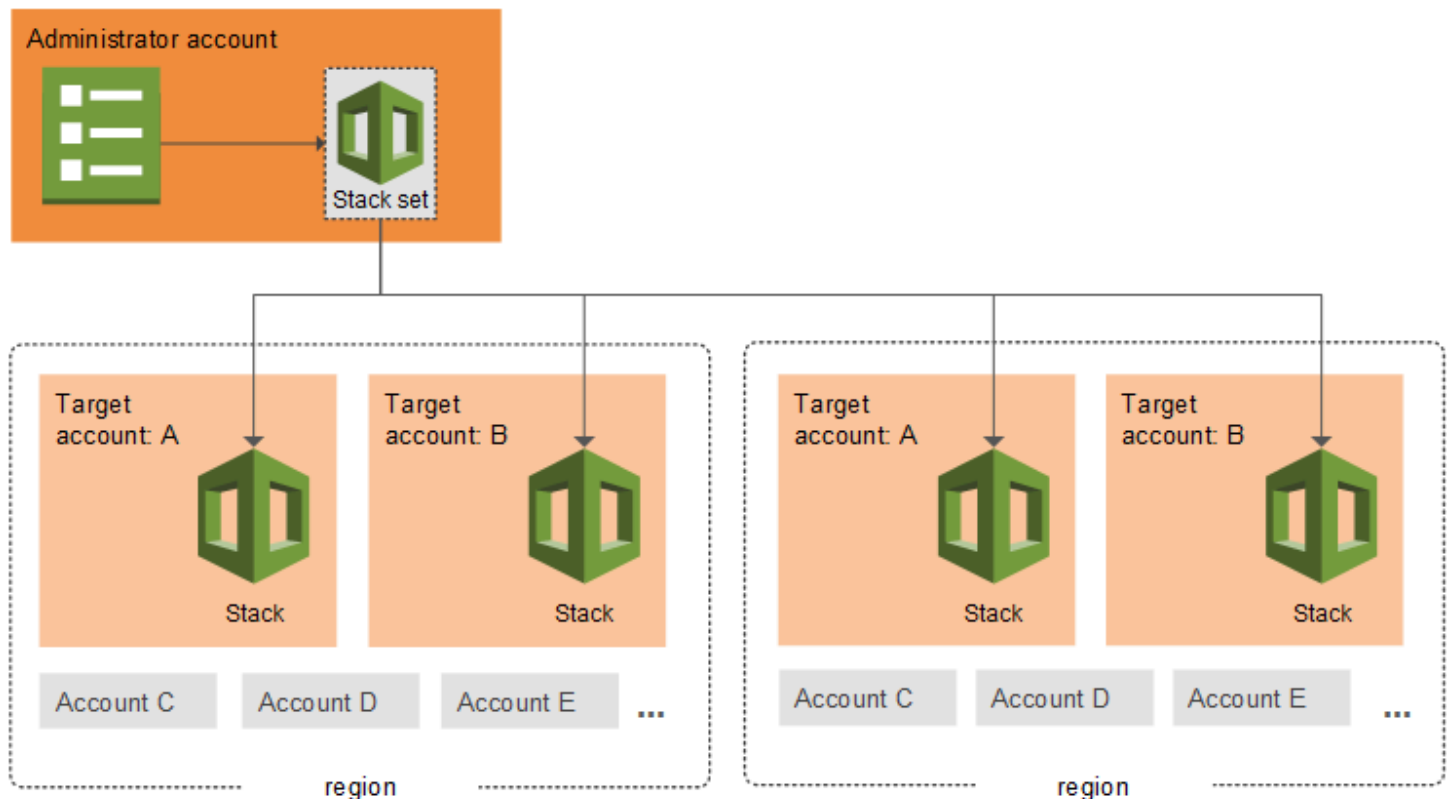
```
aws cloudformation deploy \
  --template-file packaged-template.json \
  --stack-name stack-name \
  --s3-bucket amzn-s3-demo-bucket
```

Note

Ein weiteres Beispiel für die Verwendung des Befehls `package` zum Hochladen lokaler Artefakte finden Sie unter [Aufteilung einer Vorlage in wiederverwendbare Teile mit verschachtelten Stapeln](#).

Verwaltung von Stacks über Konten und Regionen hinweg mit StackSets

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 CloudFormation StackSets](#)
- [Erste Schritte mit StackSets der Verwendung einer Beispielvorgabe](#)
- [CloudFormation StackSets Mit selbstverwalteten Berechtigungen erstellen](#)

- [CloudFormation StackSets Mit vom Service verwalteten Berechtigungen erstellen](#)
- [Automatische Bereitstellungen für StackSets in aktivieren oder deaktivieren AWS Organizations](#)
- [Aktualisierung CloudFormation StackSets](#)
- [Stapel hinzufügen zu CloudFormation StackSets](#)
- [Überschreiben Sie Parameterwerte auf Stacks in Ihrem CloudFormation StackSet](#)
- [Stapel löschen von CloudFormation StackSets](#)
- [Löschen CloudFormation StackSets](#)
- [Vermeiden Sie fehlgeschlagene StackSets Bereitstellungen mithilfe von Target Account Gates](#)
- [Wählen Sie den Parallelitätsmodus für CloudFormation StackSets](#)
- [Drifterkennung durchführen am CloudFormation StackSets](#)
- [Stapel importieren in CloudFormation StackSets](#)
- [Bewährte Methoden für die Verwendung CloudFormation StackSets](#)
- [CloudFormation StackSets Beispielvorlagen](#)
- [Problembehebung CloudFormation StackSets](#)

StackSets Konzepte

Die folgenden Begriffe und Konzepte sind für Ihr Verständnis und Ihre Verwendung von von von zentraler Bedeutung StackSets.

Themen

- [Administrator- und Ziel-Konten](#)
- [CloudFormation StackSets](#)
- [Berechtigungsmodelle für StackSets](#)
- [Stack-Instances](#)
- [StackSet Operationen](#)
- [StackSet Operationsoptionen](#)
- [Tags \(Markierungen\)](#)
- [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.

CloudFormation StackSets

A StackSet dient 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 konto- und StackSets 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 es für alle AWS-Konto bereitstellen, 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](#).

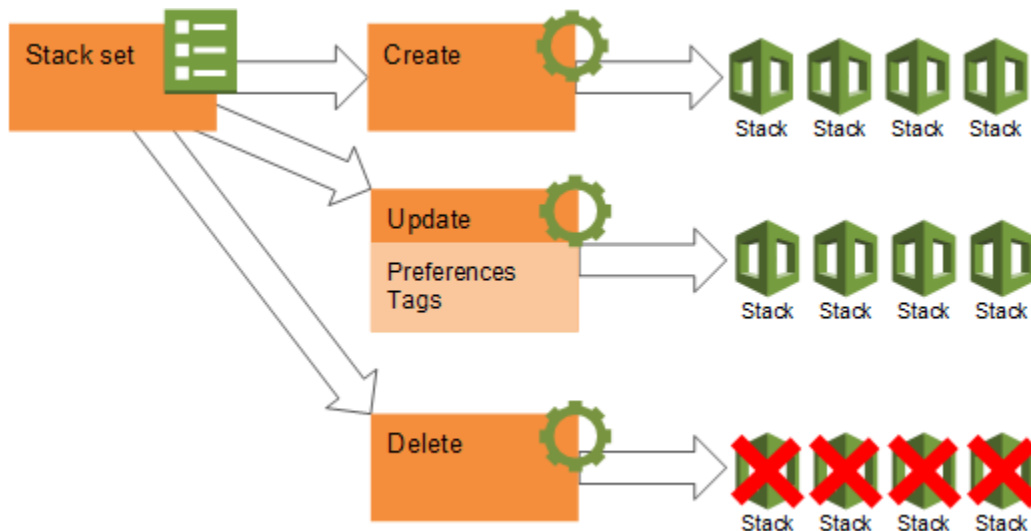
Weitere Informationen zum Erstellen und Verwalten StackSets mit vom Dienst verwalteten Berechtigungen finden Sie in den folgenden Themen:

- [Aktivieren Sie den vertrauenswürdigen Zugriff für StackSets mit AWS Organizations](#)
- [Registrieren Sie ein Mitgliedskonto für delegierte Administratoren](#)
- [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 a aktualisieren StackSet, übertragen Sie Änderungen auf 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 vorhandener Einstellungen in der Vorlage oder Hinzufügen neuer Ressourcen, z. B. zum Aktualisieren von Parametereinstellungen für einen bestimmten Service oder das Hinzufügen neuer Amazon EC2-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 Ihrem StackSet, speichern Sie sie jedoch, sodass sie unabhängig von Ihnen weiterlaufen, StackSet indem Sie die Option Stapel 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 bedienten Konten aufgrund einer Service-Ablehnung kleiner ist.

Maximale Anzahl gleichzeitiger Konten kann vom Wert von Fehlertoleranz abhängig von Ihrem Gleichzeitigkeitsmodus abhängen. Wenn Ihr Gleichzeitigkeitsmodus auf Strenge Fehlertoleranz eingestellt ist, kann Maximale Anzahl gleichzeitiger Konten höchstens eins mehr sein als die Fehlertoleranz Einstellung.

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 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 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 (Markierungen)

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 CloudFormation hinzufügen, 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 `aws:` beginnen.

StackSets Statuscodes

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. Beispiel:

- Wenn ein Konto von einer Organisationseinheit (OU) in eine andere verschoben wird, wird eine automatische Bereitstellung ausgelöst. StackSets führt einen Löschvorgang aus, um die Stack-Instance aus dem OU1 Zielkonto in der Zielregion zu entfernen, und stellt einen Erstellungsvorgang in die Warteschlange, um dem OU2 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

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 Stapel ist fehlgeschlagen.
- Der Stapel war Teil einer [CreateStackSet](#) oder [UpdateStackSet](#) Operation, die fehlgeschlagen ist oder vor der Erstellung oder Aktualisierung des Stapels angehalten wurde.

INOPERABLE

Eine [DeleteStackInstances](#) Operation ist fehlgeschlagen und hat den Stapel in einen instabilen Zustand versetzt. Stapel in diesem Zustand sind von weiteren [UpdateStackSet](#) Operationen ausgeschlossen. Möglicherweise müssen Sie eine Operation [DeleteStackInstances](#) durchführen, wobei `RetainStacks` auf `true` gesetzt wird, um die Stack-Instance zu löschen, und den Stack dann manuell 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 in das angegebene Konto und die Region ist fehlgeschlagen und hat den Stack in einen instabilen Zustand versetzt. Sobald die Probleme, die den Fehler verursacht haben, behoben sind, kann der Importvorgang erneut versucht werden. Wenn genügend StackSet Operationen auf 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 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](#).

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 unter [Aktivieren oder Deaktivieren AWS-Regionen in Ihrem Konto](#) in der AWS -Kontenverwaltung Referenzanleitung.

Um in einer StackSet Region, die standardmäßig deaktiviert StackSet ist, ein Administratorkonto (wenn Sie selbstverwaltete Berechtigungen verwenden) oder ein Verwaltungskonto der Organisation (wenn Sie vom Dienst verwaltete Berechtigungen verwenden) zu erstellen, müssen Sie zuerst diese Region für das Administrator- oder Verwaltungskonto aktivieren.

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 es für alle AWS-Konto bereitstellen, für die Sie die Berechtigung zum Erstellen einer IAM-Rolle haben.

Um dienstverwaltete Berechtigungen zu verwenden, lesen Sie stattdessen [Aktivieren Sie den vertrauenswürdigen Zugriff](#) .

Themen

- [Übersicht über selbstverwaltete Berechtigungen](#)
- [Geben Sie allen Benutzern des Administratorkontos die Berechtigung, Stapel 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.

Konkret sind die beiden erforderlichen Rollen:

- `AWSCloudFormationStackSetAdministrationRole`— Diese Rolle wird für das Administratorkonto bereitgestellt.
- `AWSCloudFormationStackSetExecutionRole`— Diese Rolle wird für alle Konten bereitgestellt, in denen Sie Stack-Instances erstellen.

Geben Sie allen Benutzern des Administratorkontos die Berechtigung, Stapel 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. Es führt Sie durch die Erstellung der erforderlichen IAM-Service-Rollen in Ihren Administrator- und Zielkonten. Jeder, der über das Administratorkonto verfügt, kann dann beliebige Stapel für alle 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 vorangegangenen Vorlage erstellte Administrationsrolle enthält die folgende Berechtigungsrichtlinie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example Beispiel Treuhandvertrag 1

Die vorangehende Vorlage enthält auch die folgende Vertrauensrichtlinie, die dem Dienst die Berechtigung zur Verwendung der Administrationsrolle und der mit der Rolle verbundenen Berechtigungen erteilt.

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

Example Beispiel Treuhandvertrag 2

Wenn Sie Stack-Instances in einem Zielkonto bereitstellen möchten, 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 gewährt dem Dienst die Berechtigung zur Verwendung der Administrationsrolle in der Region Asien-Pazifik (Hongkong) (ap-east-1), 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 Regionalcodes finden Sie unter [Regionale Endpunkte](#) in der Allgemeine AWS-Referenz Anleitung.

Target accounts

Erstellen Sie in jedem Zielkonto eine Servicerolle mit dem Namen `AWSCloudFormationStackSetExecutionRole`, dass 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, mit 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 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 Berechtigungsrichtlinie 2

Das folgende Beispiel zeigt eine Richtlinienerklärung mit den Mindestberechtigungen StackSets , um zu funktionieren. 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 Treuhand-Police

Die folgende Vertrauensbeziehung wird durch die Vorlage erstellt. Die ID des Administratorkontos wird als *admin_account_id* angezeigt.

```
{
```

```
"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, wie *admin_account_id* im vorherigen Beispiel dargestellt.

Richten Sie erweiterte Berechtigungsoptionen für StackSet Operationen ein

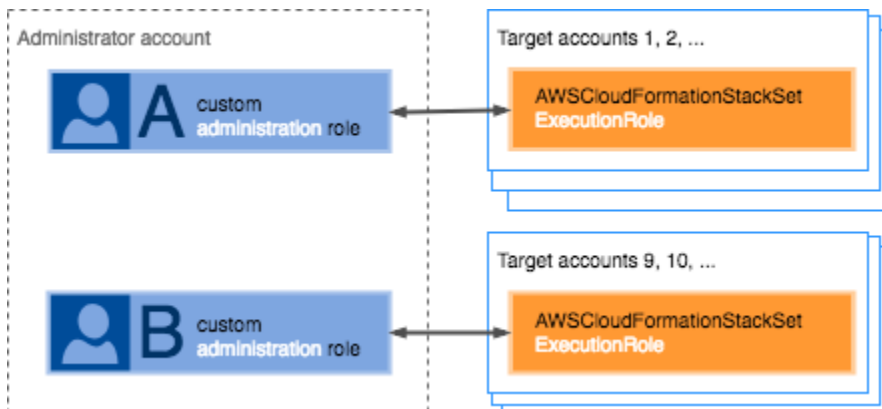
Wenn Sie eine genauere Kontrolle darüber benötigen StackSets, was Benutzer und Gruppen über ein einziges Administratorkonto erstellen, können Sie mithilfe von IAM-Rollen Folgendes angeben:

- 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 `AWSCloudFormationStackSetAdministrationRoleService`rolle 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 angegebenen 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. CloudFormation verwendet 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 des Zielkontos muss in jedem Zielkonto derselbe sein. Wenn der Rollename lautet `AWSCloudFormationStackSetExecutionRole`, wird er automatisch beim Erstellen einer StackSets verwendet StackSet. Wenn Sie einen benutzerdefinierten Rollennamen

angeben, müssen Benutzer den Namen der Ausführungsrolle beim Erstellen einer `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"
    }
  ]
}
```

Wenn Sie mehrere Konten in einer einzigen Anweisung angeben möchten, trennen Sie diese 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 einen Platzhalter (*) anstelle einer Konto-ID verwenden.

```
"Resource": [
  "arn:aws:iam::*:role/custom_execution_role"
]
```

Example Beispiel Treuhandvertrag 1

Sie müssen eine Vertrauensrichtlinie für die Dienstrolle erstellen, um festzulegen, welche IAM-Principals die Rolle übernehmen können.

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

Example Beispiel Treuhandvertrag 2

Wenn Sie Stack-Instances in einem Zielkonto bereitstellen möchten, 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 gewährt dem Dienst die Berechtigung zur Verwendung der Administrationsrolle in der Region Asien-Pazifik (Hongkong) (ap-east-1), 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 Regionalcodes finden Sie unter [Regionale Endpunkte](#) im AWS Allgemeinen Referenzhandbuch.

Example Beispiel einer Pass-Rollen-Richtlinie

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 Service-Rolle, die der benutzerdefinierten Administrationsrolle vertraut, die Sie mit diesem Konto verwenden möchten.

Für die Rolle 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 in S3. Ihr Zielkonto benötigt immer volle CloudFormation Berechtigungen, zu denen auch Berechtigungen zum Erstellen, Aktualisieren, Löschen und Beschreiben von Stacks gehören.

Der Rollename des Zielkontos muss in jedem Zielkonto derselbe sein. Wenn der Rollename lautet `AWSCloudFormationStackSetExecutionRole`, wird er automatisch beim Erstellen eines StackSets StackSet verwendet. 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 StackSets , um zu funktionieren. 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 Treuhand-Police

Sie müssen die folgende Vertrauensrichtlinie angeben, wenn Sie die Rolle erstellen, um die Vertrauensbeziehung zu definieren.

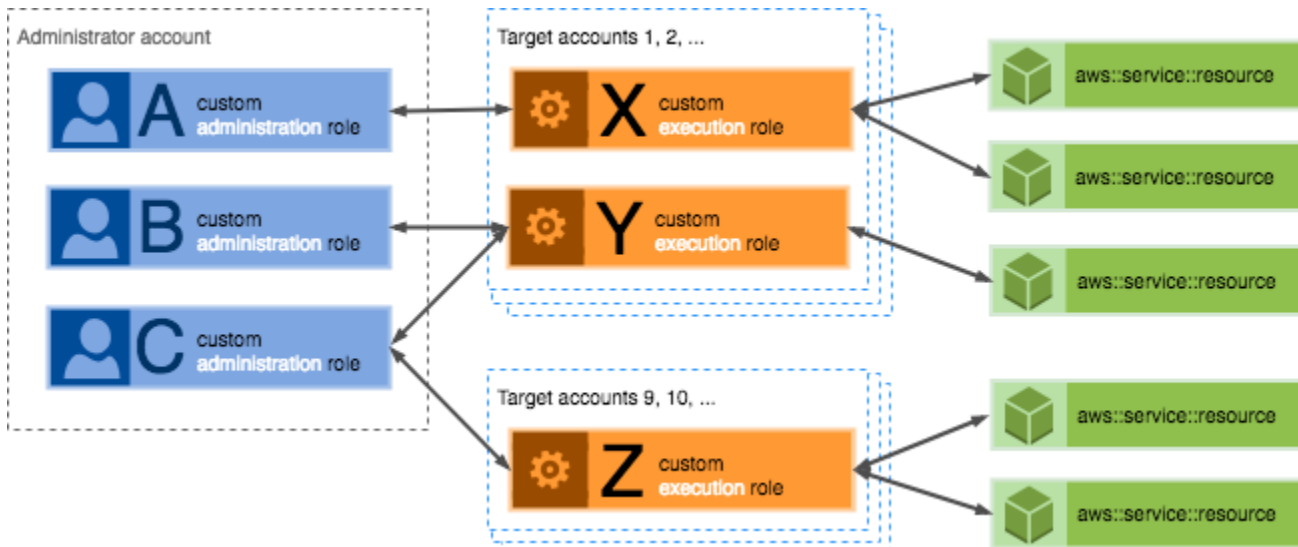
```
{
  "Version": "2012-10-17",
```

```
"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. Dazu erstellen Sie eine Vertrauensbeziehung zwischen der angepassten Administrationsrolle für jede Gruppe und einer angepassten Ausführungsrolle für jede Gruppe von Ressourcen. 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 zum Beispiel benutzerdefinierte Administrationsrollen 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.



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. 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 angepasste Administrationsrolle in Ihrem Administratorkonto, wie in [Steuern Sie, welche Benutzer StackSet Operationen in bestimmten Zielkonten ausführen können](#) beschrieben. Fügen Sie eine Vertrauensbeziehung zwischen der benutzerdefinierten Administrationsrolle und den benutzerdefinierten Ausführungsrollen ein, 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 Treuhand-Police

Sie müssen die folgende Vertrauensrichtlinie angeben, wenn Sie die Rolle erstellen, um die Vertrauensbeziehung zu definieren.

```
{
  "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 Operationen ein StackSet

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:SourceArn](#) und die [aws:SourceAccount](#) globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, 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 `AWSCloudFormationStackSetAdministrationRole` Vertrauensrichtlinie, 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"
      },
      "ArnLike": {
        "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 kontenübergreifende Bereitstellung und StackSets die AWS-Regionen Verwendung von dienstverwalteten Berechtigungen erforderlich. Um selbstverwaltete Berechtigungen zu verwenden, lesen Sie stattdessen [Selbstverwaltete 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.

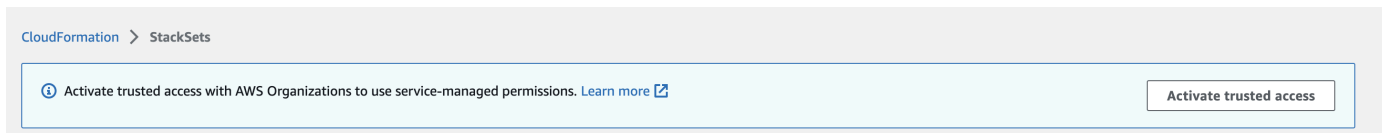
Wenn der vertrauenswürdige Zugriff aktiviert ist, können das Verwaltungskonto und die delegierten Administratorkonten dienstverwaltete StackSets Konten für ihre Organisation erstellen und verwalten.

Um den vertrauenswürdigen Zugriff zu aktivieren, müssen Sie ein Administrator-Benutzer im Verwaltungskonto sein. Ein Administrator-Benutzer ist ein Benutzer mit vollen Rechten für Ihr AWS-Konto. Weitere Informationen finden Sie in [Erstellen eines Administratorbenutzers](#) im

AWS -Kontenverwaltung Referenzhandbuch. Empfehlungen zum Schutz der Sicherheit des Verwaltungskontos finden Sie unter [Best Practices für das Verwaltungskonto](#) im AWS Organizations - Benutzerhandbuch.

So aktivieren Sie den vertrauenswürdigen Zugriff

1. Melden Sie sich AWS als Administrator des Verwaltungskontos an und öffnen Sie die CloudFormation Konsole unter. <https://console.aws.amazon.com/cloudformation>
2. StackSets. Wenn der vertrauenswürdige Zugriff deaktiviert ist, wird ein Banner mit einem Prompt 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 [CloudFormation StackSets und AWS Organizations](#) im AWS Organizations Benutzerhandbuch.

Bevor Sie den vertrauenswürdigen Zugriff mit deaktivieren können AWS Organizations, müssen Sie alle delegierten Administratoren abmelden. Weitere Informationen finden Sie unter [Einen delegierten Administrator registrieren](#).

Note

Informationen über die Verwendung von API-Vorgängen anstelle der Konsole, um den vertrauenswürdigen Zugriff zu aktivieren oder zu deaktivieren, finden Sie unter:

- [ActivateOrganizationsAccess](#)
- [DeactivateOrganizationsAccess](#)
- [DescribeOrganizationsAccess](#)

Service-verknüpfte Rollen

Das Verwaltungskonto verwendet die dienstverknüpfte Rolle.

`AWSServiceRoleForCloudFormationStackSetsOrgAdmin` 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 [CloudFormation StackSets und AWS Organizations](#) im AWS Organizations Benutzerhandbuch.

Registrieren Sie ein Mitgliedskonto für delegierte Administratoren

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 die folgenden Hinweise:

- Delegierte Administratoren haben volle Berechtigungen für die Bereitstellung auf Konten Ihrer Organisation. Das Verwaltungskonto kann delegierte Administratorrechte nicht auf die Bereitstellung auf bestimmte Vorgänge OUs oder die Ausführung bestimmter StackSet Vorgänge beschränken.
- Vergewissern Sie sich, dass Ihre delegierten Administratoren über `organizations:ListDelegatedAdministrators`-Berechtigungen 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 [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 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 CloudFormation Konsole unter <https://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 Verwendung von AWS-Managementkonsole. 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) (`us-east-1`) 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 .yaml>.

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 Vorlage StackSet mit einer Beispielvorlage aus der Konsole](#)
- [Überwachen Sie StackSet die Erstellung](#)
- [StackSet Ergebnisse anzeigen](#)

- [Aktualisieren Sie Ihre StackSet](#)
- [Füge Stapel zu deinem hinzu StackSet](#)
- [Bereinigen](#)
- [Nächste Schritte](#)

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, sollten Sie sicherstellen, 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 unter [Selbstverwaltete Berechtigungen erteilen](#).

Erstellen Sie eine Vorlage 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 Berechtigundie Option Self-Service-Berechtigungen und wählen Sie die IAM-Rollen, 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 Wählen Sie eine Mustervorlage die Vorlage Aktivieren AWS Config . 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 unterstützenden Standardwert bei, true, um alle unterstützten Ressourcentypen zu erfassen.
 - b. Behalten Sie für Globale Ressourcentypen einschließenden Standardwert bei, false, um globale Ressourcen wie IAM-Rollen auszuschließen.
 - c. Lassen Sie Liste der Ressourcentypen, wenn nicht alle unterstützt werden auf <Alle> gesetzt.
 - d. Ersetzen Sie < DeployToAnyRegion > für Die Region, die die mit dem Dienst Config verknüpfte Rollenressource enthält, durch. **us-west-2**

Das bedeutet, dass die servicegebundene Rolle mit dem Namen `AWSServiceRoleForConfig` nur erstellt wird, wenn ein Stack in der Region US West (Oregon) eingesetzt wird. Sie werden die Einsatzregionen später in diesem Verfahren auswählen.

- e. Für Konfiguration Rekorder Aufzeichnungshäufigkeit, wählen Sie TÄGLICH Aufzeichnung.
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- und Wertepaar 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:
 - Nicht konkurrierende Operationen werden gleichzeitig ausgeführt, um die Einsatzzeiten zu verkürzen.
 - Sich widersprechende Vorgänge werden automatisch in eine Warteschlange gestellt und in der Reihenfolge bearbeitet, in der sie angefordert wurden.

Während der Ausführung von Vorgängen oder in der Warteschlange stehen, werden alle eingehenden Vorgänge in CloudFormation die 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 Option Neue Stacks bereitstellen StackSet 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 für Regionen angegebendie folgenden Regionen in dieser Reihenfolge:
 1. Region US West (Oregon) (us-west-2)
 2. Region USA Ost (Nord-Virginia) (us-east-1)

Verwenden Sie den Pfeil nach oben neben der Region US West (Oregon), um sie bei Bedarf an den ersten Eintrag in der Liste zu verschieben. Die Reihenfolge der Regionen bestimmt ihre Einsatzreihenfolge.

19. Konfigurieren Sie für Einsatzoptionendie 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. Für Regionale Gleichzeitigkeit wählen Sie Sequentiell (Standard).

Diese Einstellung stellt sicher, dass Bereitstellungen in einer Region CloudFormation abgeschlossen werden, bevor zur nächsten übergegangen wird.

- d. Behalten Sie für Gleichzeitigkeitsmodus die Standardeinstellung Strenge Fehlertoleranz bei.

Bei Implementierungen mit mehreren Konten wird dadurch die Gleichzeitigkeit der Konten reduziert, wenn es zu Fehlern kommt, wobei die Fehlertoleranz +1 beibehalten wird.

20. Wählen Sie Weiter aus.

21. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie Bearbeiten in dem entsprechenden Abschnitt.

22. Wenn Sie bereit sind, Ihre zu erstellen StackSet, wählen Sie Submit.

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 Vorgangstatus sollte RUNNING anfänglich sein. CloudFormation erstellt Stapel in den Regionen, die Sie entsprechend den von Ihnen konfigurierten Bereitstellungsoptionen angegeben haben.
3. Um weitere Details zu dem Vorgang zu sehen, wählen Sie die ID des Vorgangs 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 ändert. Dies bedeutet SUCCEEDED, 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 SUCCEEDED haben, was bedeutet, dass sie erfolgreich bereitgestellt wurde.
3. Um zu überprüfen, ob dies in Ihrem Konto aktiviert ist, können Sie die AWS Config Konsole in jeder der bereitgestellten Regionen überprüfen.

Aktualisieren Sie Ihre StackSet

Nachdem Sie Ihre erstellte StackSet, möchten Sie sie möglicherweise aktualisieren, um Parameterwerte zu ändern oder weitere Regionen hinzuzufügen. In diesem Abschnitt erfahren Sie, wie Sie den Parameter für die AWS Config Aufnahme Frequenz aktualisieren.

Um die zu aktualisieren StackSet

1. Wählen Sie auf der StackSets-Seite Ihre **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 vorbereitendie Option Aktuelle Vorlage verwenden.
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. Für Distributionsoptionenbehalten Sie die gleichen Einstellungen wie zuvor.
10. Wählen Sie Weiter aus.
11. Auf der Seite Überprüfung überprüfen Sie Ihre Änderungen und wählen Absenden.
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. Dieser Abschnitt zeigt Ihnen, wie Sie einer neuen Region Stapel hinzufügen.

Um Stacks zu deinem hinzuzufügen StackSet

1. Wählen Sie auf der StackSetsSeite Ihre **my-awsconfig-stackset** aus.
2. Wenn Sie das StackSet ausgewählt haben, wählen Sie im Menü Aktionen die Option Stapel hinzufügen zu StackSet.
3. Wählen Sie auf der Seite „Bereitstellungsoptionen festlegen“ für Stacks hinzufügen zu die StackSet Option Neue Stacks bereitstellen aus.
4. Wählen Sie für Kontendie Option Stapel in Konten bereitstellen und geben Sie Ihre Konto-ID ein.
5. Wählen Sie für Regionen angebeneine neue Region aus, z. B. Europa (Irland) (eu-west-1).
6. Für Einsatzoptionenbehalten Sie die gleichen Einstellungen wie zuvor.
7. Wählen Sie Weiter aus.

8. Lassen Sie auf der Seite Überschreibungen festlegen die Eigenschaftswerte wie angegeben und wählen Sie 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 innerhalb Ihres Kontos bereitgestellt werden, ist die Bereinigung ganz einfach.

Um Stacks von Ihrem 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 Kontendie Option Stacks in Konten bereitstellen und geben Sie Ihre Konto-ID ein.
4. Wählen Sie für Regionen angebenalle Regionen aus, in denen Sie Stapel bereitgestellt haben.
5. Für Distributionsoptionenbehalten Sie die Standardeinstellungen bei.
6. Stellen Sie sicher, dass Stack beibehalten nicht aktiviert ist, damit die Stack 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 Stapel aus Ihrem. StackSet Sie können den Fortschritt auf der Registerkarte Operationen auf 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.

So löschen Sie die IAM-Dienstrollen

Da Sie die Bereitstellung nur für Ihr Konto vorgenommen haben, müssen Sie nur die IAM-Rollen dieses einzigen Kontos löschen, was die Bereinigung wesentlich einfacher macht als bei der Bereitstellung für mehrere Konten.

1. Öffnen Sie die [IAM-Konsole](#).
2. Wählen Sie im Navigationsbereich Roles (Rollen) aus.
3. Geben Sie in das Suchfeld **AWSCloudFormationStackSet** ein, um die Rollen zu finden, die Sie für dieses Lernprogramm erstellt haben.
4. Markieren Sie das Kontrollkästchen neben **AWSCloudFormationStackSetAdministrationRole**.
5. Wählen Sie oben auf der Seite Löschen aus.
6. Geben Sie im Bestätigungsdialogfeld **delete** ein und wählen Sie Löschen.
7. Wiederholen Sie den gleichen Vorgang, um die 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 sicher manuell löschen. Bevor Sie einen Bucket löschen können, müssen Sie ihn zunächst leeren. Wenn Sie einen Bucket leeren, werden alle darin enthaltenen Objekte gelöscht.

So leeren und löschen Sie die Amazon S3-Buckets

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 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 und wählen Sie dann Löschen.
8. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie den Namen des Buckets ein und wählen Sie Bucket löschen.
9. Überwachen Sie in der Liste Buckets den Fortschritt des Löschvorgangs für den Bucket. Wenn Amazon S3 die Löschung des Buckets abgeschlossen hat, entfernt es den Bucket aus der Liste.
10. Wiederholen Sie diesen Vorgang für jeden Bucket, den Sie StackSet in den verschiedenen Regionen erstellt haben.

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 außer Kraft setzen können.
- [CloudFormation StackSets Mit vom Service verwalteten Berechtigungen erstellen](#)— Erfahren Sie mehr über StackSets das Erstellen von Bereitstellungen mit mehreren Konten mit. AWS Organizations

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-Managementkonsole und öffnen Sie die 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 StackSets auf der Seite die Option Erstellen aus StackSet.
5. Wählen Sie unter Berechtigungen die Option Selbstbedienungsberechtigungen und wählen Sie die von Ihnen erstellten IAM-Rollen.
6. Wählen Sie unter Voraussetzung- Vorlage vorbereiten, Vorlage ist bereit.
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 wählen Sie dann Weiter aus.
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:
- Nicht konkurrierende Operationen werden gleichzeitig ausgeführt, um die Einsatzzeiten zu verkürzen.
 - Sich widersprechende Vorgänge werden automatisch in eine Warteschlange gestellt und in der Reihenfolge bearbeitet, in der sie angefordert wurden.

Während der Ausführung von Vorgängen oder in der Warteschlange stehen, werden alle eingehenden Vorgänge in CloudFormation die 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 für Fähigkeitendie 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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).
13. Wählen Sie Weiter aus.
14. Wählen Sie auf der Seite Bereitstellungsoptionen festlegen für Stacks hinzufügen zu die Option Neue Stacks bereitstellen StackSet 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 die ID Ihres Administratorkontos angeben, wenn Sie Stacks auch unter diesem Konto bereitstellen möchten.

16. Wählen Sie unter Regionen angegebendie Regionen aus, in denen Sie Stacks einsetzen möchten.
17. Für Einsatzoptionengehen Sie wie folgt vor:
- Geben Sie für Maximale Anzahl gleichzeitiger Kontenan, wie viele Konten gleichzeitig bearbeitet werden.

- Geben Sie für Fehlertoleranz die maximal zulässige Anzahl von Kontoausfällen pro Region an. Die Operation wird gestoppt und nicht mit anderen Regionen fortgesetzt, sobald dieses Limit erreicht ist.
 - Für Gleichzeitigkeit der Regionen wählen Sie, wie die Regionen verarbeitet werden sollen: Sequentiell (eine Region auf einmal) oder Parallel (mehrere Regionen gleichzeitig).
 - Wählen Sie für Gleichzeitigkeitsmodus, wie sich die Gleichzeitigkeit während der Ausführung der Operation verhält.
 - Strenge Fehlertoleranz- Reduziert den Gleichzeitigkeitsgrad des Kontos, wenn Fehler auftreten, und bleibt dabei innerhalb der Fehlertoleranz +1.
 - Weiche Fehlertoleranz- Behält die von Ihnen angegebene Gleichzeitigkeitsstufe (den Wert von Maximale gleichzeitige Konten) unabhängig von Fehlern bei.
18. Wählen Sie Weiter aus.
19. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie Bearbeiten in dem entsprechenden Abschnitt.
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-Instances bereit.

Um einen zu erstellen StackSet

1. Verwenden Sie den [create-stack-set](#) Befehl, 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 ein *KeyPairName* mit dem Wert *TestKey* festlegt.

```
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 den [list-stack-sets](#) Befehl aus, um zu überprüfen, ob Ihr Befehl erstellt StackSet wurde. Sie sollten Ihr neues StackSet in den Ergebnissen sehen.

```
aws cloudformation list-stack-sets
```

3. Verwenden Sie den [create-stack-instances](#) Befehl, um Stapel in Ihrem StackSet bereitzustellen. Im folgenden Beispiel werden Stapel in zwei AWS-Konten (*account_ID_1* und *account_ID_2*) in zwei Regionen (*us-west-2* und *us-east-1*) eingesetzt.

Legen Sie mit der Option `--operation-preferences` die Verarbeitung von Gleichzeitigkeitskonten und andere Bereitstellungseinstellungen fest. Dieses Beispiel verwendet zählbasierte Einstellungen. Beachten Sie, dass `MaxConcurrentCount` nicht größer sein darf als `FailureToleranceCount + 1`. Für prozentuale Einstellungen verwenden Sie 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 Befehl [describe-stack-set-operation](#) um zu überprüfen, ob Ihre Stapel erfolgreich erstellt wurden. Für die Option `--operation-id` geben Sie die Operations-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
```

CloudFormation StackSets Mit vom Service verwalteten Berechtigungen erstellen

Mit vom Dienst verwalteten Berechtigungen können Sie StackSets für Konten bereitstellen, die von AWS Organizations in bestimmten Regionen verwaltet werden. Bei diesem Modell müssen Sie nicht in jedem Zielkonto und IAM-Rollen erstellen. AWS-Region 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 Dienst 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 können entweder über das Verwaltungskonto Ihrer Organisation oder über delegierte Administratorkonten initiiert werden, aber alle Vorgänge werden über das Verwaltungskonto ausgeführt.
- CloudFormation verteilt keine StackSets für das Verwaltungskonto, auch wenn dieses Konto Teil Ihrer Organisation oder einer Organisationseinheit (OU) ist.
- Sie StackSet können Ihre gesamte Organisation (einschließlich aller Konten) oder bestimmte Ziele als Ziel angeben OUs. Wenn ein StackSet Ziel auf eine übergeordnete Organisationseinheit abzielt, schließt es automatisch jedes Kind mit ein OUs. Wenn es sich um ein bestimmtes StackSet Ziel handelt OUs, schließt es standardmäßig alle Konten innerhalb dieser Konten ein OUs. Sie können jedoch mit Hilfe von Konto-Filteroptionen bestimmte Konten anvisieren.
- Mehrere StackSets können auf dieselbe Organisation oder Organisationseinheit abzielen.
- Sie können keine Konten außerhalb Ihres Unternehmens anvisieren.
- Ihre Bereitstellungsautorisierung StackSets hängt von den Berechtigungen ab, die dem IAM-Prinzipal (Benutzer, Rolle oder Gruppe) zugewiesen wurden, mit dem Sie sich beim Verwaltungskonto anmelden. Eine IAM-Beispielrichtlinie, die Berechtigungen für die Bereitstellung in einer Organisation erteilt, finden Sie unter [Einschränkung von Stack-Set-Operationen basierend auf Region und Ressourcentypen](#).

- Delegierte Administratoren haben die volle Berechtigung zur Bereitstellung für jedes Konto in Ihrem Unternehmen. Das Verwaltungskonto kann delegierte Administratorberechtigungen nicht auf die Bereitstellung auf bestimmte OUs Operationen beschränken. StackSet
- Die Einstellungen für die automatische Bereitstellung gelten auf der entsprechenden StackSet Ebene. Sie können automatische Bereitstellungen nicht selektiv für OUs Konten oder Regionen anpassen.
- StackSets die vom Dienst verwaltete Berechtigungen verwenden, unterstützen keine verschachtelten Stacks oder Vorlagen, die Makros oder Transformationen enthalten.

Erstellen Sie eine StackSet mit vom Dienst verwalteten Berechtigungen (Konsole)

Um eine zu erstellen StackSet

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die 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 StackSets auf 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, Vorlage ist bereit.
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 den 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 AWSAWS Fakturierung und Kostenmanagement Benutzerhandbuch unter [Organisieren und Nachverfolgen von AWS Kosten mithilfe von Kostenzuordnungs-Tags](#).
10. Wählen Sie für die Ausführungskonfiguration die Option Aktiv aus, um CloudFormation die optimierte Betriebsbehandlung zu aktivieren:
 - Nicht konkurrierende Operationen werden gleichzeitig ausgeführt, um die Einsatzzeiten zu verkürzen.
 - Sich widersprechende Vorgänge werden automatisch in eine Warteschlange gestellt und in der Reihenfolge bearbeitet, in der sie angefordert wurden.

Während der Ausführung von Vorgängen oder in der Warteschlange stehen, werden alle eingehenden Vorgänge in CloudFormation die 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 für Fähigkeitendie 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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).
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 Bereitstellungszieleeinen der folgenden Schritte aus:
 - Um alle Konten in Ihrer Organisation zu verteilen, wählen Sie Verteilen an Organisation.
 - Um die Bereitstellung für alle bestimmten Konten durchzuführen OUs, wählen Sie Für Organisationseinheiten bereitstellen () aus. OUs 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 angegebendie Regionen aus, in denen Sie Stacks einsetzen möchten.
17. Für Einsatzoptionengehen Sie wie folgt vor:
- Geben Sie für Maximale Anzahl gleichzeitiger Kontenan, wie viele Konten gleichzeitig bearbeitet werden.
 - Geben Sie für Fehlertoleranzdie maximal zulässige Anzahl von Kontoausfällen pro Region an. Die Operation wird gestoppt und nicht mit anderen Regionen fortgesetzt, sobald dieses Limit erreicht ist.

- Für Gleichzeitigkeit der Regionen wählen Sie, wie die Regionen verarbeitet werden sollen: Sequentiell (eine Region auf einmal) oder Parallel (mehrere Regionen gleichzeitig).
- Wählen Sie für Gleichzeitigkeitsmodus, wie sich die Gleichzeitigkeit während der Ausführung der Operation verhält.
 - Strenge Fehlertoleranz- Reduziert den Gleichzeitigkeitsgrad des Kontos, wenn Fehler auftreten, und bleibt dabei innerhalb der Fehlertoleranz +1.
 - Weiche Fehlertoleranz- Behält die von Ihnen angegebene Gleichzeitigkeitsstufe (den Wert von Maximale gleichzeitige Konten) unabhängig von Fehlern bei.
- Fügen Sie für StackSet Abhängigkeiten abhängige hinzu StackSet ARNs und bleiben Sie dabei innerhalb von maximal 10 Abhängigkeiten. Weitere Informationen finden Sie unter [Automatische Bereitstellungen für StackSets in aktivieren oder deaktivieren AWS Organizations](#).

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 Create aus 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-Instances bereit.

Note

Wenn Sie als beauftragter Administrator handeln, müssen Sie `--call-as DELEGATED_ADMIN` in den Befehl einfügen.

Deploy to your organization

Um einen zu erstellen StackSet

1. Verwenden Sie den [create-stack-set](#) Befehl, um einen neuen StackSet Namen zu erstellen *my-stackset*. Das folgende Beispiel verwendet eine Vorlage, die in einem S3-Bucket gespeichert ist, aktiviert 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,DependsOn=ARN1,ARN2
```

2. Verwenden Sie den [list-stack-sets](#) Befehl, 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-instances](#) Befehl, um Stapel zu Ihrem StackSet hinzuzufügen. Für die Option `--deployment-targets` geben Sie die Stamm-ID der Organisation an, um sie für alle Konten in Ihrer Organisation bereitzustellen.

Legen Sie mit der Option `--operation-preferences` die Verarbeitung von Gleichzeitigkeitskonten und andere Bereitstellungseinstellungen fest. Dieses Beispiel

verwendet zählbasierte Einstellungen. Beachten Sie, dass `MaxConcurrentCount` nicht größer sein darf als `FailureToleranceCount + 1`. Für prozentuale Einstellungen verwenden Sie 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 `operation-id`-Befehl, der als Teil der `create-stack-instances`-Ausgabe zurückgegeben wurde, und überprüfen Sie mit dem folgenden [describe-stack-set-operation](#)-Befehl, ob Ihre Stack erfolgreich erstellt wurden.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to organizational units (OUs)

Um ein zu erstellen StackSet

1. Verwenden Sie den [create-stack-set](#)-Befehl, 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 ein *KeyPairName* mit dem Wert *TestKey* festlegt

```
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-sets](#)-Befehl, um zu bestätigen, dass Ihr erstelltes 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-instances](#) Befehl, 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 mit der Option `--operation-preferences` die Verarbeitung von Gleichzeitigkeitskonten und andere Bereitstellungseinstellungen fest. Dieses Beispiel verwendet zählbasierte Einstellungen. Beachten Sie, dass `MaxConcurrentCount` nicht größer sein darf als `FailureToleranceCount + 1`. Für prozentuale Einstellungen verwenden Sie 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 `operation-id`-Befehl, der als Teil der `create-stack-instances`-Ausgabe zurückgegeben wurde, und überprüfen Sie mit dem folgenden [describe-stack-set-operation](#)-Befehl, ob Ihre Stack 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.

Bestimmte Konten in einer OU anvisieren

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
```

Konten aus einer OU ausschließen

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 automatische Bereitstellungen funktionieren

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

Nehmen wir zum Beispiel StackSet1, das auf OU1 in der us-east-1-Region zielt, und StackSet2, das auf OU2 in der us-east-1-Region zielt. OU1 enthält AccountA.

Wenn Sie AccountA von OU1 zu wechseln und automatische Bereitstellungen aktiviert sind, CloudFormation wird automatisch ein Löschvorgang ausgeführt, um den StackSet1 Stapel zu entfernen, AccountA und stellt einen Erstellungsvorgang in die Warteschlange, OU2 mit dem der Stapel hinzugefügt wird StackSet2. AccountA

Überlegungen

Bei der Verwendung von automatischen Distributionen sind folgende Punkte zu beachten:

- Die Funktion für automatische Bereitstellungen ist auf der Ebene aktiviert. StackSet 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.
- Bei automatischen Bereitstellungen werden Targeting-Filter auf Kontoebene nicht berücksichtigt. Wenn Sie auf bestimmte Konten abzielen und automatische Bereitstellungen aktivieren, erfolgt die Bereitstellung StackSet weiterhin für alle neu hinzugefügten Konten innerhalb der bereitgestellten Organisation. Um die Bereitstellung für neu hinzugefügte Konten zu verhindern, deaktivieren Sie die automatische Bereitstellung.
- Abhängigkeitsverwaltung: Definieren Sie bis zu 10 Abhängigkeiten pro StackSet Konto mit bis zu 100 Abhängigkeiten pro Konto. Wenn Sie beispielsweise fünf StackSets mit jeweils fünf Abhängigkeiten haben, haben Sie 25 Abhängigkeiten, die auf das Limit von 100 Abhängigkeiten angerechnet werden. Sie können eine Erhöhung des Limits über die [Service Quota Console](#) beantragen. Abhängigkeiten werden entfernt, wenn sie StackSets gelöscht oder Organizations deaktiviert werden.

- Es wird empfohlen, die verwaltete Ausführung zu aktivieren, wenn automatische Bereitstellungen verwendet werden. Durch die verwaltete Ausführung können automatische Bereitstellungsvorgänge in mehreren Zielkonten gleichzeitig innerhalb eines Zeitraums ausgeführt werden StackSet, was die Verarbeitungsgeschwindigkeit erhöht, insbesondere für größere Organizations.

Automatische Bereitstellungen aktivieren oder deaktivieren (Konsole)

So aktivieren oder deaktivieren Sie automatische Bereitstellungen

1. [Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com /cloudformation.](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 Automatische Distribution bearbeiten aus dem Menü Aktionen in der oberen rechten Ecke.
6. Gehen Sie in dem sich öffnenden Dialogfenster wie folgt vor:
 - a. For Automatic deployment, choose Activated or Deactivated.
 - b. Für Kontoentfernungsverhalten, wählen Sie Stapel löschen oder Stapel beibehalten. Zurückbehaltene Ressourcen bleiben in ihrem aktuellen Zustand, sind aber nicht mehr Teil von StackSet.
 - c. Fügen Sie für StackSet Abhängigkeiten abhängige hinzu StackSet ARNs und bleiben Sie dabei innerhalb von maximal 10 Abhängigkeiten.
7. Wählen Sie Speichern.

Automatische Bereitstellungen aktivieren oder deaktivieren (AWS CLI)

So aktivieren oder deaktivieren Sie automatische Bereitstellungen

1. Verwenden Sie den Befehl [update-stack-set](#) mit der Option --auto-deployment .

Der folgende Befehl aktiviert automatische Distributionen.


```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment  
  Enabled=true,RetainStacksOnAccountRemoval=true,DependsOn=ARN1,ARN2
```

Um die automatische Bereitstellung zu deaktivieren, können Sie auch `Enabled=false` als Wert für die Option `--auto-deployment` angeben, 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, und [describe-stack-set-operation](#) überprüfen Sie, ob Ihre Aktualisierung erfolgreich StackSet war.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Aktualisierung 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](#) und [Stapel löschen von StackSets](#). Um die Parameterwerte für einen Stapel zu überschreiben, siehe [Parameter auf Stapeln außer Kraft setzen](#).

Themen

- [Aktualisiere deine StackSet \(Konsole\)](#)
- [Aktualisieren Sie Ihre StackSet \(AWS CLI\)](#)

Aktualisiere deine StackSet (Konsole)

Um ein zu aktualisieren StackSet

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die 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 für Voraussetzung- Vorlage vorbereitendie Option Aktuelle Vorlage verwenden um die aktuelle Vorlage zu verwenden, oder Aktuelle Vorlage ersetzen um eine S3-URL für eine andere 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 die-Parameterdie 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 Ausführungskonfigurationkönnen Sie die Ausführungskonfiguration nach Bedarf aktualisieren.

 Note

Denken Sie daran, dass Sie die Ausführungseinstellungen nicht ändern können, wenn die Vorgänge laufen oder in der Warteschlange stehen.

14. Wenn Ihre Vorlage IAM-Ressourcen enthält, wählen Sie für Fähigkeitendie 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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).
15. Wählen Sie Weiter aus.

16. Auf der Seite Bereitstellungsoptionen festlegen geben Sie die Konten und Regionen für die Aktualisierung 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, Bereitstellungsorte, Stacks in Konten bereitstellen. Fügen Sie das Zielkonto IDs , mit dem Sie Ihr Konto erstellt haben, StackSet in das Textfeld ein und trennen Sie mehrere Zahlen durch Kommas.

[Service-verwaltete Berechtigungen] Führen Sie einen der folgenden Schritte aus:

- Wählen Sie In 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. Für Einsatzoptionengehen Sie wie folgt vor:
- Geben Sie für Maximale Anzahl gleichzeitiger Konten an, wie viele Konten gleichzeitig bearbeitet werden.
 - Geben Sie für Fehlertoleranz die maximal zulässige Anzahl von Kontoausfällen pro Region an. Die Operation wird gestoppt und nicht mit anderen Regionen fortgesetzt, sobald dieses Limit erreicht ist.
 - Für Gleichzeitigkeit der Regionen wählen Sie, wie die Regionen verarbeitet werden sollen: Sequentiell (eine Region auf einmal) oder Parallel (mehrere Regionen gleichzeitig).
 - Wählen Sie für Gleichzeitigkeitsmodus, wie sich die Gleichzeitigkeit während der Ausführung der Operation verhält.
 - Strenge Fehlertoleranz- Reduziert den Gleichzeitigkeitsgrad des Kontos, wenn Fehler auftreten, und bleibt dabei innerhalb der Fehlertoleranz +1.
 - Weiche Fehlertoleranz- Behält die von Ihnen angegebene Gleichzeitigkeitsstufe (den Wert von Maximale gleichzeitige Konten) unabhängig von Fehlern bei.
 - [Vom Dienst verwaltete Berechtigungen] Fügen Sie für StackSet Abhängigkeiten abhängige StackSet ARNs Personen hinzu, wobei die maximale Anzahl von 10

Abhängigkeiten nicht überschritten werden darf. Weitere Informationen finden Sie unter [Automatische Bereitstellungen für StackSets in aktivieren oder deaktivieren AWS Organizations](#).

- d. Wählen Sie Next (Weiter), um fortzufahren.
17. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie Bearbeiten in dem entsprechenden Abschnitt.
18. Wenn Sie bereit sind fortzufahren, wählen Sie Senden.

CloudFormation beginnt mit der Installation Ihrer Updates auf Ihrem StackSet und zeigt die Registerkarte „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 beauftragter Administrator handeln, müssen Sie `--call-as DELEGATED_ADMIN` in den Befehl einfügen.

1. Um ein zu aktualisieren StackSet

Verwenden Sie den [update-stack-set](#)Befehl, um Änderungen an Ihrem vorzunehmen StackSet.

In den folgenden Beispielen aktualisieren wir die `--parameters` Option StackSet by using. Konkret ändern wir die Standard-Snapshot-Zustellungshäufigkeit für die Zustellungsanalkonfiguration von `TwentyFour_Hours` auf `Twelve_Hours`. Da wir immer noch die aktuelle Vorlage verwenden, fügen wir die Option `--use-previous-template` hinzu.

Legen Sie mit der Option `--operation-preferences` die Verarbeitung gleichzeitiger Konten und andere Bereitstellungseinstellungen fest. Diese Beispiele verwenden zählbasierte Einstellungen. Beachten Sie, dass `MaxConcurrentCount` nicht größer sein darf als `FailureToleranceCount + 1`. Für prozentuale Einstellungen verwenden Sie 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](#)

- [Füge Stapel zu einem StackSet \(\) hinzu AWS CLI](#)

Stapel zu einer StackSet (Konsole) hinzufügen

Um Stapel zu einem hinzuzufügen StackSet

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die 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 aus StackSet , die Sie erstellt haben.
4. Wenn Sie die StackSet Option ausgewählt haben, 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 unter Stapel hinzufügen zu die StackSet Option Neue Stapel bereitstellen aus.
 - b. Gehen Sie als Nächstes je nach Ihrer StackSet Berechtigungskonfiguration wie folgt vor:
 - [Selbstverwaltete Berechtigungen] Wählen Sie für Konten, Bereitstellungsorte, Stacks in Konten bereitstellen. Fügen Sie Ihre Zielkonto-Zahlen in das Textfeld ein und trennen Sie dabei die einzelnen Zahlen durch Kommas.
 - [Dienstverwaltete Berechtigungen] Führen Sie für Einsatzziele 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) aus, um die Bereitstellung für alle bestimmten Konten vorzunehmen OUs. 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, 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).

- c. Geben Sie für Regionen an, welche AWS-Regionen in den Zielkonten, die Sie im vorherigen Schritt angegeben haben, bereitgestellt werden sollen. Standardmäßig stellt CloudFormation Stacks in den angegebenen Konten innerhalb der ersten Region bereit und fährt dann mit der nächsten fort usw., solange die Bereitstellungsfehler einer Region eine angegebene Fehlertoleranz nicht überschreiten.
 - d. Für Einsatzoptionen gehen Sie wie folgt vor:
 - Geben Sie für Maximale Anzahl gleichzeitiger Konten an, wie viele Konten gleichzeitig bearbeitet werden.
 - Geben Sie für Fehlertoleranz die maximal zulässige Anzahl von Kontoausfällen pro Region an. Die Operation wird gestoppt und nicht mit anderen Regionen fortgesetzt, sobald dieses Limit erreicht ist.
 - Für Gleichzeitigkeit der Regionen wählen Sie, wie die Regionen verarbeitet werden sollen: Sequentiell (eine Region auf einmal) oder Parallel (mehrere Regionen gleichzeitig).
 - Wählen Sie für Gleichzeitigkeitsmodus, wie sich die Gleichzeitigkeit während der Ausführung der Operation verhält.
 - Strenge Fehlertoleranz- Reduziert den Gleichzeitigkeitsgrad des Kontos, wenn Fehler auftreten, und bleibt dabei innerhalb der Fehlertoleranz +1.
 - Weiche Fehlertoleranz- Behält die von Ihnen angegebene Gleichzeitigkeitsstufe (den Wert von Maximale gleichzeitige Konten) unabhängig von Fehlern bei.
 - e. Wählen Sie Weiter aus.
6. Lassen Sie auf der Seite Specify Overrides (Außerkräftsetzungen 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 Bearbeiten in dem entsprechenden Abschnitt.
 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 Stack auf der Registerkarte Stackinstances aufgeführt sein.

Füge Stapel zu einem StackSet () hinzu AWS CLI

Note

Wenn Sie als beauftragter Administrator handeln, müssen Sie `--call-as DELEGATED_ADMIN` in den Befehl einfügen.

Um Stapel zu einer 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. Für Beispielfehle, die auf bestimmte Konten abzielen, siehe [Erstellen 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 bei denen 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 die Werte der Vorlagenparameter überschreiben, wenn Sie die Stapel zum ersten Mal erstellen, und Sie können die 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

Das Überschreiben von Parameterwerten gilt für Stapel in den von Ihnen ausgewählten Konten und Regionen. Bei StackSet Aktualisierungen werden alle Parameterwerte, die für einen Stack überschrieben wurden, nicht aktualisiert, sondern behalten ihren überschriebenen Wert bei.

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 worden sind, können Sie den Parameterwert in den einzelnen Stapeln nach Belieben überschreiben.

Informationen zum Überschreiben von StackSet Parameterwerten beim Erstellen von Stacks finden Sie unter. [Stapel hinzufügen zu StackSets](#)

Themen

- [Parameter auf Stapeln außer Kraft setzen \(Konsole\)](#)
- [Parameter auf Stapeln außer Kraft setzen \(AWS CLI\)](#)

Parameter auf Stapeln außer Kraft setzen (Konsole)

So überschreiben Sie die Parameter für bestimmte Stapel

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die 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 die StackSet Option ausgewählt ist, wählen Sie im Menü Aktionen die Option StackSetParameter überschreiben aus.
5. Geben Sie auf der Seite Bereitstellungsoptionen festlegen die Konten und Regionen für die Stapel an, für die Sie Überschreibungen erstellen werden.

Standardmäßig CloudFormation werden Stacks auf den angegebenen Konten innerhalb der ersten Region bereitgestellt und dann mit der nächsten Region fortgefahren usw., vorausgesetzt, dass die Bereitstellungsfehler einer Region eine festgelegte Fehlertoleranz nicht überschreiten.

- a. [Selbstverwaltete Berechtigungen] Wählen Sie für Bereitstellungsorthe Option Stacks in Konten bereitstellen. Fügen Sie einige oder alle Zielkonten ein IDs, mit denen Sie Ihr StackSet Konto erstellt haben.

[Service-verwaltete Berechtigungen] Führen Sie einen der folgenden Schritte aus:

- Wählen Sie In 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. Für Einsatzoptionen gehen Sie wie folgt vor:

- Geben Sie für Maximale Anzahl gleichzeitiger Konten an, wie viele Konten gleichzeitig bearbeitet werden.
- Geben Sie für Fehlertoleranz die maximal zulässige Anzahl von Kontoausfällen pro Region an. Die Operation wird gestoppt und nicht mit anderen Regionen fortgesetzt, sobald dieses Limit erreicht ist.
- Für Gleichzeitigkeit der Regionen wählen Sie, wie die Regionen verarbeitet werden sollen: Sequentiell (eine Region auf einmal) oder Parallel (mehrere Regionen gleichzeitig).
- Wählen Sie für Gleichzeitigkeitsmodus, wie sich die Gleichzeitigkeit während der Ausführung der Operation verhält.
 - Strenge Fehlertoleranz- Reduziert den Gleichzeitigkeitsgrad des Kontos, wenn Fehler auftreten, und bleibt dabei innerhalb der Fehlertoleranz +1.
 - Weiche Fehlertoleranz- Behält die von Ihnen angegebene Gleichzeitigkeitsstufe (den Wert von Maximale gleichzeitige Konten) unabhängig von Fehlern 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 StackSet Wert überschreiben aus.
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 StackSet Überschreibungswert bearbeiten die Option Auf Wert setzen. Auf diese Weise werden alle überschriebenen Werte entfernt, sobald Sie die Stapel aktualisieren.

8. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie Bearbeiten in dem entsprechenden Abschnitt.
9. Wenn Sie bereit sind fortzufahren, wählen Sie Senden.

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 Stapeln außer Kraft setzen (AWS CLI)

Note

Wenn Sie als beauftragter Administrator handeln, müssen Sie `--call-as DELEGATED_ADMIN` in den Befehl einfügen.

So überschreiben Sie die Parameter für bestimmte Stapel

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 Stapel in allen Konten in der OU mit der ID *ou-rcuk-1x5j1lwo*.

Die überschriebenen Parameterwerte gelten nur für die Konten, die sich derzeit in der Ziel-OU und der zugehörigen 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. Überprüfen Sie, ob Ihre Parameterwerte erfolgreich auf den Stapeln überschrieben wurden, indem Sie den Befehl `describe-stack-set-operation` ausführen, um den Status und die Ergebnisse

Ihrer Aktualisierungsoperation 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 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-Managementkonsole und öffnen Sie die 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. Auf der Seite Bereitstellungsoptionen festlegen wählen Sie zunächst die Konten und Regionen aus, in denen Sie die Stapel löschen möchten.

- a. [Selbstverwaltete Berechtigungen] Wählen Sie für Kontendie Option Stapel in Konten bereitstellen oder Stapel in Organisationseinheiten bereitstellen.

Wenn Sie Stapel in Konten bereitstellenwählen, fügen Sie Ihre Zielkontonummern in das Textfeld Kontonummern ein, wobei Sie mehrere Nummern durch Kommas trennen.

Wenn Sie Stapel in Organisationseinheiten bereitstellenwählen, fügen Sie eine Ziel-OU-ID in das Textfeld Organisationsnummern ein, um alle Konten, die Teil der angegebenen Organisation sind, zu erfassen.

- 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 Konto-Filtertypkönnen Sie verfeinern, bei welchen Konten Stapel gelöscht werden sollen, indem Sie eine der folgenden Optionen wählen und 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 Stacks aus den angegebenen Konten OUs sowie aus weiteren Einzelkonten.

- c. Wählen Sie für Regionen angebedie Regionen, aus denen Sie Stapel innerhalb der Zielkonten löschen möchten.

6. Für Einsatzoptionengehen Sie wie folgt vor:

- Geben Sie für Maximale Anzahl gleichzeitiger Kontenan, wie viele Konten gleichzeitig bearbeitet werden.

- Geben Sie für Fehlertoleranz die maximal zulässige Anzahl von Kontoausfällen pro Region an. Die Operation wird gestoppt und nicht mit anderen Regionen fortgesetzt, sobald dieses Limit erreicht ist.
 - Aktivieren Sie für Stacks behalten diese Option, um die Stapel und die zugehörigen Ressourcen zu speichern, wenn Sie sie aus Ihrem entfernen. StackSet Die Ressourcen bleiben in ihrem aktuellen Zustand, sind aber nicht mehr Teil von. StackSet
 - Für Gleichzeitigkeit der Regionen wählen Sie, wie die Regionen verarbeitet werden sollen: Sequentiell (eine Region auf einmal) oder Parallel (mehrere Regionen gleichzeitig).
 - Wählen Sie für Gleichzeitigkeitsmodus, wie sich die Gleichzeitigkeit während der Ausführung der Operation verhält.
 - Strenge Fehlertoleranz- Reduziert den Gleichzeitigkeitsgrad des Kontos, wenn Fehler auftreten, und bleibt dabei innerhalb der Fehlertoleranz +1.
 - Weiche Fehlertoleranz- Behält die von Ihnen angegebene Gleichzeitigkeitsstufe (den Wert von Maximale gleichzeitige Konten) unabhängig von Fehlern bei.
7. Wählen Sie Weiter aus.
 8. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Um Änderungen vorzunehmen, wählen Sie Bearbeiten in dem entsprechenden Abschnitt.
 9. Wenn Sie bereit sind, die Stapel aus Ihrem zu entfernen StackSet, wählen Sie Submit.

Nachdem das Löschen des Stacks abgeschlossen ist, können Sie auf der StackSet Detailseite auf der Registerkarte Stack-Instanzen überprüfen, ob die Stapel von Ihrem StackSet gelöscht wurden.

Lösche Stacks von deinem StackSet ()AWS CLI

Note

Wenn Sie als beauftragter Administrator handeln, müssen Sie `--call-as DELEGATED_ADMIN` in den Befehl einfügen.

Verwenden Sie den `delete-stack-instances` Befehl mit Ihrem StackSet Namen.

In diesen Beispielen verwenden wir die Option `--no-retain-stacks`, weil wir keine Stack zurückhalten. Verwenden Sie `--retain-stacks` anstelle von `--no-retain-stacks`, wenn Sie die Stack und ihre Ressourcen behalten möchten.

Geben AWS-Regionen Sie für an `--regions`, aus welchen Stapel Sie löschen möchten, z. B. `us-west-2` und `us-east-1`.

Legen Sie mit der Option `--operation-preferences` die gleichzeitige Bearbeitung von Konten und andere Einstellungen fest. Diese Beispiele verwenden zählbasierte Einstellungen. Beachten Sie, dass `MaxConcurrentCount` nicht größer sein darf als `FailureToleranceCount + 1`. Für prozentuale Einstellungen verwenden Sie stattdessen `FailureTolerancePercentage` oder `MaxConcurrentPercentage`.

So löschen Sie Stapel (selbst verwaltete 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
```

So löschen Sie Stapel (vom Dienst 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 die 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 CloudFormation StackSets

Um eine zu löschen StackSet, müssen Sie zuerst alle Stapel in der StackSet löschen. Informationen darüber, wie Sie alle Stapel löschen können, finden Sie unter [Stapel löschen von StackSets](#).

Themen

- [Lösche eine StackSet \(Konsole\)](#)
- [Löschen Sie ein StackSet \(AWS CLI\)](#)
- [Löschen von Servicerollen \(optional\)](#)

Lösche eine StackSet (Konsole)

Um ein zu löschen StackSet

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die 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 beauftragter Administrator handeln, müssen Sie `--call-as DELEGATED_ADMIN` in den Befehl einfügen.

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 `DELETED` anzeigen.

```
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 unter [Selbstverwaltete Berechtigungen erteilen](#).

Bei vom Dienst verwalteten StackSets Rollen StackSets haben die Rollen, für die automatisch erstellt wurden, das Suffix `CloudFormationStackSetsOrgAdmin` im Organisationsverwaltungs-konto und `CloudFormationStackSetsOrgMember` in jedem Zielkonto. Weitere Informationen finden Sie unter [Service-verknüpfte Rollen](#).

So löschen Sie eine Dienstrolle (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>

2. Wählen Sie im Navigationsbereich Rollen, und aktivieren Sie das Kontrollkästchen neben der Rolle, 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.

So löschen Sie eine Dienstrolle (AWS CLI)

- Verwenden Sie den folgenden delete-role-Befehl. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **y**ein und drücken dann Enter.

```
aws iam delete-role --role-name role name
```

Weitere Informationen zum Löschen von Rollen finden Sie unter [Rollen oder Instanceprofile löschen](#) 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 Voraussetzungen müssen für die Kontosperrung erfüllt sein:

- Ihre Lambda-Funktion muss den Namen `AWSCloudFormationStackSetAccountGate` tragen, um dieses Feature zu nutzen.
- Sie `AWSCloudFormationStackSetExecutionRole` müssen 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 Lambda `AWSCloudFormationStackSetAccountGate`-Funktionen zu erstellen. Um einen neuen Stapel mit einer dieser Vorlagen zu erstellen, siehe [Einen Stack von der CloudFormation Konsole aus erstellen](#).

Vorlagenspeicherort	Description
https://s3.amazonaws.com/cloudformation-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-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 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 unter [Häufige Gründe für Störungen der Stack-Operationen](#).

Weitere Informationen zu den Optionen Maximale Anzahl gleichzeitiger Konten und Fehlertoleranz finden Sie unter [StackSet Operationsoptionen](#).

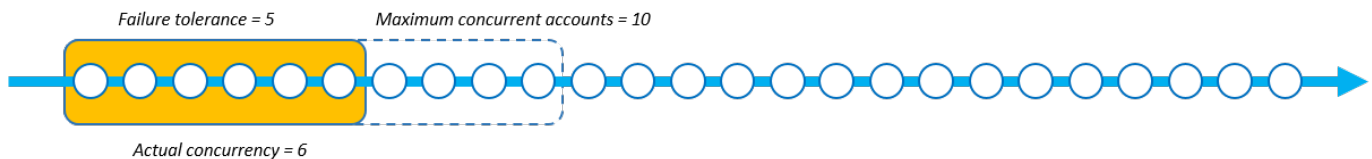
So funktionieren die einzelnen Parallelitätsmodi

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.

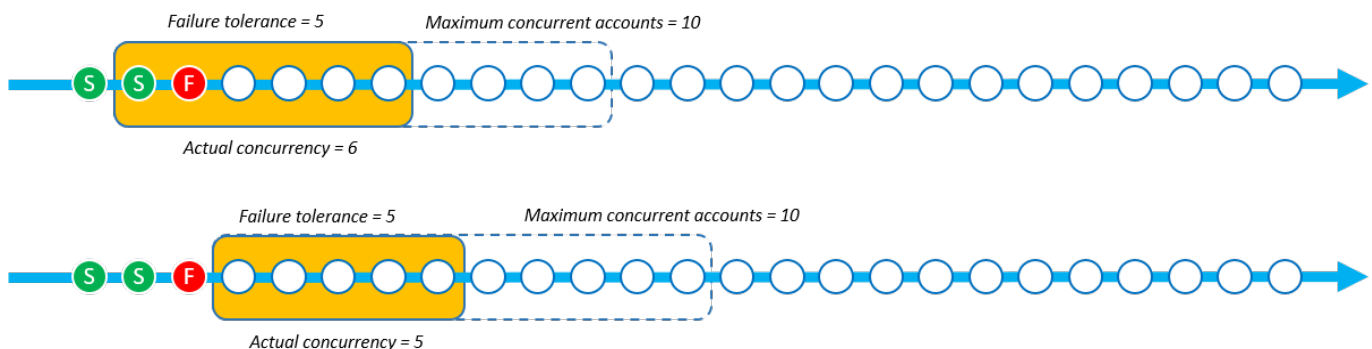
Strikte Fehlertoleranz

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

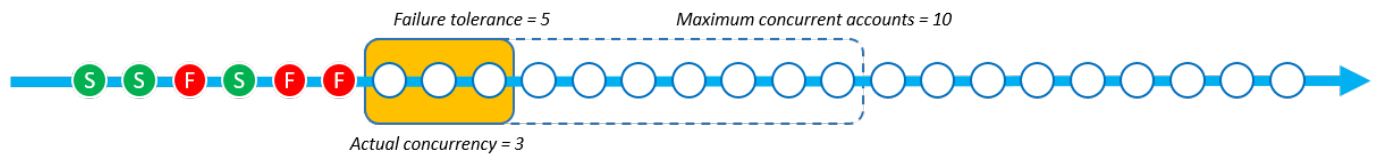


Wenn die Bereitstellung beginnt und es fehlerhafte Stack-Instances gibt, verringert sich die tatsächliche Parallelität, um eine sichere Bereitstellung zu gewährleisten. Die tatsächliche Parallelität reduziert sich von 6 auf 5, wenn 1 Stack-Instance StackSets nicht bereitgestellt werden kann.

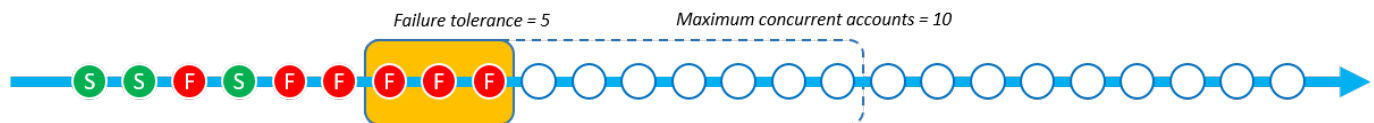


Der Modus Strikte Fehlertoleranz verringert die tatsächliche Parallelität proportional zur Anzahl der fehlerhaften Stack-Instances. Im folgenden Beispiel wird die tatsächliche Parallelität von 5

auf 3 reduziert, wenn 2 weitere Stack-Instances StackSets nicht bereitgestellt werden können, wodurch sich die Gesamtzahl der ausgefallenen Stack-Instances auf 3 erhöht.



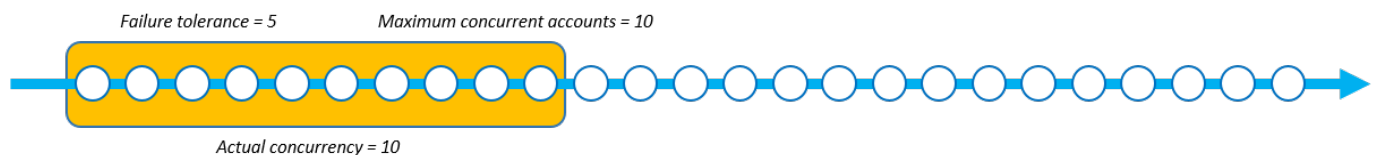
StackSets schlägt beim StackSet Vorgang fehl, wenn die Anzahl der ausgefallenen Stack-Instances dem definierten Wert für die Fehlertoleranz +1 entspricht. Im folgenden Beispiel StackSets schlägt der Vorgang fehl, wenn es 6 ausgefallene Stack-Instances gibt und der Fehlertoleranzwert 5 ist.



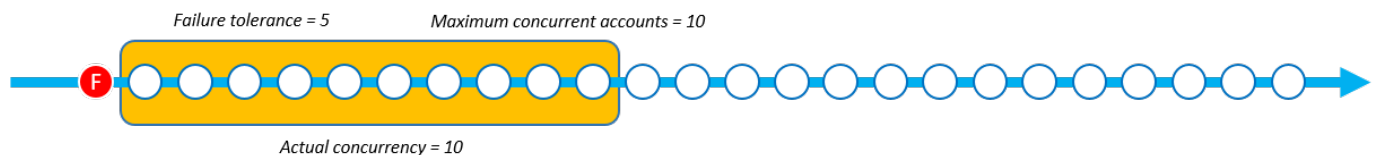
In diesem Beispiel CloudFormation wurden 9 Stack-Instances (3 erfolgreich und 6 gescheitert) bereitgestellt, bevor der StackSet Vorgang gestoppt wurde.

Lose Fehlertoleranz

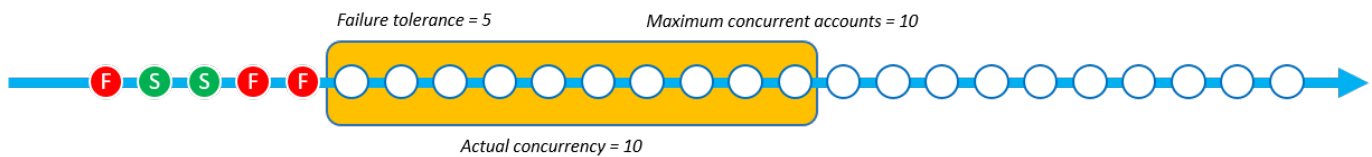
Wenn bei einem StackSet Vorgang, der Soft Failure Tolerance verwendet, der Fehlertoleranzwert auf 5 und der Wert für maximale Anzahl gleichzeitiger Konten auf 10 festgelegt ist, ist die tatsächliche Parallelität 10.



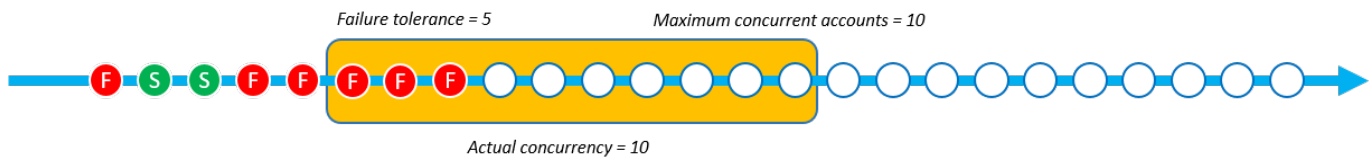
Wenn die Bereitstellung beginnt und es fehlerhafte Stack-Instances gibt, ändert sich die tatsächliche Parallelität nicht. Im folgenden Beispiel ist 1 Stack-Vorgang fehlgeschlagen, die tatsächliche Parallelität bleibt jedoch bei 10.



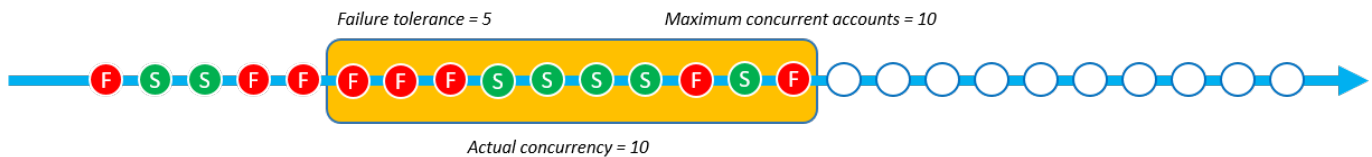
Die tatsächliche Parallelität bleibt auch nach 2 weiteren Stack-Instance-Fehlern bei 10.



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	-
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.

Auswahl des Gleichzeitigkeitsmodus (Konsole)

Wählen Sie beim Erstellen oder Aktualisieren einer StackSet auf der Seite „Bereitstellungsoptionen festlegen“ für den Parallelitätsmodus die Option Strikte Fehlertoleranz oder Weiche Fehlertoleranz aus.

Auswahl des Gleichzeitigkeitsmodus (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`

Das folgende Beispiel erstellt eine Stack-Instance unter Verwendung von `STRICT_FAILURE_TOLERANCE`, `ConcurrencyMode` wobei `FailureToleranceCount` auf 10 und `MaxConcurrentCount` auf 5 gesetzt wird.

```
aws cloudformation create-stack-instances \  
  --stack-set-name example-stackset \  
  --accounts 123456789012 \  
  --operation-preferences STRICT_FAILURE_TOLERANCE
```

```
--regions eu-west-1 \  
--operation-preferences  
ConcurrencyMode=STRICT_FAILURE_TOLERANCE,FailureToleranceCount=10,MaxConcurrentCount=5
```

Note

Ausführliche Verfahren zum Erstellen und Aktualisieren eines StackSet finden Sie in den folgenden Themen:

- [Erstellen StackSets \(selbstverwaltete Berechtigungen\)](#)
- [Erstellen StackSets \(vom Service verwaltete Berechtigungen\)](#)
- [Aktualisieren 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 veränderten 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 der 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 unter [Erkennen Sie nicht verwaltete Konfigurationsänderungen an Stacks und Ressourcen mit Drift Detection](#).

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-Instanz 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 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 aktualisierte StackSet. Diese Felder sind auf der Registerkarte „Übersicht“ der StackSet Detailseite für die ausgewählten Felder aufgeführt StackSet.

Der Vorgang zur Drifterkennung 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 nachdem Sie die Informationsleiste geschlossen haben.

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-drift](#)Befehl, um Drift bei einer gesamten Stack-Instance StackSet und den zugehörigen Stack-Instances zu erkennen.

Das folgende Beispiel initiiert die Drifterkennung auf dem StackSetstack-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-operation](#) Befehl, 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 Befehl [describe-stack-set](#) enthält die gleichen detaillierten Driftinformationen, die auch der Befehl `describe-stack-set-operation` liefert.

```

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-8c22example",
    "StackSetName": "stack-set-drift-example"
  }
}

```

Sie können den [list-stack-instances](#) Befehl 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 habenDRIFTED.

```

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-8c22example",
      "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 Befehl [describe-stack-instance](#) gibt diese Informationen ebenfalls zurück, allerdings 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 gedriftet sind, können Sie die Informationen über die Stack-Instances, die von den Befehlen `list-stack-instances` oder `describe-stack-instance` zurückgegeben werden, verwenden, um den Befehl [list-stack-instance-resource-drifts](#) 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-494PBHC076H4",
      "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 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 Befehl [stop-stack-set-operation](#). 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 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.
- Parameter importierter Stack-Instances können nicht mit der Option StackSet Details bearbeiten aktualisiert werden. Sie müssen die StackSet Override-Parameter verwenden. Weitere Informationen zum Überschreiben von Parametern finden Sie unter [Parameter auf Stapeln außer Kraft setzen](#).
- Die StackSets Kontingente und Stack-Instances gelten beim Import von Stacks. Weitere Informationen zu Kontingenten finden Sie unter [CloudFormation Kontingente verstehen](#).

Themen

- [Selbstverwalteter Stack-Import für CloudFormation StackSets](#)
- [Vom Service verwalteter Stack-Import für CloudFormation StackSets](#)
- [Stack-Importe rückgängig machen in CloudFormation StackSets](#)

Selbstverwalteter Stack-Import für CloudFormation StackSets

Mit dem CloudFormation Stack-Importvorgang können vorhandene Stacks in neue oder bestehende StackSets importiert werden, 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 importierten StackSet importiert wurde, wird der ursprüngliche Stack zu einer Stack-Instanz des angegebenen Stack-Sets.

Überlegungen für selbstverwaltete Stack-Importe

- 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 `AWSCloudFormationStackSetsExecutionRole`-Berechtigungen erteilt werden.

Themen

- [Importiert einen vorhandenen Stack in einen neuen StackSet \(Konsole\)](#)
- [Importiert einen vorhandenen Stack in einen vorhandenen StackSet \(Konsole\)](#)
- [Importiert einen Stack in einen StackSet \(AWS CLI\)](#)

Importiert einen vorhandenen Stack in einen neuen StackSet (Konsole)

Bevor Sie beginnen, identifizieren Sie den Stapel, den Sie importieren möchten.

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. StackSets.
3. Wählen Sie oben auf der StackSets-Seite **Create** aus. StackSet
4. Auf der Seite **Vorlage auswählen** geben Sie eine Vorlage mit einer der folgenden Optionen an und wählen **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 Stapel in Stapelsatz importieren.
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 Submit.

Ergebnisse: Der importierte Stack ist jetzt eine Stack-Instanz des angegebenen StackSet. Um mehr über den Status des Stapelimports zu erfahren, lesen Sie [StackSets Statuscodes](#).

Importiert einen vorhandenen Stack in einen vorhandenen StackSet (Konsole)

Bevor Sie beginnen, identifizieren Sie den Stapel, den Sie importieren möchten.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die 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 die StackSet Option ausgewählt ist, wählen Sie im Menü Aktionen die Option Stapel hinzufügen StackSet aus.

5. Wählen Sie auf der Seite Bereitstellungsoptionen festlegen die Option Stapel in Stapelsatz importieren und geben Sie die Stapel-ID des Stapels, den Sie importieren möchten, in das Feld Zu importierende Stapel 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. Auf der Seite Überschreibungen festlegen überprüfen Sie Ihre Auswahl und wählen Weiter.
8. Überprüfe auf der Seite „Überprüfen“ deine Auswahl und deine StackSet Eigenschaften. Wenn Sie bereit sind, Ihre zu erstellen StackSet, wählen Sie Absenden.

Ergebnisse: Der importierte Stack ist jetzt eine Stack-Instanz des angegebenen StackSet. Um mehr über den Status des Stapelimports zu erfahren, lesen Sie [StackSets Statuscodes](#).

Importiert einen Stack in eine 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 Stapel 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 Stapel 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 Stapel 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 Option `--stack-ids`.

```
--stack-ids "arn_1" "arn_2"
```

So klonen Sie den importierten Stapel in andere Regionen und Konten

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 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 für dienstgeführte Stack-Importe

- 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 in dem ACTIVATED Bundesstaat erfolgen, in dem die Organizations tätig sind.

- Stacks, die importiert werden, sollten in einem der Mitgliedskonten vorhanden sein und nicht im Verwaltungskonto.

Themen

- [Importieren Sie einen vom Service verwalteten Stack in einen neuen Stack 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 einen neuen Stack StackSet (Konsole)

Importieren Sie einen Stack in einen neuen StackSet mit dem AWS-Managementkonsole

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-Managementkonsole und öffnen Sie die 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. Für Voraussetzung- Vorlage vorbereitenwählen Sie Vorlage ist bereitund wählen Sie Ihre Vorlage mit einer der folgenden Optionen:
 - Für Amazon S3 URLgeben Sie Ihre Amazon S3 URL in das Feld Amazon 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 Stapel zum Stapelsatz hinzufügendie Option Stapel zum Stapelsatz importieren.
 - 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. Für Stack URL geben Sie die Amazon S3 URL ein.
7. Führen Sie unter Assoziierte Organisationseinheitenfolgende Schritte aus:
 - a. Wählen Sie Mit Organisation verknüpfen um die Stamm-OU 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 der an AWS-Managementkonsole und öffnen Sie die 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. Für Voraussetzung- Vorlage vorbereitenwählen Sie Vorlage ist bereitund wählen Sie Ihre Vorlage mit einer der folgenden Optionen:
 - Für Amazon S3 URLgeben Sie Ihre Amazon S3 URL in das Feld Amazon 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. Für den Abschnitt Assoziierte Organisationseinheiten gehen Sie wie folgt vor:
 - a. Wählen Sie Mit Organisation verknüpfen um die Stamm-OU 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 der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
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 Stapel zum Stapelsatz hinzufügendie Option Stapel zum Stapelsatz importieren.
 - b. Gehen Sie unter Stapel zum Importierenwie folgt vor
 - i. Geben Sie für Stack-ID Ihre Stack-ID ein.
 - ii. Für Stack URLgeben Sie die Amazon S3 URL ein.
 - c. Führen Sie unter Assoziierte Organisationseinheitenfolgende Schritte aus:
 - i. Wählen Sie Mit Organisation verknüpfen um die Stamm-OU 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 OU1 und Stack 3 unter OU2 liegen, geben Sie OU1 und OU2 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 Organisationseinheit in der Organisationseinheiten-ID-Liste 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 ServiceManagedStackSet \  
  --stack-ids "arn:123456789012:us-east-1:Stack1" \  
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set-name ServiceManagedStackSet \  
  --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. Durch diesen Vorgang werden keine Stack-Instances 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 CloudFormation StackSets

Wenn Sie unerwünschte Änderungen in Ihrer Stack-Instance vorgenommen haben, können Sie den Import der Stack-Instance 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.

Um einen Stapelimportvorgang rückgängig zu machen, gehen Sie wie folgt vor.

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. Löscht Stack-Instances von Ihrem. StackSet Weitere Informationen finden Sie unter [Stapel löschen von CloudFormation StackSets](#).
3. Lösche deine StackSet. Weitere Informationen finden Sie unter [Löschen CloudFormation StackSets](#).

Bewährte Methoden für die Verwendung 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.

CloudFormation StackSets Beispielvorlagen

Dieser Abschnitt enthält Links zu einigen CloudFormation Beispielvorlagen, die Ihnen bei der Verwendung 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.

Description	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
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/DataLifecycleManagerEBSSnapshotDefaultPolicy • Standardrichtlinie für EBS-gestützt — Policy.yaml AMIs https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerAMIDefault 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

Description	S3-Link
Konfigurieren Sie eine AWS Config Regel, um festzustellen, ob Root-MFA aktiviert ist	https://s3.amazonaws.com/cloudformation-stacks-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-stacks-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-stacks-sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml

Problembehebung 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 unter [Geben Sie allen Benutzern des Administratorkontos die Berechtigung, Stapel 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 Parameter ändern, die für Ihre Vorlage spezifisch sind, 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 importiert werden. StackSets Die Stack-Instance befindet sich in einem INOPERABLE Status.

Lösung: Setzen 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 Für weitere Informationen siehe [Stapel löschen von 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 zurückgegeben, 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-Managementkonsole, 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 Driftstatus 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 dieses Feature jedoch erst 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 Repositories](#) 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-Synchronisierung bei einer Strategie mit mehreren Konten finden Sie im folgenden Blog-Beitrag [Verwenden Sie CloudFormation Git-Synchronisierung zur Konfiguration von Ressourcen in Kundenkonten](#).

Themen

- [So funktioniert Git Sync mit CloudFormation](#)
- [Voraussetzungen für die Synchronisierung von Stacks mit einem Git-Repository mithilfe von Git sync](#)
- [Erstellen Sie einen Stack aus dem Repository-Quellcode mit Git-Synchronisierung](#)
- [Ermöglicht CloudFormation das Posten einer Zusammenfassung der Stack-Änderungen in Pull-Requests](#)
- [Dashboard für den Git-Synchronisierungsstatus](#)

So funktioniert Git Sync mit CloudFormation

Dieses Thema beschreibt die Funktionsweise von Git sync und stellt die wichtigsten Konzepte vor, die für die Arbeit mit Git sync erforderlich sind.

Themen

- [So funktioniert die Git-Synchronisierung](#)
- [Kommentare zu Pull-Anforderungen](#)
- [Stack-Bereitstellungsdatei](#)
- [CloudFormation Vorlagendatei](#)
- [Vorlagendefinitions-Repository](#)

So funktioniert die Git-Synchronisierung

Um Git Sync zu verwenden, müssen Sie zunächst einen Git-Anbieter CloudFormation mit der Nutzung des [CodeConnections](#) Dienstes verbinden. In den Verfahren in diesem Handbuch wird die Verbindung über die CodeConnections Konsole hergestellt. Alternativ können Sie die Verbindung auch mit dem AWS CLI erstellen. Sie können einen der folgenden Git-Anbieter verwenden:

- [GitHub](#)
- [GitHub Unternehmen](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLab selbst 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 Ihr Team Pull Requests verwendet, können Ihre 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. Die Bereitstellung einer Zusammenfassung der Änderungen in Pull Requests bedeutet, dass die Teammitglieder die Auswirkungen der vorgeschlagenen Änderungen leicht überprüfen und verstehen können, 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 enthält, die Ihren CloudFormation Stack verwalten. 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 Vorlagendefinitions-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 Dienst Rolle](#)
- [IAM-Berechtigungen für Konsolenbenutzer](#)

Git-Repository

Sie müssen ein Git-Repository haben, das auf einer der folgenden Plattformen gehostet wird.

- [GitHub](#)
- [GitHub Unternehmen](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLab selbst verwaltet](#)

Das Repository kann öffentlich oder privat sein. Sie müssen dieses Git-Repository über die CloudFormation [Connections-Konsole](#) mit diesem verbinden.

CloudFormation Vorlage

Ihr Git-Repository muss eine [CloudFormation Vorlagendatei](#) enthalten, die in den Branch eingchecked ist, den Sie mit Git Sync verbinden möchten. Diese Vorlage wird von der Stack Deployment-Datei [referenziert](#).

Git Sync Dienst Rolle

Die Git-Synchronisierung erfordert eine IAM-Rolle. Sie können eine IAM-Rolle für Ihren Stack erstellen lassen, wenn Sie die Git-Synchronisierung konfigurieren, oder Sie können eine vorhandene 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 Rolle des Git-Sync-Dienstes

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 erstellen, um die Vertrauensbeziehung zu definieren.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CfnGitSyncTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.sync.codeconnections.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Wir empfehlen Ihnen, die Zustandsschlüssel `aws:SourceArn` und `aws:SourceAccount` zu verwenden, um sich vor dem Problem der verwechselten Stellvertreter zu schützen. Das Quellkonto ist Ihre Konto-ID und der Quell-ARN ist der ARN der Verbindung im [CodeConnections](#) Dienst, 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 zu erstellen 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 Einrichtungsprozesses der Git-Synchronisierung 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

Auch wenn die Änderungssatzberechtigungen (`cloudformation:CreateChangeSet`, `cloudformation>DeleteChangeSet`, `cloudformation:ExecuteChangeSet`) für die reine Konsolennutzung nicht unbedingt erforderlich sind, werden sie empfohlen, um die vollständige Stack-Inspektion und Verwaltungsfunktionen zu ermöglichen.

Das folgende Beispiel einer IAM-Richtlinie enthält die Benutzerberechtigungen, die für die Einrichtung der Git-Synchronisierung 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 `codeconnections:CreatePullRequestForResource` enthält, wird möglicherweise eine Warnung in der IAM-Konsole angezeigt, die besagt, 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 einen Stack aus dem Repository-Quellcode mit Git-Synchronisierung

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 Stapel aus dem Repository-Quellcode

1. Melden Sie sich bei <https://console.aws.amazon.com/cloudformation> an AWS-Managementkonsole und öffnen Sie die 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 wählen Sie dann Mit neuen Ressourcen (Standard).
4. Gehen Sie auf der Seite Create stack (Stack erstellen) wie folgt vor:
 - a. Für Voraussetzung- Vorlage vorbereitenlassen Sie Eine vorhandene Vorlage auswählen ausgewählt.
 - b. Für Vorlage angebenwählen Sie Von Git synchronisierenund dann Weiter.

5. Geben Sie auf der Seite Stack-Details an 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 Stack-Bereitstellungsdatei, Erstellung der Bereitstellungsdatei:
 - Wenn Sie noch keine Stack Deployment-Datei erstellt und zu Ihrem Repository hinzugefügt haben, wählen Sie Erstellen Sie die Datei mit den folgenden Parametern und legen Sie sie in meinem Repository ab.
 - Wenn Sie eine Stack Deployment-Datei 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 für Repository-Anbieter eine der folgenden Optionen:
 - GitHub
 - GitHub Enterprise Server
 - GitLab
 - Bitbucket
 - GitLab selbst verwaltet
 - b. Wählen Sie für Verbindung eine Verbindung aus der Liste. Wenn in der Liste Verbindung keine Optionen erscheinen, wählen Sie Neue Verbindung hinzufügen um zur [Verbindungskonsole](#) zu gelangen und eine Verbindung zu Ihrem Repository zu erstellen.
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. Sie können Neue IAM-Rolle wählen, um eine neue Rolle zu erstellen, oder Vorhandene IAM-Rolle wählen, um eine vorhandene Rolle aus Ihrem AWS-Konto auszuwählen. Wenn Sie sich dafür entscheiden, eine neue Rolle zu generieren, sind die erforderlichen Berechtigungen in der Rolle enthalten.
12. Aktivieren oder deaktivieren Sie Kommentare auf Pull-Anfragen:
 - Damit in Pull-Requests für Stack-Updates Informationen nach der Änderung festgelegt werden, musst CloudFormation du den Schalter Kommentar bei Pull-Request aktivieren aktiviert lassen.
 - 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 Repositories für die Stack-Vorlagendatei an.
14. (Optional) Um die Stack-Parameter festzulegen, wählen Sie Parameter hinzufügen, geben einen Schlüssel und einen Wert für jeden Parameter an und wählen dann Weiter. Weitere Informationen finden Sie unter [Stack-Bereitstellungsdatei](#).

Um zum Beispiel einen **port=8080**-Parameter in Ihrer Stack Deployment-Datei anzugeben, gehen Sie wie folgt vor:

- a. Wählen Sie Parameter hinzufügen aus.
 - b. Geben Sie für Key (Schlüssel) **port** ein.
 - c. Geben Sie für Wert **8080** ein.
15. (Optional) Um Stapel-Tags festzulegen, wählen Sie Neues Tag hinzufügen, geben einen Tag-Schlüssel und einen Wert für jedes Tag an und wählen 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 [Stapeloptionen konfigurieren](#).

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 Sie Einreichengewählt haben, wird automatisch eine Pull-Anfrage in Ihrem Git-Repository erstellt. Sie müssen diese Pull-Anfrage in Ihr Git-Repository einbinden, um Ihren Stack zu erstellen. CloudFormation überwacht dein Git-Repository nach der Erstellung des Stacks auf Änderungen.

Aktualisieren Sie Ihren Stack aus Ihrem Git-Repository

Um den Stack zu aktualisieren, nehmen Sie Änderungen direkt an Ihrer Vorlagendatei oder der Stack-Bereitstellungsdatei in Ihrem Git-Repository vor. Nachdem Sie Ihre Änderungen in den überwachten Branch übernommen haben, CloudFormation wird der Stack automatisch aktualisiert. Wenn Sie Pull-Requests verwenden, wird automatisch ein Pull-Request in Ihrem Git-Repository erstellt, bevor der Stack aktualisiert wird. Sie müssen diese Pull-Anfrage in Ihr Git-Repository einbinden, um Ihren 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 [Dashboard für den Git-Synchronisierungsstatus](#).

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 enthält eine Zusammenfassung der hinzugefügten, geänderten oder gelöschten Ressourcen, so dass Sie eine gründliche Codeüberprüfung durchführen können, bevor Sie die Pull-Anfrage zusammenführen.

So aktivieren Sie Kommentare zu Pull-Anfragen für einen neuen Stack (Konsole)

Wenn Sie den Stack erstellen, vergewissern Sie sich auf der Seite Details zum Stack angeben unter Template definition repository, dass der Schalter Enable comment on pull request aktiviert ist. Dies ist die Standardeinstellung für neue Stapel.

So aktivieren Sie Kommentare zu Pull-Anfragen für einen bestehenden Stack (Konsole)

1. Melden Sie sich bei <https://console.aws.amazon.com/cloudformation> an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole.
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 Stapel den laufenden Stapel, den Sie aktualisieren möchten.
4. Wählen Sie die Registerkarte Git sync und dann Bearbeiten.
5. Schalten Sie auf der Seite Git-Sync-Einstellungen bearbeiten unter Schablonendefinition Repositoryden Schalter Kommentar zu Pull Request aktivieren ein.
6. Wählen Sie Update configuration (Konfiguration aktualisieren).

Dashboard für den Git-Synchronisierungsstatus

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 Bereichs 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:

Stapelstatus und optionaler detaillierter Status	Description
CREATE_COMPLETE	Erfolgreiche Erstellung eines oder mehrerer Stacks.
CREATE_IN_PROGRESS	Laufende Erstellung eines oder mehrerer Stacks.

Stapelstatus und optionaler detaillierter Status	Description
CREATE_FAILED	Fehlgeschlagene Erstellung eines oder mehrerer Stacks. Untersuchen Sie die Stack-Ereignisse auf zugehörige Fehlermeldungen. Mögliche Gründe für eine fehlgeschlagene 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.

Stapelstatus und optionaler detaillierter Status	Description
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.

Stapelstatus und optionaler detaillierter Status	Description
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 <code>resource import</code> unterstützen.

Stapelstatus und optionaler detaillierter Status	Description
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 Vorlagenkonfiguration zurückgesetzt.

Erweiterungen mit der CloudFormation Registry verwalten

Die 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 privater Nebenstellen

Themen

- [Zugehörige 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 für Sie freigegeben wurden](#)
- [Konfigurationsdaten für Erweiterungen in Ihrem Konto bearbeiten](#)
- [Ressourcentypen aufzeichnen in AWS Config](#)

Zugehörige Dokumentation

Wenn Sie als Entwickler daran interessiert sind, Ihre eigenen 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 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 AWS Ressourcen innerhalb Ihrer eigenen Ressourcen tun. AWS-Konto

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.

Erweiterungstypen

Die CloudFormation Registrierung bietet die folgenden Erweiterungstypen:

Hook

Hooks sind Validierungsprüfungen, die Ihre Stacks oder bestimmte Ressourcen überprüfen, bevor sie CloudFormation erstellt, aktualisiert oder gelöscht werden. Hooks können auch während des Erstellens und Ändern von Set-Operationen aufgerufen werden. Sie bieten einen Mechanismus zur Durchsetzung von Unternehmensstandards und Best Practices, indem sie Ressourcenkonfigurationen anhand bestimmter Anforderungen validieren. Wenn ein Hook Konfigurationen entdeckt, die nicht mit seiner Logik übereinstimmen, kann er entweder eine Warnung ausgeben oder den Bereitstellungsprozess fehlschlagen lassen, um zu verhindern, dass nicht konforme Ressourcen bereitgestellt werden. Weitere Informationen finden Sie im [CloudFormation Hooks-Benutzerhandbuch](#).

Eine Dokumentation zur Konfiguration von Hooks mithilfe der CloudFormation Konsole finden Sie in den folgenden Abschnitten des CloudFormation Hooks User Guide:

- [AWS Control Tower proaktive Steuerungen als Hooks](#)
- [Schutzhaken](#)
- [Lambda-Hook](#)

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 der Infrastrukturimplementierungen in Ihrem Unternehmen.

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 als AWS zur allgemeinen Verwendung zur Verfügung gestellt. Um eine öffentliche Erweiterung eines Drittanbieters zu verwenden, müssen Sie diese zunächst in Ihrem Konto und 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 öffentlicher Erweiterungen von Drittanbietern, die Sie explizit 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 die Ihnen von Ihrem Unternehmen oder anderen Dritten zur Verfügung gestellt 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 für Sie freigegeben wurden](#).

Die Verwendung von privat registrierten Erweiterungen und aktivierten öffentlichen Erweiterungen von Drittanbietern in Ihrem Konto ist wie die Verwendung in einer Sandbox-Umgebung. Erweiterungen werden mit Versionskontrolle verwaltet, so dass ihr Bereitstellungsverhalten an eine bestimmte Version gebunden ist. Daher funktionieren diese Erweiterungen auf dieselbe Weise wie öffentliche Erweiterungen und unterliegen denselben 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 zusätzliche Gebühren für Ihr Konto anfallen, zusätzlich zu den Gebühren für die erstellten Ressourcen. Weitere Informationen finden Sie unter [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-Managementkonsole oder die verwenden AWS CLI.

So zeigen Sie die verfügbaren und aktivierten Nebenstellen an (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.

2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie im Navigationsbereich unter Registrierung die Kategorie der Erweiterung, die Sie anzeigen möchten:
 - Öffentliche Erweiterungen zeigt die öffentlichen Erweiterungen an, die in Ihrem Konto verfügbar sind.
 1. Für Filter, Erweiterungstyp, wählen Sie Ihren Erweiterungstyp: Ressourcentypen, Module, oder Hook.
 2. Für Filter, Herausgeber, wählen Sie einen Herausgeber: AWS oder 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 Dropdown-Menü Filter zur weiteren Auswahl der anzuzeigenden Erweiterungen:
 - 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.
 - Registriert – listet alle privaten Erweiterungen auf, die Sie in diesem Konto aktiviert haben.
4. Suchen Sie oder wählen Sie den Namen der Erweiterung, um Details zur Erweiterung anzuzeigen.

So zeigen Sie die verfügbaren und aktivierten Erweiterungen an (AWS CLI)

Verwenden Sie den Befehl [list-types](#).

Verwenden Sie öffentliche Erweiterungen von Drittanbietern aus der CloudFormation Registrierung

Um eine öffentliche Erweiterung eines Drittanbieters in Ihrer Vorlage zu verwenden, müssen Sie zunächst die Erweiterung für das Konto und die Region, in der Sie sie verwenden möchten, aktivieren

. Wenn Sie eine Erweiterung aktivieren, kann sie in Stapeloperationen in dem Konto und der Region verwendet werden, in der sie aktiviert wurde.

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. Hier können Sie alle Konfigurationseigenschaften einstellen, die die Erweiterung enthält. Die Konfigurationseigenschaften definieren, wie die Erweiterung für eine bestimmte AWS-Konto Region konfiguriert wird.

Neben der Einstellung der Konfigurationseigenschaften können Sie die Erweiterung auch auf die folgenden Arten 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 öffentlichen Erweiterungszugriff](#)
- [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](#)
- [Aktivieren Sie eine öffentliche Erweiterung eines Drittanbieters in Ihrem 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 öffentlichen Erweiterungszugriff

Wenn Sie eine öffentliche Erweiterung über die 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 werden im Abschnitt Handler des Erweiterungsschemas definiert. Sie müssen eine IAM-Richtlinie erstellen, die die von der Erweiterung benötigten spezifischen Berechtigungen gewährt, und diese mit der Ausführungsrolle verknüpfen.

Zusätzlich zur Berechtigungsrichtlinie muss die Ausführungsrolle auch über eine Vertrauensrichtlinie verfügen, die es ermöglicht, die Rolle CloudFormation zu übernehmen. Folgen Sie der Anleitung unter [Erstellen einer Rolle mit benutzerdefinierten Vertrauensrichtlinien](#) im IAM-Benutzerhandbuch zum Erstellen einer Rolle mit einer benutzerdefinierten Vertrauensrichtlinie.

Vertrauensstellung

Im Folgenden finden Sie Beispiele für Vertrauensrichtlinien, die Sie verwenden können.

Sie können den Geltungsbereich der Berechtigung zur Verhinderung von dienstübergreifenden Konfusionen optional einschränken, indem Sie einen oder mehrere globale Bedingungskontextschlüssel mit dem Feld `Condition` verwenden. Weitere Informationen finden Sie unter [Serviceübergreifende Confused-Deputy-Prävention](#).

- Legen Sie den Wert `aws:SourceAccount` auf Ihre Konto-ID fest.
- Setzen Sie den Wert `aws:SourceArn` auf den ARN Ihrer Nebenstelle.

Beispiel Treuhandvertrag 1

Im Folgenden finden Sie ein Beispiel für eine IAM-Rollenvertrauensrichtlinie für eine Ressourcentyp-Erweiterung.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "iam:PassRole",  
      "Resource": "arn:aws:iam::aws:role/AWSLambdaRole",  
      "Condition": {  
        "StringEquals": {  
          "aws:SourceAccount": "123456789012",  
          "aws:SourceArn": "arn:aws:lambda::aws:execution-role-lambda-role" }  
      }  
    }  
  ]  
}
```

```

    {
      "Effect": "Allow",
      "Principal": {
        "Service": "resources.cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-
west-2:123456789012:type/resource/Organization-Service-Resource"
        }
      }
    }
  ]
}

```

Beispiel Treuhandvertrag 2

Im Folgenden finden Sie ein Beispiel für eine IAM-Rollenvertrauensrichtlinie für eine Hook-Erweiterung.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "resources.cloudformation.amazonaws.com",
          "hooks.cloudformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}

```

```
    "ArnLike": {  
        "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 jeweils neueste Version AWS-Region.

Important

Da Sie kontrollieren, ob und wann die Erweiterungen in Ihrem Konto auf die neueste Version aktualisiert werden, können Sie am Ende verschiedene Versionen derselben Erweiterung in verschiedenen Konten und Regionen einsetzen.

Dies kann zu unerwarteten Ergebnissen führen, wenn Sie dieselbe Vorlage, die diese Erweiterung enthält, für diese Konten und Regionen verwenden.

Verwenden Sie Aliase, um auf Erweiterungen zu verweisen

Sie können nicht mehr als eine Erweiterung mit einem bestimmten Namen in einer bestimmten AWS-Konto und Region aktivieren. Da verschiedene Herausgeber öffentliche Erweiterungen mit demselben Erweiterungsamen anbieten können, 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 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

Obwohl Erweiterungs-Aliase nur in einem bestimmten Konto und einer bestimmten Region eindeutig sein müssen, empfehlen wir dringend, dass Benutzer nicht denselben Alias für verschiedene öffentliche Erweiterungen von Drittanbietern in verschiedenen Konten und Regionen zuweisen. Dies könnte zu unerwarteten Ergebnissen führen, wenn Sie eine Vorlage, die den Alias der Erweiterung enthält, für mehrere Konten oder Regionen verwenden.

Häufig verwendete AWS CLI Befehle für die Arbeit mit öffentlichen Erweiterungen

Zu den häufig verwendeten Befehlen für die Arbeit mit öffentlichen Nebenstellen gehören:

- [activate-type](#) um ein öffentliches Modul oder einen Ressourcentyp eines Drittanbieters in Ihrem Konto zu aktivieren.
- [set-type-configuration](#) um die Konfigurationsdaten für eine Erweiterung in Ihrem Konto anzugeben und Hooks zu deaktivieren und zu aktivieren.
- [list-types](#) um die Nebenstellen in Ihrem Konto aufzulisten.
- [describe-type](#) um detaillierte Informationen über eine bestimmte Erweiterung oder eine bestimmte Erweiterungsversion zurückzugeben, einschließlich der aktuellen Konfigurationsdaten.

- [set-type-default-version](#) um anzugeben, welche Version einer Erweiterung die Standardversion ist.
- [deactivate-type](#) um ein öffentliches Modul oder einen Ressourcentyp eines Drittanbieters zu deaktivieren, der zuvor in Ihrem Konto aktiviert wurde.

Aktivieren Sie eine öffentliche Erweiterung eines Drittanbieters in Ihrem Konto

Das folgende Thema zeigt Ihnen, wie Sie eine öffentliche Erweiterung eines Drittanbieters in Ihrem Konto aktivieren, so dass sie in dem Konto und der Region verwendet werden kann, in der sie aktiviert wurde.

Note

Bevor Sie fortfahren, vergewissern Sie sich, dass Sie das [IAM role \(IAM-Rolle\)](#) erstellt haben, das Sie mit dieser Erweiterung verwenden werden.

Themen

- [Aktivieren Sie eine öffentliche Nebenstelle \(Konsole\)](#)
- [Aktivieren Sie eine öffentliche Erweiterung \(AWS CLI\)](#)

Aktivieren Sie eine öffentliche Nebenstelle (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 Daten zur Konfiguration der Erweiterung für Ihr Konto an

So aktivieren Sie eine öffentliche Erweiterung zur Verwendung in Ihrem Konto

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie im Navigationsbereich unter Registrierung die Option Öffentliche Erweiterungen.

4. Verwenden Sie den Filter zur Auswahl des Erweiterungstyps, und wählen Sie Drittanbieter. (Erweiterungen, die von AWS veröffentlicht werden, sind standardmäßig aktiviert.)
5. Wählen Sie die Erweiterung und dann Aktivieren.

Wenn mehrere Versionen einer Erweiterung verfügbar sind, können Sie das Menü Version verwenden, um die Version der Erweiterung auszuwählen, die Sie aktivieren möchten. Die Standardversion ist die neueste Version.

6. Für Name der Nebenstelle können Sie entweder Standard verwenden ausgewählt lassen oder Standard überschreibenwählen und dann den Alias des Nebentellentyps eingeben, den Sie für diese Nebenstelle 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 öffentlichen Erweiterungszugriff](#).
8. Für Protokollierungskonfiguration geben Sie, falls gewünscht, Informationen zur Protokollierungskonfiguration für eine Nebenstelle an. 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 die angegebene Ausführungsrolle hinzu, damit die Protokollrolle Protokolle in Ihre Protokollgruppe schreiben kann.

Bei den Schlüsselnamen `logRoleArn` und `logGroupName` wird zwischen Groß- und Kleinschreibung unterschieden.

9. Wählen Sie für Versionierung, Automatische Updates, 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 die Erweiterung eine zusätzliche Konfiguration erfordert, haben Sie die Möglichkeit, die Konfigurationsdaten jetzt oder nach der Aktivierung der Erweiterung anzugeben.

 Important

Wenn die Erweiterung, die Sie aktivieren, ein Hook ist, ist dieser Schritt erforderlich. Sie müssen ENABLED für die HookInvocationStatus-Eigenschaft angeben. Diese Operation aktiviert die Eigenschaften des Hooks, die im Schemaabschnitt `properties` des Hooks definiert sind. Weitere Informationen finden Sie unter [Referenz der Syntax des Hook-Konfigurationsschemas](#) im CloudFormation Hooks User Guide.

So geben Sie die Konfigurationsdaten an

1. Wählen Sie bei Konfiguration die Option Jetzt konfigurieren und dann Erweiterung aktivieren.

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 in das Textfeld Konfiguration JSON einen JSON-String ein, der 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.

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 bei Konfiguration die Option Später konfigurieren und dann Erweiterung aktivieren.

2. Nachdem die Erweiterung aktiviert wurde, können Sie sie konfigurieren, indem Sie von der Seite mit den aktivierten Erweiterungen zur Erweiterung navigieren und die Konfigurationsdaten eingeben.

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 Daten zur Konfiguration der Erweiterung für Ihr Konto an

Öffentliche Hooks aktivieren

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. CloudFormation empfiehlt 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.

So aktivieren Sie einen öffentlichen Hook zur Verwendung in Ihrem Konto (AWS CLI)

1. Besorgen Sie sich den ARN für Ihren Hook und speichern Sie ihn. Sie können den ARN eines Hooks mit dem AWS-Managementkonsole 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 Befehl [set-type-configuration](#), um die Konfigurationsdaten anzugeben. Das JSON, das Sie für `--configuration` übergeben, muss gegen das Konfigurationsschema des Hooks validiert werden. Um den Hook für alle Stapeloperationen zu aktivieren, müssen Sie die Eigenschaft `HookInvocationStatus` im Abschnitt `HookConfiguration` auf `ENABLED` setzen.

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
  {"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
  Organization-Service-Hook/
```

```
--type-arn $HOOK_TYPE_ARN --region us-west-2
```

Weitere Informationen zu den HookConfiguration Konfigurationsoptionen finden Sie unter [Referenz der Syntax des Hook-Konfigurationsschemas](#) im CloudFormation Hooks User Guide.

Aktivieren Sie öffentliche Module und Ressourcentypen

So aktivieren Sie eine öffentliche Erweiterung zur Verwendung in Ihrem Konto (AWS CLI)

- Verwenden Sie den Befehl [activate-type](#), um die Erweiterung zu aktivieren, und legen Sie fest, ob die Erweiterung automatisch aktualisiert werden soll, wenn eine neue kleinere Version 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 Nebenstelle 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 Option `--version-bump`, um anzugeben, ob die Erweiterung auf die neueste MAJOR-Version oder die neueste MINOR-Version aktualisiert werden soll.

```
aws cloudformation activate-type --type RESOURCE \  
  --type-name Example::Test::1234567890abcdef0 \  
  --version-bump MAJOR
```

```
--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 Details der Erweiterung über Ihr Konto aktualisieren.

So aktualisieren Sie eine öffentliche Erweiterung in Ihrem Konto (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie im Navigationsbereich unter Registrierung die Option Aktivierte Erweiterungen.
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:
 - Um das Konfigurationsschema zu aktualisieren, siehe [Konfigurationsdaten für Erweiterungen in Ihrem Konto bearbeiten](#).
 - So aktivieren oder deaktivieren Sie automatische Updates:
 1. Wählen Sie Automatische Updates bearbeiten.
 2. Wählen Sie Ein oder Aus, und wählen Sie dann 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 öffentlichen Erweiterungszugriff](#).
 - So aktualisieren Sie die Protokollierungskonfiguration:
 1. Wählen Sie Protokollierungskonfiguration bearbeiten.

2. Bearbeiten Sie die JSON-Konfiguration für die Protokollierung und wählen Sie anschließend Speichern.

Deaktivieren Sie öffentliche Erweiterungen von Drittanbietern in Ihrem Konto

Wenn Sie eine aktivierte öffentliche Erweiterung eines Drittanbieters nicht mehr benötigen, verwenden Sie die folgenden Verfahren, um sie in Ihrem Konto zu deaktivieren.

Themen

- [Deaktivieren Sie eine öffentliche Erweiterung in Ihrem Konto \(Konsole\)](#)
- [Deaktivieren Sie eine öffentliche Erweiterung in Ihrem Konto \(AWS CLI\)](#)
- [Deaktivieren eines Hooks in Ihrem Konto \(AWS CLI\)](#)

Deaktivieren Sie eine öffentliche Erweiterung in Ihrem Konto (Konsole)

So deaktivieren Sie eine öffentliche Erweiterung in Ihrem Konto

1. Melden Sie sich bei <https://console.aws.amazon.com/cloudformation> an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie im Navigationsbereich unter Registrierung die Option Aktivierte Erweiterungen.
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.

Deaktivieren Sie eine öffentliche Erweiterung in Ihrem 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
```

Deaktivieren eines Hooks in Ihrem 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 Befehl [set-type-configuration](#) und geben Sie HookInvocationStatus als DISABLED an, um einen Hook zu deaktivieren.

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 CloudFormationCloudFormation Hooks](#) User Guide.

Verwenden Sie private Erweiterungen von Drittanbietern, die für Sie freigegeben 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 Ihrem Konto entfernen](#)

IAM-Berechtigungen für die Registrierung einer privaten Erweiterung eines Drittanbieters

Als Teil der Registrierung einer privaten Erweiterung können Sie einen Amazon S3-Bucket angeben, der das Projektpaket der Erweiterung 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 [GetObject](#)-Berechtigungen für das Erweiterungspaket verfügen.

Dies gilt unabhängig davon, ob Sie entweder den [register-type](#)Befehl der AWS CLI oder den [submit](#)Befehl der CloudFormation CLI verwenden.

Weitere Informationen finden Sie unter [Actions, Resources, and Condition Keys for 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 Nebenstellen gehören:

- [register-type](#) um eine private Nebenstelle in Ihrem Konto zu registrieren.
- [describe-type-registration](#) um den aktuellen Status einer Registrierungsanfrage zurückzugeben.
- [list-types](#) um die Nebenstellen in Ihrem Konto aufzulisten.
- [describe-type](#) um detaillierte Informationen über eine bestimmte Erweiterung oder eine bestimmte Erweiterungsversion zurückzugeben, einschließlich der aktuellen Konfigurationsdaten.
- [set-type-configuration](#) um die Konfigurationsdaten für eine Erweiterung in Ihrem Konto anzugeben und Hooks zu deaktivieren und zu aktivieren.
- [set-type-default-version](#) um anzugeben, welche Version einer Erweiterung die Standardversion ist.
- [deregister-type](#) um eine private Erweiterung oder Erweiterungsversion aus Ihrem Konto zu entfernen.

Registrieren Sie eine private Erweiterung eines Drittanbieters in Ihrem Konto

Dieses Thema behandelt die Schritte zur Registrierung einer privaten Erweiterung eines Drittanbieters, die mit Ihnen geteilt wurde, damit sie in Ihrem Konto verwendet werden kann.

 Note

Bevor Sie fortfahren, vergewissern Sie sich, dass Sie über die erforderlichen [IAM-Berechtigungen](#) zur Registrierung einer privaten Nebenstelle verfügen.

So registrieren Sie eine private Erweiterung, die für Sie freigegeben ist (AWS CLI)

1. Suchen Sie den Amazon S3-Bucket, der das Projektpaket für die private Erweiterung enthält, die Sie in Ihrem Konto registrieren möchten.
2. Verwenden Sie den Befehl [register-type](#), um die private Nebenstelle in Ihrem Konto zu registrieren.

Der folgende Befehl registriert zum Beispiel den Ressourcentyp `My::Resource::Example` in dem angegebenen 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 Registrierungs-Token mit dem Befehl [describe-type-registration](#) um den Fortschritt Ihrer Registrierungsanfrage zu verfolgen.

Wenn die Registrierungsanfrage CloudFormation abgeschlossen ist, wird der Fortschrittsstatus der Anfrage auf festgelegtCOMPLETE.

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 die Erweiterung, die Sie registrieren, ein Hook ist, ist dieser nächste Schritt erforderlich. Sie müssen `ENABLED` für die `HookInvocationStatus`-Eigenschaft angeben. Diese Operation aktiviert die Eigenschaften des Hooks, die im Schemaabschnitt `properties` des Hooks definiert sind. Weitere Informationen finden Sie unter [Referenz der Syntax des Hook-Konfigurationsschemas](#) im CloudFormation Hooks User Guide.

So geben Sie die Konfigurationsdaten für einen Hook (AWS CLI) an

1. Besorgen Sie sich den ARN für Ihren Hook und speichern Sie ihn. Sie können den ARN eines Hooks mit der AWS-Managementkonsole 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 Befehl [set-type-configuration](#), um die Konfigurationsdaten anzugeben. Das JSON, das Sie für `--configuration` übergeben, muss gegen das Konfigurationsschema des Hooks validiert werden. Um den Hook zu aktivieren, müssen Sie die Eigenschaft `HookInvocationStatus` im Abschnitt `HookConfiguration` auf `ENABLED` 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 unter [Referenz der Syntax des Hook-Konfigurationsschemas](#) im CloudFormation Hooks User Guide.

Private Erweiterungen von Drittanbietern aus Ihrem Konto entfernen

Um eine private Erweiterung oder Erweiterungsversion eines Drittanbieters zu entfernen, verwenden Sie den Befehl [deregister-type](#).

Sie können eine bestimmte Version der Erweiterung oder die Erweiterung als Ganzes abmelden. Um eine Erweiterung zu deregistrieren, müssen Sie alle registrierten Versionen dieser Erweiterung einzeln deregistrieren. Wenn eine Erweiterung nur eine einzige registrierte Version hat, führt die Deregistrierung dieser Version dazu, dass auch 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.

Warning

Die Abmeldung einer privaten Erweiterung kann nicht rückgängig gemacht werden. Diese Aktion wird:

- Machen Sie die Erweiterung für alle CloudFormation Operationen unbrauchbar.
- Fehler in zukünftigen Stack-Updates verursachen, die diese Erweiterung verwenden (für Module und Ressourcentypen). Sie können die Erweiterung zwar später privat erneut

registrieren, dies kann jedoch zu Fehlern führen, wenn sie auf einer früheren Version CloudFormation basiert.

Bevor Sie fortfahren, verwenden Sie die Befehle [list-stacks](#) und [get-template](#) um zu überprüfen, dass keine aktiven Stapel diese Erweiterung verwenden.

Beispiel für Befehle zur Deregistrierung der Erweiterung

In diesem Abschnitt finden Sie Beispiele, die die verschiedenen Möglichkeiten zur Deregistrierung privater Erweiterungen zeigen.

Deregistrierung nach Typname

Verwenden Sie den Befehl [deregister-type](#) mit den Optionen `--type` und `--type-name` zur Deregistrierung Ihrer Nebenstelle.

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE
```

Deregistrierung nach Typname und Version

Um eine bestimmte Version Ihrer Erweiterung zu deregistrieren, geben Sie die Option `--version-id` in dem Befehl an.

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE \  
  --version-id 00000001
```

Tip

Um zunächst eine andere Version der Erweiterung als Standard festzulegen, verwenden Sie den Befehl. [set-type-default-version](#)

Deregistrierung nach ARN

Verwenden Sie die Option `--arn` und geben Sie den ARN Ihrer Nebenstelle an, um sie zu deregistrieren.

```
aws cloudformation deregister-type \  
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Organization-Service-Resource
```

Konfigurationsdaten für Erweiterungen in Ihrem Konto bearbeiten

Dieses Thema enthält eine Anleitung zur Bearbeitung der Konfigurationsdaten für Erweiterungen in Ihrem Konto innerhalb einer bestimmten Region. Erweiterungen können Konfigurationseigenschaften enthalten, die für alle Instances der Erweiterung für ein bestimmtes Konto und eine bestimmte Region gelten. Diese werden vom Autor der Erweiterung in der Konfigurationsdefinition der Erweiterung festgelegt. 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.

- [Referenz der Syntax des Hook-Konfigurationsschemas](#)
- [Definieren der Konfiguration einer Erweiterung auf Konto-Ebene](#)

Themen

- [Erforderliche Berechtigungen für die Verwendung dynamischer Referenzen](#)
- [Konfigurationsdaten für eine Nebenstelle bearbeiten \(Konsole\)](#)
- [Konfigurationsdaten für eine Nebenstelle bearbeiten \(AWS CLI\)](#)

Erforderliche Berechtigungen für die Verwendung dynamischer Referenzen

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 Parameter enthalten, der im Parameterspeicher AWS Systems Manager gespeichert ist, muss der Benutzer oder die Rolle, die für die Bereitstellung des Typs verwendet wird, über die Berechtigung zum Aufruf von [GetParameter](#) verfügen.
- Wenn die Konfigurationsdaten ein Geheimnis enthalten, das in gespeichert ist AWS Secrets Manager, muss der Benutzer oder die Rolle, die zur Bereitstellung des Typs verwendet wurde, über Aufrufberechtigungen verfügen [GetSecretValue](#).

Weitere Informationen finden Sie unter [Abrufen von in anderen Diensten gespeicherten Werten mit Hilfe von dynamischen Referenzen](#).

Konfigurationsdaten für eine Nebenstelle bearbeiten (Konsole)

Befolgen Sie die Schritte in diesem Abschnitt, um die Konsole für Folgendes zu verwenden:

- Anzeigen der aktuellen Konfigurationsdaten für eine Nebenstelle
- Konfigurationsdaten der Erweiterung für Ihr Konto aktualisieren

So zeigen Sie die aktuellen Konfigurationsdaten für eine Nebenstelle an

1. Melden Sie sich bei <https://console.aws.amazon.com/cloudformation> an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren AWS-Region.
3. Wählen Sie im Navigationsbereich unter Registrierung die Option Aktivierte Erweiterungen.
4. Suchen Sie die Erweiterung, die Sie anzeigen 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.

So aktualisieren Sie die Konfigurationsdaten für eine Erweiterung

1. Wählen Sie auf der Detailseite der Erweiterung auf der Registerkarte Konfiguration die Option Konfiguration bearbeiten.

Alternativ können Sie auch von Aktionen aus Bearbeiten und dann Konfiguration bearbeiten wählen.

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.

Konfigurationsdaten für eine Nebenstelle bearbeiten (AWS CLI)

Folgen Sie den Schritten in diesem Abschnitt, um Folgendes AWS CLI zu verwenden:

- Anzeigen der aktuellen Konfigurationsdaten für eine Nebenstelle
- Konfigurationsdaten der Erweiterung für Ihr Konto aktualisieren

So zeigen Sie die aktuellen Konfigurationsdaten für eine Nebenstelle an

- Verwenden Sie den Befehl [describe-type](#), um detaillierte Informationen über die Nebenstelle zu erhalten. 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.

So aktualisieren Sie die Konfigurationsdaten für eine Erweiterung

- Verwenden Sie den Befehl [set-type-configuration](#), um die Konfigurationsdaten anzugeben. Das JSON, das Sie für `--configuration` übergeben, muss anhand des Konfigurationsschemas der Erweiterung validiert werden.

Im folgenden Beispiel gibt der Befehl `set-type-configuration` die Konfigurationsdaten `{"CredentialKey": "testUserCredential"}` für die Option `--configuration` 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 [Aufzeichnung von Konfigurationen für Ressourcen von Drittanbietern mit 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 zu sehen, das für die Erstellung des Konfigurationsobjekts verwendet wird, können Sie die Eigenschaft `schema` der Aktion [DescribeType](#) 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 [writeOnlyProperties](#) im 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](#)
- [CloudFormation Referenz zu Konfigurationseigenschaften](#)
- [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 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 vorausgesetzt, 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. Die AWS Ressourcen, wie z. B. die EC2-Instance, die Sie in Ihren Stack aufnehmen, werden Ihnen jedoch zum jeweils aktuellen Tarif in Rechnung gestellt. Weitere Informationen zur AWS Preisgestaltung finden Sie auf der Detailseite der einzelnen Produkte 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. CodePipeline kopiert 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 Beispielvorgabe 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, z. B. um den Namen eines vorhandenen EC2-Schlüsselpaars,

das Sie besitzen, anzugeben. Weitere Informationen zu Artefakten finden Sie unter [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-guide/continuous-deployment/wordpress-single-instance.zip>.

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 Schlüssels KeyName in den Namen eines in der Region, in der Sie die Pipeline erstellen, vorhandenen EC2-Schlüsselpaars.
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 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 [Fortschritt des Stapels 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 Servicerelevante, 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 Bestätigungsdiaologfeld **Yes, Delete** aus.

CloudFormation löscht aus dem Stack alle Ressourcen des Stacks, wie die EC2-Instance, 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 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 .
- Um Beispielvorlagen für Pipeline-Stacks im YAML- oder JSON-Format herunterzuladen, siehe [Tutorial: Erstellen einer Pipeline mit CloudFormation](#) im AWS CodePipeline -Benutzerhandbuch.

CloudFormation Referenz zu Konfigurationseigenschaften

Wenn Sie eine CodePipeline Pipeline erstellen, fügen Sie der Pipeline eine `Deploy` Aktion mit CloudFormation als Anbieter hinzu. Anschließend müssen Sie angeben, welche CloudFormation

Aktion die Pipeline aufruft und welche Einstellungen die Aktion hat. In diesem Thema werden die 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 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 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 (Erweitert)-Abschnitt sind nur verfügbar, wenn Sie eine Pipeline bearbeiten.

Action-Modus

Die 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, 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 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_FAILED,DELETE_FAILED,, oderUPDATE_ROLLBACK_FAILED), CloudFormation wird der Stapel gelöscht und anschließend ein neuer erstellt. Wenn sich der Stack nicht in einem ausgefallenen Zustand befindet, wird er 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 CloudFormation Vorlagendatei, die dem Format folgt *ArtifactName::TemplateName*.

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 [CloudFormation Artefakte](#).

Funktionen

Bei Stacks, die bestimmte Ressourcen enthalten, ausdrückliche Bestätigung, dass diese Ressourcen 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 [CreateStack](#) API-Operation Anfrageparameter.

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 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 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 CloudFormation Vorlagen. Beispiele finden Sie unter [Walkthrough: Erstellen einer Pipeline für Test- und Produktions-Stacks](#) und [CloudFormation Referenz zu Konfigurationseigenschaften](#).

ActionMode

Die 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, 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 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_COMPLETE`, `ROLLBACK_FAILED`, `CREATE_FAILED`, `DELETE_FAILED`, oder `UPDATE_ROLLBACK_FAILED`), CloudFormation wird der Stapel gelöscht und anschließend ein neuer Stapel erstellt. Wenn sich der Stack nicht in einem ausgefallenen Zustand befindet, wird er 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 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 [CreateStack](#) API-Operation Anfrageparameter.

Diese Eigenschaft ist bedingt erforderlich. Wenn Sie IAM-Ressourcen in Ihrer Stack-Vorlage verwenden, müssen Sie diese Eigenschaft angeben.

Sie können mehrere Funktionen angeben. Das folgende Beispiel fügt der Vorlage die Eigenschaften `CAPABILITY_IAM` und `CAPABILITY_AUTO_EXPAND` hinzu:

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. B. `CreateStackOutput.json`. 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 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 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 [CloudFormation Artefakte](#).

Diese Eigenschaft ist optional.

TemplatePath

TemplatePath steht für die 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 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 .
- Um Beispielvorlagen für Pipeline-Stacks im YAML- oder JSON-Format herunterzuladen, siehe [Tutorial: Erstellen einer Pipeline mit CloudFormation](#) im AWS CodePipeline -Benutzerhandbuch.

CloudFormation Artefakte

CodePipeline führt Aufgaben an Artefakten aus, während eine Pipeline CodePipeline ausgeführt wird. Denn CloudFormation Artefakte können eine Stack-Vorlagendatei, eine Vorlagen-Konfigurationsdatei oder beides enthalten. CodePipeline verwendet diese Artefakte, um mit 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 Repositories, wie z. B. GitHub und AWS CodeCommit, Artefakte hoch, ohne sie zu komprimieren. Weitere Informationen finden Sie unter [Erstellen einer Pipeline, Phasen und Aktionen](#) im AWS CodePipeline -Benutzerhandbuch.

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 bereitgestellt und CloudFormation konfiguriert werden. Bei diesen Dateien handelt es sich um dieselben Vorlagendateien, die Sie verwenden, wenn Sie Stacks mithilfe von erstellen oder aktualisieren. 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 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 .
- Um Beispielvorlagen für Pipeline-Stacks im YAML- oder JSON-Format herunterzuladen, siehe [Tutorial: Erstellen einer Pipeline mit CloudFormation](#) im AWS CodePipeline -Benutzerhandbuch.

Verwenden von Funktionen zum Überschreiben von Parametern mit CodePipeline Pipelines

In einer CodePipeline Phase können Sie [Parameterüberschreibungen](#) für CloudFormation Aktionen angeben. Mit Parameterüberschreibungen können Sie Vorlagenparameterwerte angeben, die Werte in einer Vorlagenkonfigurationsdatei überschreiben. 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 CloudFormation Vorlage, die die Lambda-Funktion erstellt, aber den Dateinamen 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 CloudFormation Stapel abzurufen und sie als Eingabe für eine andere Aktion zu verwenden. Wenn Sie beispielsweise einen Ausgabedateinamen für eine 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"]}
}
```

CloudFormation Schnipsel aus Vorlagen

Die folgenden 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 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 .
- Um Beispielvorlagen für Pipeline-Stacks im YAML- oder JSON-Format herunterzuladen, siehe [Tutorial: Erstellen einer Pipeline mit CloudFormation](#) im AWS CodePipeline -Benutzerhandbuch.

Das CloudFormation Vorlagen-Referenzhandbuch

Ausführliche Informationen zu den Ressourcen und Komponenten, die für die Erstellung von CloudFormation Vorlagen zur Verfügung stehen, finden Sie jetzt im [CloudFormation Template Reference Guide](#). Dieses Handbuch dient als technische Referenz für Vorlagen und ergänzt die Inhalte für die ersten Schritte im CloudFormation Benutzerhandbuch.

Das CloudFormation Referenzhandbuch für Vorlagen enthält Einzelheiten über:

- Ressourcentypen und ihre Eigenschaften
- DeletionPolicy und andere Attribute auf Ressourcenebene
- Intrinsische Funktionen, wie Ref und Sub
- Transformationen (Makros) CloudFormation
- Hilfsskripte für EC2 die Instanzkonfiguration
- Maschinenlesbare Ressourcentyp-Definitionen

Dieser Leitfaden wird regelmäßig aktualisiert:

- Neue Ressourcentypen
- Änderungen bei Immobilien
- Neue Rückgabewerte
- Aktualisierungen der Dokumentation

Sicherheit in 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 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 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 CloudFormation. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen 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 CloudFormation Ressourcen unterstützen.

Topics

- [CloudFormation Stapel vor dem Löschen schützen](#)
- [Verhindern von Aktualisierungen der Stack-Ressourcen](#)
- [Datenschutz in CloudFormation](#)
- [Steuern CloudFormation Sie den Zugriff mit AWS Identity and Access Management](#)
- [CloudFormation API-Aufrufe protokollieren mit AWS CloudTrail](#)
- [Infrastruktursicherheit in CloudFormation](#)
- [Resilienz in CloudFormation](#)
- [Konformitätsvalidierung für CloudFormation](#)
- [Konfiguration und Schwachstellenanalyse in 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 Beendigungsschutz für einen Stapel mit einem beliebigen Status außer DELETE_IN_PROGRESS oder DELETE_COMPLETE einstellen.

Wenn Sie den Beendigungsschutz für einen Stack aktivieren oder deaktivieren, wird diese Entscheidung auch an alle verschachtelten Stacks weitergegeben, 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 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 [Stapeloptionen konfigurieren](#).

So aktivieren oder deaktivieren Sie den Beendigungsschutz eines vorhandenen Stacks

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren 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-Managementkonsole und öffnen Sie die CloudFormation Konsole.
2. Wählen Sie in der Navigationsleiste am oberen Rand des Bildschirms Ihren 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 Befehl [update-termination-protection](#).

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 CloudFormation finden Sie unter [Steuern 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.

Action

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. 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 festlegen, z. B. alle EC2 Instances und RDS-DB-Instances, wie im folgenden Beispiel veranschaulicht:

```
{
  "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 erteilt Aktualisierungsberechtigungen für alle Ressourcen und die Deny-Anweisung verweigert Aktualisierungen für EC2 Instances und RDS-DB-Instances. Die Deny-Anweisung überschreibt immer Zulassungsaktionen.

Sie können einen Platzhalter mit Ressourcentypen verwenden. Sie können beispielsweise Aktualisierungsberechtigungen für alle Amazon EC2-Ressourcen verweigern, z. B. Instances, Sicherheitsgruppen und Subnetze, indem Sie wie im folgenden Beispiel einen Platzhalter verwenden:

```
"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 AWS CLI eine Stack-Richtlinie anwenden, 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 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` zur Eingabe einer geänderten Richtlinie oder die Option `--stack-policy-url` zur Angabe einer Datei, 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` zur Eingabe einer geänderten Richtlinie oder die Option `--stack-policy-url` zur Angabe einer Datei, die die Richtlinie enthält.

 Note

Um eine Richtlinie zu einem vorhandenen Stack hinzuzufügen, müssen Sie über die entsprechende Berechtigung für die CloudFormation [SetStackPolicy](#)Aktion 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 [SetStackPolicy](#)Aktion 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 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 Versionskontrolle haben, können Sie eine bestimmte Version der Vorlage angeben, indem Sie `?versionId=version-id` an die URL anhängen. Weitere Informationen finden Sie unter [Arbeiten mit Objekten in einem versionierungsfähigen Bucket](#) im Amazon Simple Storage Service User Guide.

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 über die Verwendung von NoEcho zur Maskierung sensibler Informationen sowie über die Verwendung dynamischer Parameter zur Verwaltung von Geheimnissen finden Sie in der bewährten Vorgehensweise [Keine Anmeldeinformationen in Vorlagen einbetten](#).

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 [Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation](#).

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. CloudFormation generiert 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 Stapel hat jetzt den Status `UPDATE_IN_PROGRESS`. Nachdem CloudFormation die Aktualisierung des Stacks erfolgreich abgeschlossen wurde, wird der Stack-Status auf `UPDATE_COMPLETE` gesetzt.

Schlägt die Stack-Aktualisierung fehl CloudFormation, werden die Änderungen automatisch rückgängig gemacht und der Stack-Status wird auf `UPDATE_ROLLBACK_COMPLETE` gesetzt.

So aktualisieren Sie eine geschützte Ressource (AWS CLI)

- Verwenden Sie den Befehl [update-stack](#) mit der Option `--stack-policy-during-update-body` zur Eingabe einer geänderten Richtlinie oder die Option `--stack-policy-during-update-url` zur Angabe einer Datei, 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, müssen Sie über die Berechtigung zur Verwendung der Aktion [SetStackPolicy](#) verfügen.

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` zur Eingabe einer geänderten Richtlinie oder die Option `--stack-policy-url` zur Angabe einer Datei, 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 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 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

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

CloudFormation verwendet verschlüsselte Kanäle für die Dienstkommunikation im Rahmen des Modells der gemeinsamen Verantwortung.

Richtlinie für den Datenverkehr zwischen Netzwerken

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 Amazon EC2 EC2-Instances erstellen, Datenbank-Instances beenden oder Updates VPCs durchführen können. Dieselben Berechtigungen gelten immer dann, wenn CloudFormation 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 Sicherheitsanmeldeinformationen verwenden, um die Sicherheit in Ihrer Umgebung zu erhöhen. AWS

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 verweigte Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zulässig oder verweigert werden. CloudFormationunterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel.

Wenn Sie neu bei IAM sind, machen Sie sich zunächst mit den Elementen einer IAM JSON-Richtlinie vertraut. Weitere Informationen finden Sie unter [IAM JSON policy element reference](#) im IAM User

Guide. Um zu erfahren, wie Sie IAM-Richtlinien erstellen, lesen Sie das Tutorial [Erstellen und Anhängen Ihrer ersten vom Kunden verwalteten Richtlinie](#) in der IAM-Dokumentation.

Themen

- [Politische Maßnahmen für CloudFormation](#)
- [Konsolenspezifische Aktionen für CloudFormation](#)
- [Politische Ressourcen für CloudFormation](#)
- [Richtlinien-Bedingungsschlüssel für CloudFormation](#)

Politische Maßnahmen für CloudFormation

Im Action Element Ihrer IAM-Richtlinienerklärung können Sie jede API-Aktion angeben, die CloudFormation angeboten wird. Sie müssen dem Aktionsnamen die klein geschriebene Zeichenfolge `cloudformation:` voranstellen. Beispiel: `cloudformation:CreateStack`, `cloudformation:CreateChangeSet` und `cloudformation:UpdateStack`.

Wenn Sie mehrere Aktionen in einer einzigen Anweisung angeben möchten, trennen Sie diese wie folgt durch Kommas:

```
"Action": [ "cloudformation:action1", "cloudformation:action2" ]
```

Sie können auch mehrere Aktionen mittels Platzhaltern angeben. Sie können zum Beispiel alle Aktionen, deren Namen mit dem Wort `Get` beginnen, wie folgt angeben:

```
"Action": "cloudformation:Get*"
```

Eine vollständige Liste der Aktionen, die mit dem Service-Präfix `cloudformation` verbunden sind, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für CloudFormation](#) und [Aktionen, Ressourcen und Bedingungsschlüssel für AWS -Cloud-Control- API](#) in der Service Authorization Reference.

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 Stapel erstellen oder löschen, benötigen zusätzliche Berechtigungen, die auf ihren Stapelvorlagen 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 das Erstellen und Anzeigen von Stack-Aktionen sowie alle Amazon SQS-Aktionen erlaubt

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.

Gewähren Sie für alle unten aufgeführten Aktionen die Berechtigung für alle Ressourcen

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 sie außerdem die folgenden Amazon 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. Zum Beispiel sind die folgenden Berechtigungen erforderlich, wenn diese Parametertypen in der Vorlage verwendet werden:

- `ec2:DescribeKeyPairs`- Erforderlich für den Parametertyp `AWS::EC2::KeyPair::KeyName`.
- `ec2:DescribeSecurityGroups`- Erforderlich für den Parametertyp `AWS::EC2::SecurityGroup::Id`.
- `ec2:DescribeSubnets`- Erforderlich für den Parametertyp `AWS::EC2::Subnet::Id`.

- `ec2:DescribeVpcs`- Erforderlich für den Parametertyp `AWS::EC2::VPC::Id`.

Weitere Hinweise zu AWS-spezifischen Parametertypen finden Sie unter. [Geben Sie vorhandene Ressourcen zur Laufzeit mit von CloudFormation -bereitgestellten Parametertypen an](#)

Politische Ressourcen 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 CloudFormation](#) in der Service Authorization Reference. Um zu erfahren, welche Aktionen Sie mit dem ARN jeder Ressource angeben können, siehe [Aktionen definiert durch CloudFormation](#).

Sie können Aktionen für einen bestimmten Stapel festlegen, wie im folgenden Beispiel gezeigt. Wenn Sie einen ARN bereitstellen, ersetzen Sie das *placeholder text* durch Ihre ressourcenspezifischen Informationen.

Example 1: Eine Beispielrichtlinie, die das Löschen und Aktualisieren von Stack-Aktionen 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, so dass Stack löschen und Stack aktualisieren sowohl für die vollständige Stack-ID (beispielsweise `arn:aws:cloudformation:region:account-id:stack/MyProductionStack/abc9dbf0-43c2-11e3-a6e8-50fa526be49c`) als auch für den Stack-Namen (beispielsweise `MyProductionStack`) verweigert werden.

Um AWS::Serverless-Transformationen die Erstellung eines Änderungssatzes zu gestatten, fügen Sie die Berechtigung auf Ressourcenebene `arn:aws:cloudformation:region:aws:transform/Serverless-2016-10-31` hinzu, wie in der folgenden Richtlinie gezeigt.

Example 2: Eine Beispielrichtlinie, die die Aktion Änderungssatz erstellen für die angegebene Transformation erlaubt

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:aws:transform/Serverless-2016-10-31"
    }
  ]
}
```

Richtlinien-Bedingungsschlüssel für CloudFormation

In einer IAM-Richtlinienanweisung können Sie optional Bedingungen angeben, die steuern, wann eine Richtlinie in Kraft ist. 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. `festlegenDateLessThan`, 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 CloudFormation stellt 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 Amazon EC2 EC2-Instance zu starten oder einen Amazon S3 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 über den Parameter `ResourcesToImport` finden Sie in der Aktion [CreateChangeSet](#) in der AWS CloudFormation API-Referenz.

Eine Liste der möglichen `ResourcesToImport` finden Sie unter [Unterstützung für Ressourcentypen](#).

Verwenden Sie die dreiteilige Ressourcennamenskennung, 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.

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-Instances).

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 über den Parameter ResourceTypes finden Sie in der Aktion [CreateStack](#) in der AWS CloudFormation API-Referenz.

Eine Liste der Ressourcentypen finden Sie unter [CloudFormation Template Reference Guide](#).

Verwenden Sie die dreiteilige Ressourcennamenskennung, 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.

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-Instances).

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, setzen Sie den S3-Bucket für diese Benutzer auf schreibgeschützt.

`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 Stapel mit den von Ihnen hochgeladenen Vorlagen erstellen oder aktualisieren können, setzen Sie den S3-Bucket für diese Benutzer auf schreibgeschützt.

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 unter [Beispiel für identitätsbasierte IAM-Richtlinien für CloudFormation](#).

Bestätigung von IAM-Ressourcen in Vorlagen CloudFormation

Bevor Sie einen Stack erstellen können, wird Ihre Vorlage CloudFormation 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-stack](#) Befehle [create-stack](#) und 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 Aktionen [CreateStack](#) und [UpdateStack](#) verwenden, geben Sie `Capabilities.member.1=CAPABILITY_IAM` oder `Capabilities.member.1=CAPABILITY_NAMED_IAM` 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.

 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 Amazon EC2-Instances laufen

Wenn Sie eine Anwendung haben, die auf einer Amazon-EC2-Instance ausgeführt wird und die Anforderungen an AWS -Ressourcen wie Amazon-S3-Buckets oder eine DynamoDB-Tabelle stellen muss, benötigt die Anwendung AWS -Sicherheitsanmeldeinformationen. Allerdings stellt die Verteilung und Einbettung langfristiger Sicherheitsanmeldeinformationen in jede Instance, die Sie starten, eine Herausforderung und ein potenzielles Sicherheitsrisiko dar. Anstelle von langfristigen Anmeldeinformationen, wie z. B. IAM-Benutzeranmeldeinformationen, empfehlen wir, eine IAM-Rolle zu erstellen, die einer Amazon EC2-Instance zugeordnet wird, wenn die Instance gestartet wird. Eine Anwendung kann dann von der Amazon EC2-Instance temporäre Sicherheitsanmeldeinformationen abrufen. Sie müssen auf der Instance keine langfristigen Anmeldeinformationen einbetten. Um die Verwaltung von Anmeldeinformationen zu erleichtern, können Sie für mehrere Amazon EC2-Instances auch nur eine Rolle 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 Sicherheitsanmeldeinformationen unterstützen. Eine Liste der Dienste, die temporäre Sicherheitsnachweise akzeptieren, finden Sie unter [AWS Dienste, die mit IAM arbeiten](#) im IAM-Benutzerhandbuch.

Gewährung von temporärem Zugang (Verbundzugang)

In einigen Fällen möchten Sie Benutzern ohne AWS Anmeldeinformationen möglicherweise 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 er berechtigt, temporäre Sicherheitsnachweise aus der Administratorenrolle zu erhalten. Entwickler können jedoch nur über die Entwicklerrolle temporäre Sicherheitsanmeldeinformationen erhalten.

Sie können Verbundbenutzern auch Zugriff auf die gewähren. AWS-Managementkonsole Nachdem sich Benutzer bei Ihrem On-Premises-Identitätssystem authentifiziert haben, können Sie programmgesteuert eine temporäre URL erstellen, die direkten Zugriff auf die AWS-Managementkonsole 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-Managementkonsole besitzen.

Sie können mehrere verschiedene verwenden, um temporäre Sicherheitsanmeldeinformationen AWS STS APIs zu generieren. Weitere Informationen darüber, welche API Sie verwenden sollten, finden Sie unter [Vergleichen AWS STS Anmeldeinformationen](#) 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 Sicherheitsnachweise akzeptieren, finden Sie unter [AWS Dienste, die mit IAM arbeiten](#) im IAM-Benutzerhandbuch.

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-Managementkonsole, 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

- [Eine bestimmte Vorlagen-URL verlangen](#)
- [Alle CloudFormation Importvorgänge ablehnen](#)
- [Importvorgänge für bestimmte Ressourcentypen zulassen](#)
- [IAM-Ressourcen in Stapelvorlagen verweigern](#)
- [Erlauben Sie die Erstellung von Stapeln mit bestimmten Ressourcentypen](#)
- [Zugriffskontrolle basierend auf ressourcenverändernden API-Aktionen](#)
- [Einschränkung von Stack-Set-Operationen basierend auf Region und Ressourcentypen](#)
- [Erlauben Sie alle IaC-Generatoroperationen](#)

Eine bestimmte Vorlagen-URL verlangen

Die folgende Richtlinie gewährt die Berechtigung, nur die URL der Vorlage *https://s3.amazonaws.com/amzn-s3-demo-bucket/test.template* zum Erstellen oder Aktualisieren eines Stapels 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": [
            "*"
          ]
        }
      }
    }
  ]
}
```

Importvorgänge für bestimmte Ressourcentypen zulassen

Die folgende Richtlinie gewährt die Berechtigung für alle Stack-Operationen, zusätzlich zu den Import-Operationen 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 Stapelvorlagen verweigern

Die folgende Richtlinie gewährt die Berechtigung zum Erstellen von Stapeln, verweigert jedoch Anfragen, wenn die Vorlage des Stapels eine Ressource des IAM-Dienstes 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 Zugriffserlaubnis anweisung 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"
        }
    }
}
]
}

```

Erlauben Sie die Erstellung von Stapeln mit bestimmten Ressourcentypen

Die folgende Richtlinie ähnelt dem vorherigen Beispiel. Die Richtlinie gewährt die Berechtigung zur Erstellung eines Stacks, sofern die Vorlage des Stacks keine Ressource des IAM-Dienstes 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"
    }
}
]
}

```

Zugriffskontrolle basierend auf ressourcenverändernden API-Aktionen

Die folgende Richtlinie gewährt die Berechtigung, den Zugriff nach dem Namen einer ressourcenverändernden API-Aktion zu filtern. Damit wird gesteuert, mit welchen APIs IAM-Benutzern Tags zu einem Stack oder Stack-Set hinzugefügt oder entfernt werden können. Die Operation, die zum Hinzufügen oder Entfernen von Tags verwendet wird, sollte als Wert für den Bedingungsschlüssel hinzugefügt werden. Die folgende Richtlinie gewährt die Berechtigungen TagResource und UntagResource für die mutierende Operation CreateStack.

JSON

```

{
  "Version":"2012-10-17",
  "Statement": [{
    "Sid": "CreateActionConditionPolicyForTagUntagResources",
    "Effect": "Allow",
    "Action": [
      "cloudformation:TagResource",
      "cloudformation:UntagResource"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "cloudformation:CreateAction": [
          "CreateStack"
        ]
      }
    }
  ]
}

```


Einschränkung von Stack-Set-Operationen basierend auf Region und Ressourcentypen

Die folgende Richtlinie gewährt Service-verwaltete Stack-Set-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 Organisationsverwaltungs-konto angemeldet ist `123456789012`, kann er auch nur Operationen an Stack-Sets ausführen, die auf die Organisationseinheit mit der ID abzielen `ou-1fsfsrsdsfrewr`, und er kann 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-2:111122223333: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"
                ]
            }
        }
    ]
}

```

Erlauben Sie alle IaC-Generatoroperationen

Die folgende Richtlinie ermöglicht den Zugriff auf CloudFormation Aktionen im Zusammenhang mit dem Scannen von Ressourcen des IaC-Generators und der Vorlagenverwaltung.

Die erste Anweisung gewährt die Berechtigung zum Beschreiben, Auflisten und Starten von Ressourcen-Scans. Sie ermöglicht auch den Zugriff auf zusätzliche erforderliche Berechtigungen (`cloudformation:GetResource`, `cloudformation:ListResources` und `cloudformation:ListTypes`), mit denen der IaC-Generator Informationen über Ressourcen und verfügbare Ressourcentypen abrufen kann. Die zweite Anweisung gewährt volle Berechtigung 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 erforderliche IAM-Berechtigungen](#).

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 verwaltete Richtlinien für AWS CloudFormation

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [vom Kunden verwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet

ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn ein neuer AWS Dienst gestartet wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AWSCloudFormationFullAccess

Sie können `AWSCloudFormationFullAccess` an Ihre Benutzer, Gruppen und Rollen anfügen.

Diese Richtlinie gewährt Berechtigungen, die den vollen Zugriff auf CloudFormation Aktionen und Ressourcen ermöglichen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `cloudformation`— Ermöglicht Prinzipalen, alle CloudFormation Aktionen für alle Ressourcen auszuführen.

Die Berechtigungen für diese Richtlinie finden Sie [AWSCloudFormationFullAccess](#) im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSCloudFormationReadOnlyAccess

Sie können `AWSCloudFormationReadOnlyAccess` an Ihre Benutzer, Gruppen und Rollen anfügen.

Diese Richtlinie gewährt Berechtigungen, die den schreibgeschützten Zugriff auf CloudFormation Ressourcen und Aktionen ermöglichen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `cloudformation`— Ermöglicht Prinzipalen das Ausführen schreibgeschützter CloudFormation Aktionen wie das Beschreiben von Stacks, das Auflisten von Ressourcen und das Anzeigen von Vorlagen, aber nicht das Erstellen, Aktualisieren oder Löschen von Stacks.

Die Berechtigungen für diese Richtlinie finden Sie [AWSCloudFormationReadOnlyAccess](#) im Referenzhandbuch für verwaltete Richtlinien.AWS

CloudFormation Aktualisierungen AWS verwalteter Richtlinien

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien CloudFormation seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst. Abonnieren Sie den RSS-Feed auf der Seite CloudFormation Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderungen	Beschreibung	Datum
AWSCloudFormationReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFormation Es wurden neue Berechtigungen hinzugefügt, um <code>cloudformation:BatchDescribe*</code> Aktionen für Batch-Describe-Operationen zuzulassen.	30. Januar 2026
AWSCloudFormationReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFormation neue Berechtigungen wurden hinzugefügt, um <code>cloudformation:Detect*</code> Aktionen für Funktionen zur Erkennung von Stack-Drift zu ermöglichen.	13. November 2019
AWSCloudFormationReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFormation neue Berechtigungen zum Zulassen <code>cloudformation:EstimateTemplateCost</code> <code>cloudformation:Get*</code> , und <code>cloudformation:ValidateTemplate</code> Aktionen hinzugefügt.	2. November 2017
AWSCloudFormationFullAccess – Neue Richtlinie	CloudFormation hat eine neue AWS verwaltete Richtlinie hinzugefügt, die vollen Zugriff	26. Juli 2019

Änderungen	Beschreibung	Datum
	auf CloudFormation Aktionen und Ressourcen bietet.	
AWS CloudFormation Read-Only Access – Aktualisierung auf eine bestehende Richtlinie	CloudFormation neue Zugriffsberechtigungen <code>cloudformation:DetectStackDrift</code> und <code>cloudformation:DetectStackResourceDrift</code> Aktionen zur Erkennung von Stack-Drift hinzugefügt.	6. Februar 2019
AWS CloudFormation Read-Only Access – Neue Richtlinie	CloudFormation hat eine neue AWS verwaltete Richtlinie hinzugefügt, die nur Lesezugriff auf CloudFormation Aktionen und Ressourcen bietet.	6. Februar 2015
CloudFormation hat begonnen, Änderungen zu verfolgen	CloudFormation hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	6. Februar 2015

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, verwendet CloudFormation 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 [Stapeloptionen konfigurieren](#). Sie können die Service-Rolle auch ändern, wenn Sie den Stack in der Konsole aktualisieren, oder [DeleteStack](#) den Stack über die API. 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 Übergabe einer Rolle an einen AWS 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 die Berechtigung haben, Operationen auf diesem Stapel durchzuführen, können diese Rolle verwenden, unabhängig davon, ob diese Benutzer die Berechtigung `iam:PassRole` haben 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:SourceArn](#) und die [aws:SourceAccount](#) globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken 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 Wert `aws:SourceArn` nicht die Konto-ID enthält, müssen Sie beide globalen Bedingungskontextschlüssel verwenden, um die Berechtigungen zu begrenzen.

Das folgende Beispiel zeigt, wie Sie die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globale Bedingung verwenden können, CloudFormation um das Problem des verwirrten Stellvertreters zu vermeiden.

Beispiel einer Vertrauensrichtlinie, die die Bedingungsschlüssel `aws:SourceArn` und `aws:SourceAccount` verwendet

Ruft bei Registrierungsdiensten AWS -Security-Token-Service (AWS STS) auf, CloudFormation um eine Dienstrolle in Ihrem Konto zu übernehmen. Diese Rolle ist für `ExecutionRoleArn` in der [RegisterType](#) Operation und `LogRoleArn` in der [LoggingConfig](#) Operation konfiguriert. Weitere Informationen finden Sie unter [Konfigurieren Sie eine Ausführungsrolle mit IAM-Berechtigungen und einer Vertrauensrichtlinie für den öffentlichen Erweiterungszugriff](#).

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 unter [Richten Sie globale Schlüssel ein, um Confused-Deputy-Probleme zu mindern..](#)

Weitere Informationen finden Sie unter [Aktualisieren einer Rollenvertrauensrichtlinie](#) im IAM-Benutzerhandbuch.

Weiterleitung von Zugriffsanfragen (FAS) und Bewertung von Berechtigungen

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 zu verwaltenden Ressourcen verfügen. Wenn ein Benutzer seine eigene IAM-Rolle angibt, CloudFormation übernimmt er diese Rolle, um in seinem Namen Dienstinteraktionen durchzuführen.

Unabhängig davon, ob der Benutzer eine IAM-Rolle bereitstellt, CloudFormation generiert es für jeden Ressourcenvorgang ein neues, abgegrenztes FAS-Token. Folglich werden die [FAS-bezogenen Bedingungsschlüssel](#), einschließlich `aws:ViaAWSService` in beiden Szenarien ausgefüllt.

Die Verwendung von FAS wirkt sich darauf aus, wie IAM-Richtlinien während des Betriebs bewertet werden. CloudFormation Wenn Sie einen Stapel mit einer Vorlage erstellen, die Ressourcen enthält, die von FAS-bezogenen Bedingungsschlüsseln betroffen sind, kann es zu einer Verweigerung von Berechtigungen 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 konsequent durchgesetzt, unabhängig davon, ob während des Stack-Vorgangs eine IAM-Rolle angegeben wird oder nicht. Das liegt daran, dass der Bedingungsschlüssel `aws:ViaAWSService` aufgrund der Verwendung von FAS immer auf `true` 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 Stapelvorlage

Wenn ein Benutzer beispielsweise einen Stapel mit der folgenden Beispielvorlage erstellt, wird `aws:ViaAWSService` auf `true` gesetzt, und die Rollenberechtigungen werden durch die FAS-Richtlinie außer Kraft gesetzt. Die Erstellung von Stapeln wird durch `Statement2` der IAM-Richtlinie beeinflusst, die die Aktion `CreateKey` verweigert. Dies führt zu einem Fehler mit verweigerter Berechtigung.

```

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 über FAS finden Sie unter [Forward access sessions](#) im IAM User Guide.

Note

Die meisten Ressourcen halten sich an dieses Verhalten. Wenn Sie jedoch beim Erstellen, Aktualisieren oder Löschen einer Ressource einen unerwarteten Erfolg oder Misserfolg erleben und Ihre IAM-Richtlinie FAS-bezogene Bedingungsschlüssel enthält, gehört die betreffende Ressource wahrscheinlich zu einer kleinen Untergruppe von Ressourcen, die nicht diesem Standardmuster folgen.

CloudFormation API-Aufrufe protokollieren mit AWS CloudTrail

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": "AIDAABCDEFGHijklmOPQ",
    "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": "AIDAABCDEFGHijklmOPQ",
    "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": "AIDAABCDEFGHJKLMNOPQ",
        "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": "AIDAABCDEFGHijklmOPQ",
    "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": "AIDAABCDEFGHijklmOPQ",
    "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 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 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.

Sie können diese API-Operationen von jedem Netzwerkstandort aus aufrufen, CloudFormation unterstützt jedoch ressourcenbasierte Zugriffsrichtlinien, die Einschränkungen auf der Grundlage der Quell-IP-Adresse beinhalten können. Sie können auch 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 CloudFormation Ressource effektiv nur von der spezifischen VPC innerhalb des AWS Netzwerks isoliert.

Resilienz in 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 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. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der [AWS Sicherheitsdokumentation](#).

Konfiguration und Schwachstellenanalyse in 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

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 [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 [Abrufen von in anderen Diensten gespeicherten Werten mit Hilfe von dynamischen Referenzen](#).

[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 [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 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 [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 options sets in Amazon VPC](#) im Amazon VPC User Guide.
- 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.cloudformation` — Erstellt einen Endpunkt für API-Operationen. CloudFormation

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`.

AWS-Regionen Dort, wo FIPS-spezifische Endpunkte unterstützt werden, können Sie auch einen Schnittstellenendpunkt für die CloudFormation Verwendung des folgenden Dienstnamens erstellen:

- `com.amazonaws. region.cloudformation-fips` — Erstellt einen Endpunkt für die CloudFormation API, der dem [Federal Information Processing Standard \(FIPS\) 140-2](#) entspricht.

[Eine vollständige Liste der Endpunkte finden Sie unter CloudFormation Endpunkte und Kontingente in der CloudFormation Allgemeine Amazon Web Services-Referenz](#)

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 unter [Kontrolle des Zugriffs auf VPC-Endpunkte mit Endpunktrichtlinien](#) im AWS PrivateLink Leitfaden.

Beispiel: VPC-Endpunktrichtlinie für Aktionen CloudFormation

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

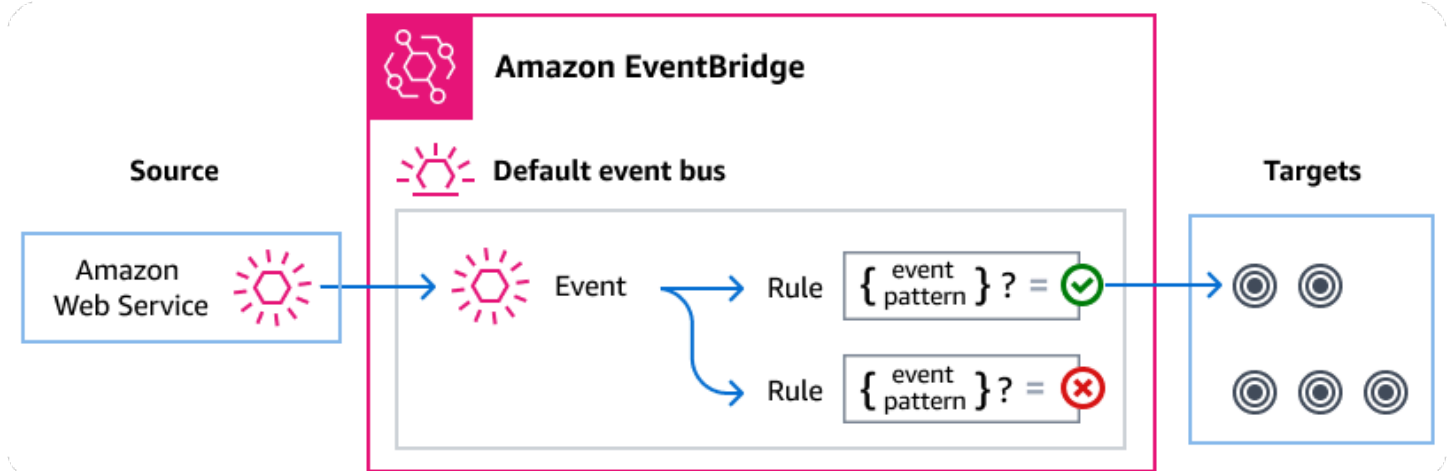
```
{
  "Statement": [
    {
      "Action": "cloudformation:*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    },
    {
      "Action": "cloudformation:CreateStack",
      "Effect": "Deny",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```


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.

Wie bei vielen AWS Services CloudFormation generiert und sendet er Ereignisse an den EventBridge Standard-Event-Bus, der in jedem automatisch bereitgestellt wird. AWS-Konto Ein Event Bus ist ein Router, der Ereignisse empfängt und sie an null oder mehr Ziele weiterleitet. Die Regeln, die Sie für den Event Bus festlegen, werten die Ereignisse aus, sobald sie eintreffen. Jede Regel prüft, ob ein Ereignis dem Ereignismuster der Regel entspricht. Wenn das Ereignis übereinstimmt, sendet der Event Bus das Ereignis an das/die angegebene(n) Ziel(e).

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. Diese Ereignisse werden garantiert zugestellt, und sie können auch außer der Reihe 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. 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	Description	Ereignisquelle
Änderung des Ressourcenstatus	Aktualisierungen, die für einen Stack durchgeführt 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
Stapel-Statusänderung	Stellt eine Statusänderung eines bestimmten Stacks dar. Einzelheiten zum Code finden Sie unter Stack-Statuscodes.	AWS CloudFormation

Ereignistyp	Description	Ereignisquelle
Statusänderung der Drifterkennung	Stellt eine vom Benutzer initiierte Aktualisierung der Abweichungserkennung für einen bestimmten Stack dar. Eine vollständige Liste der vollständig veränderlichen und unveränderlichen Typen, die die Drifterkennung unterstützen, finden Sie unter Unterstützung für Ressourcentypen.	AWS CloudFormation
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 unter Stack-Instance-Statuscodes.	AWS CloudFormation
StackSet Betriebssatus	Stellt eine Statusänderung einer bestimmten StackSet Operation dar. Einzelheiten zum Code finden Sie unter StackSets 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	Description	Ereignisquelle
Repository-Synchronisierungsstatus-Änderung	Stellt eine Statusänderung bei der Synchronisierung eines Git-Repositorys dar.	AWS CodeConnections

Ereignistyp	Description	Ereignisquelle
Ressourcensynchronisations-Statusänderung	Stellt eine Statusänderung einer Git-Ressourcen-Synchronisierung 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.

Die von Ihnen angegebenen Ziele benötigen möglicherweise bestimmte Berechtigungen oder Konfigurationen. 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 erstellen, indem Sie eines dieser vordefinierten Ereignismuster verwenden 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 kommt. 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 passt zum Beispiel zu allen Ereignissen, die den Ressourcenstatus ändern:

```
{
  "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::S3::Bucket` oder eine `AWS::SNS::Topic` Resource 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 [AWS Dienstereignis-Metadaten](#) 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 Feld `source` 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

- [Ereignisdetail Ressourcenstatusänderung](#)
- [Stack Status Änderung Ereignis Detail](#)
- [Ereignisdetail Drift Detection Status Change](#)
- [StackSet Details zum Ereignis „Statusänderung“](#)
- [StackSet Einzelheiten zum Ändern des Stack-Instance-Status](#)
- [StackSet Einzelheiten zum Ereignis „Betriebsstatus ändern“](#)
- [Repository Sync Status Änderung Ereignis Detail](#)
- [Ereignisdetail Ressourcensynchronisationsstatusänderung](#)

Ereignisdetail Ressourcenstatusänderung

Nachfolgend finden Sie die Detailfelder für Ereignisse zur Änderung des Ressourcenstatus.

Die `detail-type` Felder `source` und `detail` sind enthalten, da sie spezifische Werte für 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 Ressourcenstatusänderung

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": ""
    }
}

```

Stack Status Änderung Ereignis Detail

Nachfolgend finden Sie die Detailfelder für Stack Status Change-Ereignisse.

Die `detail-type` Felder `source` und `detail` sind enthalten, da sie spezifische Werte für 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 unter [Stack-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: Ereignis Stack Status

Im Folgenden finden Sie ein Beispiel für ein Stackstatusereignis, 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":""
  }
}
```

Ereignisdetail Drift Detection Status Change

Im Folgenden finden Sie die Detailfelder für Ereignisse zur Stack-Abweichungserkennung.

Die `detail-type` Felder `source` und `detail` sind enthalten, da sie spezifische Werte für 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-Abweichungserkennungseignissen 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-Abweichungserkennungseignissen 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 "Stack Drift Detection"

Im Folgenden sehen Sie ein Beispiel für ein Stack Drift Detection Ereignis. Dieses Ereignis gibt Auskunft darüber, ob die Drift-Erkennung für den angegebenen Stack abgeschlossen CloudFormation 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 `detail` sind enthalten, da sie spezifische Werte für 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 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 der Stack-Set-Operation.

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 `source-type` sind enthalten, da sie spezifische Werte für 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"
    }
}
}

```

Repository Sync Status Änderung Ereignis Detail

Nachfolgend finden Sie die Detailfelder für Repository Sync Status Change Ereignisse.

Die `detail-type` Felder `source` und `sink` sind enthalten, da sie spezifische Werte für 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 Repository Sync-Statusereignisse ist dieser Wert `Git Sync Repository Sync Status Change`.

source

Identifiziert den Service, aus dem das Ereignis stammt. Für Git-Synchronisierungsereignisse 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 Repository-Synchronisationsstatus-Ereignisse umfassen 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 ARN, der mit dem Git-Repository verknüpft ist.

repositoryLinkId

Die eindeutige ID, die dem Repository-Link zugeordnet ist.

ownerId

Die eindeutige ID, die dem Eigentümer des Repositorys zugeordnet ist.

commit

Die eindeutige ID, die mit der Repository-Übertragung verknüpft ist.

branch

Die eindeutige ID, die mit dem Repository-Zweig verknüpft ist.

syncType

Die Art der Synchronisierung, die durchgeführt wird.

status

Der aktuelle Synchronisierungsstatus des Repositorys.

Zulässige Werte: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

previousSync

Der Synchronisationsstatus vor dem aktuellen Status.

Zulässige Werte: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Example Beispiel: Ereignis "Änderung des Repository-Synchronisationsstatus"

Im Folgenden finden Sie ein Beispiel für ein Ereignis zur Änderung des Repository-Synchronisierungsstatus. Die Ereignisdetails, mit denen das Repository erfolgreich synchronisiert CodeConnections wurde.

```
{
  "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",
  }
}
```

```
}
```

Ereignisdetail Ressourcensynchronisationsstatusänderung

Nachfolgend finden Sie die Detailfelder für Ereignisse zur Änderung des Ressourcensynchronisationsstatus.

Die `detail-type` Felder `source` und `detail` sind enthalten, da sie spezifische Werte für 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 Repository Sync-Statusereignisse ist dieser Wert `Git Sync Repository Sync Status Change`.

`source`

Identifiziert den Service, aus dem das Ereignis stammt. Für Git-Synchronisierungereignisse 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.

Bei Ereignissen zum 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 mit der Repository-Übertragung verknüpft ist.

`repositoryName`

Der Name des Git-Repositorys.

`branch`

Die eindeutige ID, die mit dem Repository-Zweig verknüpft ist.

`syncType`

Die Art der Synchronisierung, die durchgeführt wird.

`syncTarget`

Der Zielstapel für die Ressourcensynchronisierung.

`status`

Der aktuelle Synchronisierungsstatus des Repositorys.

Zulässige Werte: `FAILED` | `INITIATED` | `IN_PROGRESS` | `SUCCEEDED`

`previousSync`

Der Synchronisationsstatus vor dem aktuellen Status.

Zulässige Werte: `FAILED` | `INITIATED` | `IN_PROGRESS` | `SUCCEEDED`

Example Beispiel: Ressource Sync Statusänderung Ereignis

Im Folgenden finden Sie ein Beispiel für ein Ereignis zur Änderung des Ressourcen-Synchronisationsstatus. Die Ereignisdetails, mit denen die Ressource erfolgreich synchronisiert CodeConnections wurde.

```
{
```



```
"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	Description	Wert	Optimierungsstrategie
cfn-signal Wartebedi ngung Daten	Die maximale Datenmenge, die cfn-signal passieren kann.	4,096 Bytes	Um eine größere Menge zu übergeben , senden Sie die Daten an einen Amazon S3-Bucket und verwenden Sie cfn-signal , um die Amazon S3-URL an diesen Bucket zu übergeben.
Benutzerdefinierte Ressourcenantwort	Die maximale Menge an Daten, die ein benutzerdefinierter Ressourcenanbieter weitergeben kann.	4,096 Bytes	
Dynamische Referenzen pro Vorlage	Die maximale Anzahl dynamischer Verweise, die in einer einzelnen CloudForm ation Stack-Vorlage zulässig sind.	60 dynamische Referenzen in einer Stapelvorlage	
Hooks pro Konto	Die maximal zulässige Anzahl von Hooks pro Konto und Region.	100 Hooks	

Kontingente	Description	Wert	Optimierungsstrategie
Hooks pro Ressource	Die maximale Anzahl von Hooks, die pro Ressource konfiguriert werden können.	100 Hooks	
Größe der Hookkonfiguration	Die maximale Menge an Konfigurationsdateien, 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 Mapping-Attributen 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 der einzelnen Mapping-Namen.	255 Zeichen	
Module	Die maximale Anzahl von Modulen, die Sie in der CloudFormation Registrierung registrieren können, pro Konto und Region.	100 Module	

Kontingente	Description	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, Deregistrieren 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 zum Beispiel eine verschachtelte Stapelhierarchie 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	Description	Wert	Optimierungsstrategie
Ausgaben	Die maximale Anzahl von Ausgaben, die Sie in Ihrer CloudFormation Vorlage deklarieren können.	200 Ausgaben	
Ausgabename	Die maximale Größe eines Ausgabenamens.	255 Zeichen	
Parameter	Die maximale Anzahl von Parametern, 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 Parameternamens.	255 Zeichen	
Parameterwert	Die maximale Größe eines Parameterwertes.	4,096 Bytes	Um einen größeren parametrisierten Wert zu verwenden, erstellen Sie mehrere Parameter und verwenden dann die Funktion <code>Fn::Join</code> , um die mehreren Werte zu einem einzigen Wert zu verketteten.

Kontingente	Description	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, DeregisterType um 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 zu einem bestimmten Zeitpunkt an Stapeloperationen (Erstellen, Aktualisieren oder Löschen) in Ihrer Region beteiligt sein können.	Verwenden Sie die DescribeAccountLimits API, um das aktuelle Limit für ein Konto in einer bestimmten Region zu ermitteln.	

Kontingente	Description	Wert	Optimierungsstrategie
Ressourcenname	Die maximale Größe eines Ressourcenname.	255 Zeichen	
Stacks	Die maximale Anzahl von CloudFormation Stacks, die Sie erstellen können.	2000 Stacks	Um mehr Stapel zu erstellen, löschen Sie Stapel, die Sie nicht benötigen, oder beantragen Sie eine Erhöhung der maximalen Anzahl von Stapeln in Ihrem AWS-Konto. Weitere Informationen finden Sie unter AWS Servicekontingente im Allgemeine AWS-Referenz.
Stack name	Die maximale Größe eines Stapelnamens.	128 Zeichen	

Kontingente	Description	Wert	Optimierungsstrategie
StackSets	Die maximale Anzahl von CloudFormation Stack-Sets, die Sie in Ihrem Administratorkonto erstellen können.	1 000 Stack-Sets	Um mehr Stapelsätze zu erstellen, löschen Sie Stapelsätze, die Sie nicht benötigen, oder beantragen Sie eine Erhöhung der maximalen Anzahl von Stapelsätzen in Ihrem 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-Instances pro Stack-Set	Um mehr Stack-Instances zu erstellen, löschen Sie Stack-Instances, die Sie nicht benötigen, oder beantragen Sie eine Erhöhung der maximalen Anzahl von Stack-Instances in Ihrem AWS-Konto. Weitere Informationen finden Sie unter AWS Servicekontingente im Allgemeine AWS-Referenz.

Kontingente	Description	Wert	Optimierungsstrategie
StackSets Instanzoperationen	Die maximale Anzahl von Stack-Instances in allen 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 einen Stapelsatz zu einem bestimmten Zeitpunkt.	10 000 Vorgänge	
Mit einem S3-Objekt importierte Stapel pro Stapelsatzoperation	Die maximale Anzahl von Stapeln, die Sie mit einem S3-Objekt pro Stapelsatzvorgang importieren können.	200 Stacks	
Stapel, die mit Inline-Stapel-IDs pro Stapelsatzoperation importiert werden	Die maximale Anzahl von Stacks, die Sie mithilfe eines IDs Inline-Stack-Per-Stack-Set-Vorgangs importieren können.	10 Stapel	

Kontingente	Description	Wert	Optimierungsstrategie
Vorlagentextgröße in einer Anforderung	Die maximale Größe eines Vorlagenkörpers, die Sie in einer <code>CreateStack</code> , <code>UpdateStack</code> oder <code>ValidateTemplate</code> -Anfrage 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 Vorlagenkörpers, den Sie in einem Amazon S3-Objekt für eine <code>CreateStack</code> , <code>UpdateStack</code> oder <code>ValidateTemplate</code> -Anfrage mit einer Amazon 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 der Beschreibung einer Vorlage.	1,024 Bytes	
Versionen pro Hook	Die maximale Anzahl von Versionen, die Sie pro Hook erstellen können.	100 Versionen	

Verfügbarkeit von Features

CloudFormation Möglicherweise sind nicht alle Funktionen von in jeder Region verfügbar. Weitere Informationen über AWS-Regionen finden Sie unter [Globale Infrastruktur Region Tabelle](#).

- 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 Service 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 CloudFormation können Probleme beim Erstellen, Aktualisieren oder Löschen von CloudFormation Stacks auftreten. Die folgenden Abschnitte können zum Beheben von häufig auftretenden Problemen helfen.

Allgemeine Fragen zu finden CloudFormation Sie unter. [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 Betriebs IDs - und Fehlerfilterung für eine gezielte Problembehandlung. CloudFormation weist Bereitstellungsereignissen einen bestimmten Vorgang IDs zu, sodass verwandte Ereignisse leichter nachverfolgt und korreliert werden können. Sie können Stack-Ereignisse nach Vorgangs-ID filtern, um sich auf bestimmte Bereitstellungsphasen zu konzentrieren, und Fehlerfilter verwenden, um die Hauptursache von Bereitstellungsfehlern schnell zu ermitteln. Siehe [Stack-Ereignisse nach Vorgang anzeigen](#)
- Verwenden Sie die CloudFormation Konsole, um den Status Ihres Stacks einzusehen. In der Konsole können Sie eine Liste von Stack-Ereignissen anzeigen, während Ihr Stack erstellt, aktualisiert oder gelöscht wird. Zu den Stack-Ereignissen gehören Operationen IDs , mit denen Sie verwandte Ereignisse korrelieren und einzelne Vorgänge verfolgen können. Für eine gezielte Problembehandlung können Sie auf die Vorgangs-ID klicken. Suchen Sie in der Liste der Vorgangereignisse nach dem Fehlerereignis und sehen Sie sich 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 während des Betriebs IDs finden Sie unter [Stack-Ereignisse nach Vorgang anzeigen](#).

- Bei Problemen mit Amazon EC2 sehen Sie sich die `cloud-init`- und `cfn`-Protokolle an. Diese Protokolle werden auf der Amazon EC2 Instance im Verzeichnis `/var/log/` 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 und `cfn %ProgramData%\Amazon\EC2Launch\log` 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-Managementkonsole sodass Sie keine Verbindung zu Ihrer Amazon EC2 EC2-Instance herstellen müssen. Weitere Informationen finden Sie im Anwendungsmanagement-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](#)
- [Warte-Bedingung erhielt nicht die erforderliche Anzahl an Signalen von einer Amazon EC2-Instance](#)
- [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. Beispiel: Sie müssen alle Objekte in einem Amazon S3-Bucket löschen oder alle Instances in einer Amazon EC2-Sicherheitsgruppe entfernen, 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 EC2 zu nutzen.
- Wenn sich Stacks im DELETE_FAILED Status befinden, weil eine Ressource nicht gelöscht werden konnte, führen Sie den [Löschvorgang](#) mit dem `RetainResources` Parameter erneut aus und geben Sie die Ressource an, die nicht gelöscht 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 auch die Option `FORCE_DELETE_STACK` mit dem Parameter `DeletionMode` verwenden. Weitere Informationen zum erzwungenen Löschen eines Stapels finden Sie unter [Einen Stack aus der CloudFormation Konsole löschen](#).

- 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 kollidieren.
- Suchen Sie in Ihrer CloudFormation Vorlage nach Ressourcen, die von AWS Config und Systems Manager verwaltet werden. Prüfen Sie, ob es Überschneidungen oder Abhängigkeiten gibt, und überlegen Sie, ob Sie die Vorlage oder die Automatisierungskonfiguration anpassen, 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 beizubehalten.

- 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-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 Amazon EC2-Instance starten, brauchen 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 Amazon EC2-Schlüsselpaar oder eine VPC-ID bestimmen, muss die Ressource im Konto und in der Region vorhanden sein, in der Sie den 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 [AWS Ressourcen- und Eigenschaftstypen](#).

Kontingent überschritten

Vergewissern Sie sich, dass Sie nicht den Ressourcenkontingent erreicht haben. Beispielsweise ist die standardmäßige maximale Anzahl von On-Demand-Instances von Amazon EC2, die Sie launchen können, 5. Wenn Sie versuchen, mehr On-Demand-Instances von Amazon EC2 als Ihr Konto-Limit zu erstellen, wird die Instance-Erstellung fehlschlagen und Sie erhalten den Fehler `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 die Bereinigung der Ressourcen des verschachtelten Stacks erst, CloudFormation wenn alle verschachtelten Stapel aktualisiert oder ein Rollback durchgeführt wurden. 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 CloudFormation werden jedoch nicht als Update erkannt, z. B. Änderungen an einer Löschrichtlinie, Aktualisierungsrichtlinie, Bedingungsdeklaration oder 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 [Metadaten-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 zum Beispiel Ihr Kontolimit für die Anzahl der On-Demand-Instances von EC2 5 ist und der Aktualisierungs-Rollback dieses Kontingent überschreitet, wird es fehlschlagen.

- 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 [ContinueUpdateRollback](#) API-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.

Warte-Bedingung erhielt nicht die erforderliche Anzahl an Signalen von einer Amazon EC2-Instance

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 Hilfskripts nicht hat, können Sie diese auch auf Ihre Instance runterladen. Weitere Informationen finden Sie in der [Referenz zu CloudFormation Hilfsskripten](#) im 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 CFN-Logs `C:\cfn\log` und EC2 Config Service Logins `%ProgramFiles%\Amazon\EC2ConfigService` einsehen.
- 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 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 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 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.
- Bei Amazon EC2-Problemen sammeln Sie die cloud-init- und cfn-Protokolle. Diese Protokolle werden auf der Amazon EC2 Instance im Verzeichnis /var/log/ veröffentlicht. Diese Protokolle erfassen Prozesse und Befehlsausgaben während Ihre Instance eingerichtet wird. Rufen Sie für Windows den EC2 Configure-Dienst auf und cfn meldet sich an und. %ProgramFiles%\Amazon\EC2ConfigService 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 betreiben. Informationen über Amazon Q Developer finden Sie im [Amazon-Q-Developer-Benutzerhandbuch](#).

In diesem Dokument wird beschrieben, wie Sie Amazon Q Developer verwenden, um Ihnen bei der Behebung fehlgeschlagener CloudFormation Bereitstellungen zu helfen. Dieses Feature soll Ihnen dabei helfen, Probleme, die während der Stack-Bereitstellung auftreten, schnell zu erkennen und zu beheben.

Angetrieben von Amazon Bedrock: Amazon Q basiert auf Amazon Bedrock und beinhaltet [automatische Missbrauchserkennung](#) implementiert in Amazon Bedrock, um die Sicherheit und den verantwortungsvollen Einsatz von KI zu gewährleisten.

Themen

- [Features](#)
- [Funktionsweise](#)
- [Preisgestaltung](#)
- [Regionale Verfügbarkeit](#)
- [Einschränkungen](#)

Features

Die Fehlerbehebung bei Stack-Implementierungen mit der Amazon Q Developer-Lösung bietet die folgenden Features:

- 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-Managementkonsole.

Sie können die Schaltfläche Diagnose mit Q neben der Fehlermeldung wählen.

Amazon Q Developer analysiert den Fehler und liefert eine Erklärung für das Problem.

Wenn Sie zusätzliche Hilfe benötigen, wählen Sie Hilfe bei der Lösung um detaillierte Lösungsschritte zu erhalten.

Preisgestaltung

Informationen zu den Preisen 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-Benutzerhandbuch.

Einschränkungen

- Dieses Feature ist nur über das Feature AWS-Managementkonsole zugänglich.
- Dieses Feature behandelt die 20 häufigsten Fehler. Sie 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, die Aktualisierungen des Inhalts der Vorlagenreferenz ab Mai 2018 beschreiben, wurden in die neue AWS CloudFormation Vorlagenreferenzverschoben. Diese Aktualisierungen und alle future Änderungen finden Sie im [Dokumentverlauf für die AWS CloudFormation Vorlagenreferenz](#) in der AWS CloudFormation Vorlagenreferenz.

Änderung	Beschreibung	Datum
PDF-Anleitung verfügbar	Sie können jetzt das AWS CloudFormation -Benutzer handbuch als PDF herunterladen.	30. Mai 2025
Inhalt der Vorlagenreferenz in eine neue Anleitung verschoben	CloudFormation hat den CloudFormation Vorlagenreferenzleitfaden veröffentlicht. Einzelheiten finden Sie im Referenzhandbuch für CloudFormation Vorlagen .	30. Mai 2025
IaC-Generator unterstützt partielles Scannen	Sie können jetzt bestimmte Ressourcentypen auswählen , nach denen gesucht werden soll, was das Generieren von Infrastructure-as-Code (IaC) -Vorlagen aus Ihren vorhandenen Ressourcen	27. März 2025

erleichtert. Weitere Informationen finden Sie unter [Starten eines Ressourcenscans mit dem CloudFormation IaC-Generator](#).

[Stapel-Refactoring](#)

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

[Graph der Zeitleiste für die Stack-Bereitstellung](#)

Sie sehen nun eine visuelle Darstellung Ihrer Stack-Bereitstellung. Das Zeitdiagramm der Stack-Bereitstellung zeigt den Status der Stack-Bereitstellung, die einzelnen Status der Ressourcenbereitstellung und die Zeiten, zu denen sich die Bereitstellungsstatus geändert haben. 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 Infrastruktur as Code (IaC)-Generator-Workflows optimieren, indem Sie die Details der Scan-Zusammenfassung visualisieren und die generierten Vorlagen in der Vorschau anzeigen, bevor Sie Ihren Infrastruktur-Stack bereitstellen. Weitere Informationen finden Sie unter [Scan-Zusammenfassung in der CloudFormation Konsole anzeigen und Einen CloudFormation Stapel aus gescannten Ressourcen erstellen](#).

22. August 2024

[EventBridge Amazon-Integration mit AWS CloudFormation Git-Sync](#)

AWS CloudFormation Git Sync veröffentlicht jetzt Änderungen des Synchronisierungssstatus als Ereignisse an Amazon EventBridge. Weitere Informationen finden Sie unter [Ereignisdetail Repository Sync-Statusänderung](#) und [Ereignisdetail Ressourcen sync-Statusänderung](#).

29. Juli 2024

[Erzwingen des Löschen festgefahrener Stapel](#)

Zwei neue Optionen zum Erzwingen des Löschens von Stapeln sind verfügbar, wenn die Stapel-Löschoperationen stecken bleiben. Sie können nun wählen, ob Sie das Löschen des Stapels erzwingen, aber die Ressource beibehalten wollen, oder ob Sie das Löschen des gesamten Stapels erzwingen wollen. Weitere Informationen findest du unter [Einen Stack aus der CloudFormation Konsole löschen](#).

22. Mai 2024

[AWS CloudTrail 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 Stapelfehlers](#).

15. Mai 2024

[Eigenschaftslevel-Änderungssätze](#)

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 jetzt das Ereignis CONFIGURATION_COMPLETE verwenden, um schnellere Arbeitsabläufe bei der Erstellung von Ressourcen zu ermöglichen. 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 [Erzeugen 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 abweichen und welche Ressourcen sich geändert haben.

24. Juli 2023

[ListStackInstanceResourceDrifts](#)

Gibt Abweichungsinformationen für Ressourcen in einer Stack-Instance zurück.

[StackInstanceResourceDriftsSummary](#)

Die Struktur, die zusammenfassende Informationen über Ressourcenabweichungen für eine Stack-Instance enthält.

[CloudFormation StackSets
APIs um den vertrauenswürdig
en Zugriff zu kontrollieren
AWS Organizations](#)

CloudFormation StackSets bietet Kunden Folgendes APIs für die Verwaltung des AWS Organizations Vertrauenszugriffs:

5. Juni 2023

[ActivateOrganizationsAccess](#)

Aktivieren Sie den vertrauenswürdigen Zugriff mit AWS Organizations. Wenn der vertrauenswürdige Zugriff zwischen StackSets und Organizations aktiviert ist, verfügt das Verwaltungskonto über Berechtigungen zum Erstellen und Verwalten StackSets für Ihre Organisation.

[DeactivateOrganizationsAccess](#)

Deaktiviert den vertrauenswürdigen Zugriff mit AWS Organizations. Wenn der vertrauenswürdige Zugriff deaktiviert ist, verfügt das Verwaltungskonto nicht über die erforderlichen Berechtigungen zum Erstellen und Verwalten von Service-Managed Services StackSets für Ihre Organisation.

DescribeOrganizationsAccess

Ruft Informationen über den Organisations-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 für die Liste der Regionen, in denen ein bestimmter Stacksatz eingesetzt 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 Informati
onen zu Stack-Instances für
Stack-Set-Operationen:

04. November 2022

[DescribeStackSetOperation](#)

Sie können DescribeS
tackSetOperation
jetzt verwenden, um die
Anzahl der fehlgesch
lagenen Stack-Instances
für Stack-Set-Operationen
während der Bereitstellung
anzugeben.

[ListStackInstances](#)

Sie können jetzt die
Filteroption LastOpera
tionID verwenden, um
Stack-Instances für Stack-
Set-Operationen aufzulist
en.

[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 servicemanged 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 [CloudFormation Hooks-Benutzerhandbuch](#).

10. Februar 2022

[Stack-Fehleroptionen](#)

Sie können Ihre Anwendung en 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 [Auswählen, wie mit Fehlern bei der Bereitstellung von Ressourcen umgegangen werden soll.](#)

30. August 2021

[Stack in Stack-Set importieren](#)

Sie können jetzt bestehende Stapel in neue oder bestehende Stapel-Sets importieren. Weitere Informationen finden Sie unter [Stacks importieren in.](#) CloudFormation StackSets

28. Juli 2021

Erhöhte Kontingente

Sie können jetzt ein standardmäßiges Maximum von 2000 Stapeln in Ihrem AWS-Konto angeben. Weitere Informationen finden Sie unter [Grundlegendes zu CloudFormation Kontingenten](#).

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 letzten Wert eines Systems Manager-Parameters in einer dynamischen Referenz](#)

Sie können jetzt immer die neueste Version eines Systems Manager Parameters CloudFormation verwenden, wenn Sie einen Stack erstellen oder aktualisieren. Sie müssen keine bestimmte Version mehr angeben. Weitere Einzelheiten finden Sie unter [Abrufen eines Klartextwerts aus dem Systems Manager Parameter Store](#).

13. April 2021

[Module unterstützen die Verwendung von Perioden-Trennzeichen in Ressourcenamen](#)

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

22. Oktober 2020

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 Zuordnungssattributen 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.

[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 [Ressourcentyp Unterstützung](#).

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 durch CodeDeploy](#)

Sie können es jetzt verwenden 19. Mai 2020

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](#)

[CloudFormation StackSets Verfügbarkeit in Regionen](#)

CloudFormation StackSets 18. Mai 2020

ist jetzt in der Region AWS GovCloud (USA West) verfügbar.

[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. 18. November 2019

[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.

[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 Verwaltung erstellt haben, können Sie diese vorhandene Ressource mit in die CloudFormation Verwaltung übernehmen `resource 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 Administratorkonto ausführen. Weitere Informationen finden Sie unter [Grundlegendes zu CloudFormation Kontingenten](#).

2. August 2019

Grenzwert für Ressourcen bei gleichzeitigen Stack-Vorgängen

CloudFormation erzwingt jetzt ein Kontolimit für die Anzahl der Ressourcen bei gleichzeitigen Stack-Vorgängen. Dieses Limit wird durch die Region bestimmt. Weitere Informationen finden Sie unter [Grundlegendes zu Kontingenten 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 jetzt die Funktion CAPABILITY_AUTO_EXPAND verwenden, um einen Stapel direkt aus einer Stapelvorlage, die Makros enthält, zu erstellen oder zu aktualisieren, ohne zuvor die daraus resultierenden Änderungen in einem Änderungssatz zu überprüfen. 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 sie darin CloudFormation definiert ist, abgewichen ist. Sie können die Drift eines gesamten Stapels oder einzelner Stapelressourcen erkennen. Weitere Informationen finden Sie unter [Erkennen von nicht verwalteten Konfigurationsänderungen an Stacks und Ressourcen mit Drifterkennung](#).

13. November 2018

[secretsmanager](#)
[dynamische Referenz jetzt](#)
[verfügbar](#)

Sie können nun die dynamische Referenz `secretsmanager` verwenden, um ganze Geheimnisse oder geheime Werte, die in AWS Secrets Manager gespeichert sind, abzurufen. 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 Geheimnisses oder eines geheimen Werts aus dem Secrets Manager](#).

9. November 2018

[Makros sind jetzt verfügbar](#)

Sie können jetzt Makros verwenden, um benutzerdefinierte Verarbeitungen an Vorlagen durchzuführen, von einfachen Aktionen wie find-and-replace Operationen bis hin zu umfangreichen Transformationen ganzer Vorlagen. Weitere Informationen finden Sie unter [Durchführen einer benutzerdefinierten Verarbeitung von CloudFormation Vorlagen mit Vorlagenmakros](#).

6. September 2018

[CloudFormation unterstützt jetzt VPC-Endpunkte, die betrieben werden von PrivateLink](#)

Sie können einen VPC-Endpunkt verwenden, um eine private Verbindung zwischen Ihrer VPC und CloudFormation herzustellen, ohne dass ein Zugriff über das Internet, über eine NAT-Instance, eine VPN-Verbindung oder Direct Connect erforderlich ist. Weitere Informationen finden Sie unter [Zugriff CloudFormation über einen Schnittstellenendpunkt \(AWS PrivateLink\)](#).

22. August 2018

[Dynamische Referenzen unterstützen sichere Zeichenfolgen](#)

Sie können jetzt neue dynamische Referenzen verwenden, um in Ihren Stack-Vorlagen Werte anzugeben, die in anderen Diensten gespeichert und verwaltet werden, einschließlich Parameter vom Typ Systems Manager Parameter Store SecureString. Weitere Informationen finden Sie unter [Abrufen eines sicheren Stringwerts aus dem Systems Manager Parameter Store](#).

16. August 2018

[Stack-Sets unterstützen jetzt benutzerdefinierte Ausführungsrollen.](#)

Sie können jetzt benutzerdefinierte Ausführungsrollen in Zielkonten verwenden, um die Stack-Ressourcen zu kontrollieren, die Benutzer oder Gruppen in ihre Stack-Sets aufnehmen können. Weitere Informationen finden Sie unter [Einrichten von erweiterten Berechtigungsoptionen für Stack-Set-Operationen](#).

30. Mai 2018

[Ausgewählte Aktualisierungen von Stack-Instances](#)

Sie können jetzt die optionalen Parameter Accounts und Regions verwenden, um die Konten und Regionen anzugeben, in denen die Stack-Instances während einer Stack-Set-Aktualisierungsoperation aktualisiert werden sollen. Weitere Informationen finden Sie unter [UpdateStackSet](#) in der 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.

[CloudFormation Informationen zu Endpunkten und Kontingen](#)
[ten finden Sie URLs im für den](#)
[neuen Allgemeine Amazon](#)
[Web Services-Referenz](#)FIPS-kompatiblen Endpunkt.

17. Mai 2018

Aktualisierungen des CloudFormation Hooks User Guide finden Sie im Hooks User Guide unter [Document History for the CloudFormation Hooks User Guide](#).CloudFormation

Archivierte Updates

In der folgenden Tabelle werden wichtige Änderungen in jeder Version des AWS CloudFormation Benutzerhandbuchs vor Mai 2018 beschrieben.

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	22. Juli 2019	Verwenden Sie die <code>encryptionOptions</code> -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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	10. April 2018	AWS::AppSync::ApiKey Verwenden Sie die <code>AWS::AppSync::ApiKey</code> Ressource, um einen eindeutigen Schlüssel zu erstellen, den Sie an Clients verteilen können, mit AWS AppSync den GraphQL-Operationen ausgeführt werden. AWS::AppSync::DataSource Verwenden Sie die <code>AWS::AppSync::DataSource</code> Ressource, um Datenquellen für Resolver in zu erstellen. AWS AppSync AWS::AppSync::GraphQLApi Verwenden Sie die <code>AWS::AppSync::GraphQLApi</code> Ressource, um eine neue AWS AppSync GraphQL-API zu erstellen. AWS::AppSync::GraphQLSchema Verwenden Sie die <code>AWS::AppSync::GraphQLSchema</code> Ressource, um das Datenmodell für Ihre AWS AppSync GraphQL-API zu erstellen. AWS::AppSync::Resolver Verwenden Sie die Ressource <code>AWS::AppSync::Resolver</code>, um den logischen GraphQL-Resolver zu definieren, den Sie an Felder in einem Schema anfügen.	15.05.2018

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressource	10. April 2018	AWS::Config::ConfigurationAggregator Verwenden Sie den <code>OrganizationAggregationSource</code> 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 <code>AWS::Config::AggregationAuthorization</code> Ressource, um einem Aggregatorkonto die Erlaubnis zur Erfassung Ihrer Daten zu erteilen. AWS Config AWS::Config::ConfigurationAggregator Verwenden Sie die <code>AWS::Config::ConfigurationAggregator</code> -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 Einrichten von erweiterten Berechtigungsoptionen für Stack-Set-Operationen.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressource	29. März 2018	AWS::EC2::LaunchTemplate Verwenden Sie die Ressource <code>AWS::EC2::LaunchTemplate</code> , um eine Einführungsvorlage für eine Amazon EC2-Instance zu erstellen.	15.05.2010
Aktualisierte Ressourcen	29. März 2018	AWS::AutoScaling::AutoScalingGroup Verwenden Sie die Eigenschaft <code>LaunchTemplate</code> , um die Einführungsvorlage für das Starten von Instances festzulegen. AWS::EC2::SpotFleet Verwenden Sie im Eigenschaftstyp SpotFleetRequestConfigData die Eigenschaft <code>LaunchTemplateConfigs</code> , um eine Startvorlage und Überschreibungen zu beschreiben.	15.05.2010
Neue intrinsische Funktion <code>Fn::Cidr</code>	6. März 2018	Gibt den angegebenen Cidr-Adressblock zurück. Weitere Informationen finden Sie unter Fn::Cidr .	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	6. März 2018	AWS::ApiGateway::VpcLink Verwenden Sie die Ressource <code>AWS::ApiGateway::VpcLink</code> zur Festlegung eines API Gateway VPC-Links für eine <code>AWS::ApiGateway::RestApi</code>, um auf Ressourcen in einer Amazon Virtual Private Cloud (VPC) zuzugreifen. AWS::GuardDuty::Master Verwenden Sie die <code>AWS::GuardDuty::Master</code> Ressource, um ein GuardDuty primäres Konto zu erstellen. AWS::GuardDuty::Member Verwenden Sie die <code>AWS::GuardDuty::Member</code> Ressource, um ein GuardDuty Mitgliedskonto zu erstellen. AWS::SES::ConfigurationSet Verwenden Sie die Ressource <code>AWS::SES::ConfigurationSet</code>, um Gruppen von Regeln zu erstellen, die Sie auf die E-Mails, die Sie senden, anwenden können. AWS::SES::ConfigurationSetEventDestination Mit der Ressource <code>AWS::SES::ConfigurationSetEventDestination</code> können Sie ein Konfigurationssatz-Ereignisziel angeben. AWS::SES::ReceiptFilter Mit der Ressource <code>AWS::SES::ReceiptFilter</code> können Sie angeben, ob E-Mails, die von einer IP-	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		Adresse oder einem IP-Adressbereich stammen, akzeptiert oder abgelehnt werden sollen. AWS::SES::ReceiptRule Mit der Ressource <code>AWS::SES::ReceiptRule</code> 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 <code>AWS::SES::ReceiptRuleSet</code> können Sie einen leeren Regelsatz für Amazon SES angeben. AWS::SES::Template Verwenden Sie die Ressource <code>AWS::SES::Template</code>, um den Inhalt der E-Mail anzugeben, bestehend aus einer Betreffzeile, einem HTML-Teil und einem Nur-Text-Teil.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	6. März 2018	AWS::AutoScaling::AutoScalingGroup Mit der Eigenschaft <code>AutoScalingGroupName</code> können Sie den Namen der Auto-Scaling-Gruppe angeben. AWS::ApiGateway::RestApi Mit der Eigenschaft <code>ApiKeySourceType</code> können Sie die Quelle des API-Schlüssels für Messungsa nforderungen nach einem Nutzungsplan angeben. Verwenden Sie die <code>MinimumCompressionSize</code> - 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 <code>DisableScaleIn</code> , um anzugeben, ob das Herunterskalieren durch die Richtlinie für die Ziel-Nachverfolgung deaktiviert ist. AWS::EC2::SpotFleet Verwenden Sie im LaunchSpecifications Eigenschaftstyp die <code>TagSpecifications</code> Eigenschaft, um die Tags anzugeben, die bei der SpotFleet Erstellung angewendet werden sollen. AWS::Elasticsearch::Domain Verwenden Sie das Attribut <code>Arn</code>, damit <code>Fn::GetAtt</code> den Amazon-Ressourcennamen (ARN) der Domain zurückgibt.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		Das Attribut <code>DomainArn</code> von <code>Fn::GetAtt</code> wurde als veraltet gekennzeichnet. AWS::RDS::DBCluster Mit der Eigenschaft <code>DBClusterIdentifier</code> können Sie die DB-Cluster-Kennung angeben. AWS::RDS::DBCluster Mit der Eigenschaft <code>DBClusterIdentifier</code> können Sie die DB-Cluster-Kennung angeben. AWS::Redshift::Cluster Mit der Eigenschaft <code>ClusterIdentifier</code> können Sie die eindeutige Kennung des Clusters angeben. AWS::Route53::HealthCheck Geben Sie im HealthCheckConfig Eigenschaftstyp mithilfe der <code>Regions</code> Eigenschaft die Regionen an, von denen aus Route 53-Integritätsprüfungen den angegebenen Endpunkt überprüfen sollen. AWS::SSM::Document Mit der Eigenschaft <code>Tags</code> können Sie die CloudFormation-Ressourcen-Tags für das Dokument angeben.	
Aktualisierte Ressource	19. Februar 2015	AWS::CodeBuild::Project Mit der Eigenschaft <code>Triggers</code> 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressource	8. Februar 2018	AWS::DynamoDB::Table Mit der Eigenschaft <code>SSESpecification</code> können Sie die Einstellungen zum Aktivieren der serverseitigen Verschlüsselung angeben.	15.05.2010
Aktualisierte Ressource	5. Februar 2018	AWS::CodeBuild::Project Im Eigenschaftstyp Source <code>CodeBuild Project Source</code> : • Mit der Eigenschaft <code>GitCloneDepth</code> können Sie die Tiefe des herunterzuladenden Verlaufs angeben. • Verwenden Sie die <code>InsecureSsl</code> Eigenschaft, um anzugeben, ob SSL-Warnungen ignoriert werden sollen, während Sie eine Verbindung zu Ihrem GitHub Enterprise-Projekt-Repository herstellen.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	23. Januar 2018	AWS::AutoScaling::LifecycleHook Mit der Eigenschaft <code>LifecycleHookName</code> können Sie den Namen des Lebenszyklus-Hook anzugeben. AWS::DynamoDB::Table Die Eigenschaft <code>AttributeDefinitions</code> muss beim Aktualisieren nun ersetzt werden. AWS::EC2::Instance Mit der Eigenschaft <code>CreditSpecification</code> können Sie die Guthaben-Option für die CPU-Nutzung einer T2-Instance angeben. Verwenden Sie die <code>ElasticGpuSpecifications</code> 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 <code>InstanceTenancy</code> ist nun keine Unterbrechung erforderlich, wenn sie von "dedicated" auf "default" aktualisiert wird. AWS::ECS::Service Mit der Eigenschaft <code>HealthCheckGracePeriodSeconds</code> 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.	15.05.2018

Änderungen	Veröffentlichungsdatum	Description	API-Version
		AWS::IoT::TopicRule Im Eigenschaftstyp DynamoDBAction werden die Eigenschaften <code>RangeKeyField</code> und <code>RangeKeyValue</code> nicht mehr benötigt. AWS::KinesisAnalytics::ApplicationOutput Im -Eigenschaftstyp ApplicationOutput können Sie beim Konfigurieren der Anwendungsausgabe mit der Eigenschaft <code>LambdaOutput</code> eine Lambda-Funktion als Ziel identifizieren. AWS::Kinesis::Stream Verwenden Sie die <code>StreamEncryption</code> 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 <code>ReservedConcurrentExecutions</code> können Sie die maximale Anzahl gleichzeitiger Ausführungen angeben, die für die Funktion reserviert werden sollen. AWS::RDS::DBSubnetGroup Mit der Eigenschaft <code>DBSubnetGroupName</code> können Sie den Namen für die DB-Subnetzgruppe anzugeben. AWS::S3::Bucket Verwenden Sie die <code>BucketEncryption</code> Eigenschaft, um die Standardverschlüsselung für einen Bucket anzugeben, der serverseitige Verschlüsselung mit	

Änderungen	Veröffentlichungsdatum	Description	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 <code>SourceSelectionCriteria</code> zusätzliche Filter bei der Identifizierung von Quellobjekten angeben, die repliziert werden sollen. Im Eigenschaftstyp ReplicationDestination: • Verwenden Sie die <code>AccessControlTranslation</code> Eigenschaft, um den Besitz des Replikats für das Konto anzugeben, dem der Ziel-Bucket gehört. AWS • Mit der Eigenschaft <code>Account</code> können Sie die Konto-ID des Ziel-Bucket-Eigentümers angeben. • Mit der Eigenschaft <code>EncryptionConfiguration</code> können Sie verschlüsselungsbezogene Informationen für einen Bucket angeben, der ein Ziel für replizierte Objekte ist. AWS::SSM::Association Mit der Eigenschaft <code>AssociationName</code> können Sie den Namen der Zuordnung zwischen einem SSM-Dokument und EC2-Instances angeben, die einen Konfigurationsagenten zur Verarbeitung des Dokuments enthalten.	

Änderungen	Veröffentlichungsdatum	Description	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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	5. Dezember 2017	AWS::Inspector::AssessmentTarget Mit der Ressource <code>AWS::Inspector::AssessmentTarget</code> können Sie ein Amazon Inspector-Bewertungsziel erstellen. AWS::Inspector::AssessmentTemplate Mit der Ressource <code>AWS::Inspector::AssessmentTemplate</code> können Sie eine Amazon Inspector-Bewertungsvorlage erstellen. AWS::Inspector::ResourceGroup Verwenden Sie die Ressource <code>AWS::Inspector::ResourceGroup</code>, 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 <code>AWS::ServiceDiscovery::Instance</code> können Sie Informationen zu einer Instance angeben, die von Amazon Route 53 erstellt wird. AWS::ServiceDiscovery::PrivateDnsNamespace Mit der Ressource <code>AWS::ServiceDiscovery::PrivateDnsNamespace</code> können Sie Informationen zu einem privaten Namespace für Amazon Route 53 angeben.	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		AWS::ServiceDiscovery::PublicDnsNamespace Mit der Ressource <code>AWS::ServiceDiscovery::PublicDnsNamespace</code> können Sie Informationen zu einem öffentlichen Namespace für Amazon Route 53 angeben. AWS::ServiceDiscovery::Service Sie können mit der Ressource <code>AWS::ServiceDiscovery::Service</code> 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 <code>InputProcessingConfiguration</code> Datensätze transformieren, wenn diese vom Stream empfangen werden.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressource	1. Dezember	AWS::CodeBuild::Project Mit der Eigenschaft <code>BadgeEnabled</code> können Sie eine öffentlich zugängliche URL für den Build-Auswahl eines Projekts erstellen. Mit der Eigenschaft <code>Cache</code> können Sie Cache-Einstellungen für Build-Abhängigkeiten konfigurieren. Verwenden Sie die <code>VpcConfig</code> Eigenschaft, um den Zugriff auf Ressourcen in einer Amazon VPC CodeBuild zu ermöglichen. Im Eigenschaftstyp EnvironmentVariable können Sie mit der Eigenschaft <code>Type</code> die Art der Umgebungsvariablen angeben.	15.05.2010
Neue Ressource	30. November 2017	AWS::Cloud9::EnvironmentEC2 Verwenden Sie die Ressource <code>AWS::Cloud9::EnvironmentEC2</code>, um eine Amazon-EC2-Entwicklungsumgebung in AWS Cloud9 zu erstellen.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	29. November 2017	AWS::ECS::TaskDefinition Mit der Eigenschaft <code>Cpu</code> können Sie die Anzahl der CPU-Einheiten angeben, die für die zu erledigende Aufgabe erforderlich sind. Mit der Eigenschaft <code>ExecutionRoleArn</code> können Sie den ARN der Ausführungsrolle angeben. Mit der Eigenschaft <code>Memory</code> können Sie die Menge (in MiB) an Speicher angeben, die für die zu erledigende Aufgabe erforderlich ist. Mit der Eigenschaft <code>RequiresCompatibilities</code> können Sie angeben, welcher Starttyp für die Aufgabe benötigt wird. AWS::ECS::Service Mit der Eigenschaft <code>LaunchType</code> können Sie den Starttyp angeben, auf dem Ihr Service ausgeführt werden soll. Mit der Eigenschaft <code>NetworkConfiguration</code> können Sie die Netzwerkkonfiguration für den Service angeben. Mit der Eigenschaft <code>PlatformVersion</code> können Sie die Version der Plattform angeben, auf der Ihr Service ausgeführt wird.	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
Neue Ressourcen	28. November 2017	AWS::GuardDuty::Detector Verwenden Sie die <code>AWS::GuardDuty::Detector</code> Ressource, um einen einzelnen GuardDuty Amazon-Detektor zu erstellen. AWS::GuardDuty::IPSet Verwenden Sie die <code>AWS::GuardDuty::IPSet</code> Ressource, um einen GuardDuty Amazon-IP-Satz zu erstellen. AWS::GuardDuty::ThreatIntelSet Verwenden Sie die <code>AWS::GuardDuty::ThreatIntelSet</code> Ressource, um eine zu erstellen ThreatIntelSet.	15.05.201 0

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	28. November 2017	AWS::CodeDeploy::Application Verwenden Sie die <code>ComputePlatform</code> Eigenschaft, um eine AWS Lambda Rechenplattform anzugeben <code>CodeDeploy</code>, auf der eine Anwendung bereitgestellt werden soll. AWS::CodeDeploy::DeploymentGroup Im Eigenschaftstyp DeploymentStyle können Sie mit der Eigenschaft <code>DeploymentType</code> eine Blau/Grün-Bereitstellung einer Lambda-EDV-Plattform bereitstellen. AWS::EC2::SpotFleet Im Eigenschaftstyp SpotFleetRequestConfigData ist die Eigenschaft <code>SpotPrice</code> nun optional. AWS::Lambda::Alias Verwenden Sie die <code>RoutingConfig</code> 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 <code>CodeDeployLambdaAliasUpdate</code> -Aktualisierungsrichtlinie	28. November 2017	Verwenden Sie die <code>CodeDeployLambdaAliasUpdate</code> Aktualisierungsrichtlinie, um eine <code>CodeDeploy</code> Bereitstellung durchzuführen, wenn sich die Version auf einer <code>AWS::Lambda::Alias</code> Ressource ändert. Weitere Informationen finden Sie unter UpdatePolicy Attribut .	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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 <code>SecureString</code> derzeit nicht. Weitere Informationen finden Sie unter SSM-Parametertypen .	15.05.2010
Neues <code>ResolvedValue</code> - Feld für Parameter - Datentyp	21. November 2017	Das Feld <code>ResolvedValue</code> 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 CloudFormation.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	20. November 2017	AWS::ApiGateway::ApiKey Verwenden Sie die <code>CustomerId</code> -Eigenschaft, um eine AWS -Marketplace-Kunden-ID anzugeben. Mit der Eigenschaft <code>GenerateDistinctId</code> können Sie angeben ob sich die Schlüssel-ID von dem erstellten API-Schlüsselwert unterscheidet. AWS::ApiGateway::Authorizer Mit der Eigenschaft <code>AuthType</code> können Sie ein benutzerdefiniertes Feld angeben, das in Swagger-Importen und -Exporten ohne funktionale Auswirkungen verwendet wird. AWS::ApiGateway::DomainName Mit der Eigenschaft <code>EndpointConfiguration</code> können Sie die Endpunkttypen eines API Gateway-Domainnamens angeben. Mit der Eigenschaft <code>RegionalCertificateArn</code> können Sie auf Zertifikat für die Verwendung durch den regionalen Endpunkt für einen Domainnamen verweisen. AWS::ApiGateway::Method In den Eigenschaftstypen Integration und IntegrationResponse verwenden Sie die Eigenschaft <code>ContentHandling</code> , um festzulegen, wie die Konvertierung des Inhaltstyps von Anforderungsnutzdaten behandelt werden soll.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		AWS::ApiGateway::RestApi Mit der Eigenschaft <code>EndpointConfiguration</code> können Sie die Endpunkttypen einer API Gateway-REST-API angeben. AWS::ApplicationAutoScaling::ScalableTarget Mit der Eigenschaft <code>ScheduledActions</code> können Sie geplante Aktionen für ein skalierbares Auto Scaling von Anwendungen-Ziel angeben. AWS::ECR::Repository Mit der Eigenschaft <code>LifecyclePolicy</code> können Sie eine Lebenszyklusrichtlinie für ein Amazon ECR-Repository angeben. AWS::ECS::TaskDefinition Im Eigenschaftstyp ContainerDefinition können Sie mit der Eigenschaft <code>LinuxParameters</code> Linux-spezifische Optionen für einen Amazon ECS-Container angeben. AWS::ElastiCache::ReplicationGroup Mit der Eigenschaft <code>AtRestEncryptionEnabled</code> können Sie die Verschlüsselung im Ruhezustand aktivieren. Mit der Eigenschaft <code>AuthToken</code> können Sie ein Passwort für den Zugriff auf einen passwortgeschützten Server angeben. Mit der Eigenschaft <code>TransitEncryptionEnabled</code> können Sie die Verschlüsselung während der Übertragung aktivieren.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
		AWS::ElasticLoadBalancingV2::TargetGroup Sie können das Attribut <code>TargetGroupName</code> mit der Funktion <code>Fn::GetAtt</code> verwenden, um den Namen einer Elastic Load Balancing Zielgruppe abzurufen. AWS::Elasticsearch::Domain Verwenden Sie die <code>VPCOptions</code> Eigenschaft, um eine VPC-Konfiguration für die OpenSearch Service-Domain anzugeben. AWS::EMR::Cluster Mit der Eigenschaft <code>EbsRootVolumeSize</code> können Sie die Größe des EBS-Stamm-Volumes für einen Amazon EMR-Cluster angeben. AWS::RDS::DBInstance Mit den Eigenschaften <code>SourceRegion</code> und <code>KmsKeyId</code> können Sie eine verschlüsselte Read Replica aus einer regionsübergreifenden Quell-DB-Instance erstellen. AWS::Route53::HostedZone Mit der Eigenschaft <code>QueryLoggingConfig</code> können Sie eine Konfiguration für die DNS-Abfrageprotokollierung angeben.	

Änderungen	Veröffentlichungsdatum	Description	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 <code>AWS::StepFunctions::StateMachine</code> 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 CloudFormation Limits finden Sie unter CloudFormationKontingente verstehen.	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
Neue Ressourcen	2. November 2017	AWS::CloudFront::CloudFrontOriginAccessIdentity Verwenden Sie die <code>AWS::CloudFront::CloudFrontOriginAccessIdentity</code> Ressource, um die CloudFront Amazon-Ursprungszugriffsidetität anzugeben, die dem Ursprung einer CloudFront Distribution zugeordnet werden soll. AWS::CloudFront::StreamingDistribution Verwenden Sie die <code>AWS::CloudFront::StreamingDistribution</code> Ressource, um eine Adobe Real-Time Messaging Protocol (RTMP) - Streaming-Verteilung für CloudFront anzugeben.	15.05.201 0

Änderungen	Veröffentlichungsdatum	Description	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 DefaultCacheBehavior Eigenschaftstypen CacheBehavior und die LambdaFunctionAssociations Eigenschaft, um Lambda-Funktionszuordnungen für eine CloudFront Verteilung anzugeben. Verwenden Sie im CustomOriginConfig-Eigenschaftstyp die OriginKeepaliveTimeout -Eigensch	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		ft, um ein benutzerdefiniertes Keepalive-Timeout anzugeben, und verwenden Sie die <code>OriginReadTimeout</code> -Eigenschaft, um ein Timeout für das Lesen des benutzerdefinierten Ursprungs anzugeben. Verwenden Sie im DistributionConfig Eigenschaftstyp die <code>IPv6Enabled</code> 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 <code>TargetGroupInfoList</code> -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 <code>Description</code> -Eigenschaft, um die Beschreibung einer Sicherheitsgruppenregel festzulegen. AWS::EC2::Subnet Die <code>Ipv6CidrBlock</code> -Eigenschaft unterstützt jetzt <code>No interruption</code> -Aktualisierungen. AWS::EC2::VPNGateway Verwenden Sie die <code>AmazonSideAsn</code> -Eigenschaft, um eine private Autonomous System Number (ASN) für die Amazon-Seite einer BGP-Sitzung anzugeben.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
		AWS::EC2::VPNConnection Verwenden Sie die <code>VpnTunnelOptionsSpecifications</code> -Eigenschaft, um Tunneloptionen für eine VPN-Verbindung zu konfigurieren. AWS::ElasticBeanstalk::ConfigurationTemplate und AWS::ElasticBeanstalk::Environment In den Eigenschaftstypen ConfigurationOptionSetting und OptionSetting verwenden Sie die Eigenschaft <code>ResourceName</code> , um einen Ressourcennamen für eine zeitbasierte Skalierungskonfigurationsoption anzugeben. AWS::EMR::Cluster Verwenden Sie die <code>CustomAmiId</code> -Eigenschaft, um ein benutzerdefiniertes Amazon Linux-AMI für ein Cluster anzugeben. AWS::KinesisFirehose::DeliveryStream Verwenden Sie das <code>Arn</code>-Attribut mit der <code>Fn::GetAtt</code> -Funktion, um den Amazon-Ressourcennamen (ARN) des Bereitstellungs-Streams abzurufen. AWS::KMS::Key Verwenden Sie die <code>Tags</code>-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 <code>Tags</code> Eigenschaft, um eine beliebige Gruppe von Tags (Schlüssel-Wert-Paare)	

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		anzugeben, die einer OpsWorks Ebene oder einem Stapel zugeordnet werden sollen. AWS::RDS::OptionGroup Verwenden Sie im OptionConfiguration-Eigenschaftstyp die <code>OptionVersion</code> -Eigenschaft, um eine Version für die Option anzugeben. AWS::S3::Bucket Verwenden Sie die <code>AnalyticsConfigurations</code> -Eigenschaft zum Konfigurieren eines Analysefilters für einen Amazon S3-Bucket.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	24. Oktober 2017	AWS::Glue::Classifier Verwenden Sie die <code>AWS::Glue::Classifier</code> - Ressource zum Erstellen eines AWS Glue -Classifiers. AWS::Glue::Connection Verwenden Sie die <code>AWS::Glue::Connection</code> Ressource, um eine AWS Glue Verbindung zu einer Datenquelle anzugeben. AWS::Glue::Crawler Mithilfe der <code>AWS::Glue::Crawler</code> -Ressource können Sie einen AWS Glue -Crawler festlegen. AWS::Glue::Database Verwenden Sie die <code>AWS::Glue::Database</code> - Ressource zum Erstellen einer AWS Glue -Datenbank. AWS::Glue::DevEndpoint Verwenden Sie die <code>AWS::Glue::DevEndpoint</code> - Ressource, um einen Entwicklungs-Endpunkt anzugeben, mit dem ETL-Skripts remote debuggt werden. AWS::Glue::Job Verwenden Sie die <code>AWS::Glue::Job</code> -Ressource, um einen AWS Glue -Job im Datenkatalog anzugeben.	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		AWS::Glue::Partition Verwenden Sie die <code>AWS::Glue::Partition</code> - Ressource, um eine AWS Glue -Partition zu erstellen, die für einen Slice der Tabellendaten steht. AWS::Glue::Table Verwenden Sie die <code>AWS::Glue::Table</code> -Ressource zum Erstellen einer AWS Glue -Tabelle. AWS::Glue::Trigger Verwenden Sie die <code>AWS::Glue::Trigger</code> Ressource, um Trigger anzugeben, die AWS Glue Jobs ausführen.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	11. Oktober 2017	AWS::SSM::MaintenanceWindow Verwenden Sie die <code>AWS::SSM::MaintenanceWindow</code> -Ressource, um ein AWS Systems Manager Wartungsfenster zu erstellen. AWS::SSM::MaintenanceWindowTarget Verwenden Sie die <code>AWS::SSM::MaintenanceWindowTarget</code> -Ressource zum Registrieren eines Ziels bei einem Wartungsfenster. AWS::SSM::MaintenanceWindowTask Verwenden Sie die <code>AWS::SSM::MaintenanceWindowTask</code> -Ressource zum Definieren einer Wartungsfenster-Aufgabe. AWS::SSM::PatchBaseline Verwenden Sie die <code>AWS::SSM::PatchBaseline</code> -Ressource zum Definieren einer Systems Manager-Patch-Baseline.	15.05.2010
Neue Ressource	10. Oktober 2017	AWS::ElasticLoadBalancingV2::ListenerCertificate Verwenden Sie die <code>AWS::ElasticLoadBalancingV2::ListenerCertificate</code> -Ressource zum Angeben von Zertifikaten für einen Elastic Load Balancing-Listener.	15.05.2010
Neue Ressource	27. September 2017	AWS::Athena::NamedQuery Verwenden Sie zum Erstellen einer Amazon Athena-Abfrage die <code>AWS::Athena::NamedQuery</code> -Ressource.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	27. September 2017	AWS::EC2::NatGateway Verwenden Sie die Eigenschaft <code>Tags</code>, um Ressourcen-Tags für ein NAT-Gateway anzugeben. AWS::ElasticBeanstalk::Application Verwenden Sie die Eigenschaft <code>ResourceLifecycleConfig</code>, um die Lebenszykluseinstellungen für Ressourcen zu definieren, die zur Anwendung gehören, sowie die <code>ServiceRole</code>, die Elastic Beanstalk unterstellt, um die Lebenszykluseinstellungen anzuwenden. AWS::ElasticBeanstalk::ConfigurationTemplate und AWS::ElasticBeanstalk::Environment Verwenden Sie die Eigenschaft <code>PlatformArn</code>, um eine benutzerdefinierte Plattform für Elastic Beanstalk anzugeben. AWS::ElasticLoadBalancingV2::TargetGroup Geben Sie im TargetDescription Eigenschaftstyp mithilfe der <code>AvailabilityZone</code> 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. • <code>EcsParameters</code> • <code>InputTransformer</code> • <code>KinesisParameters</code>	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		RunCommandParameters AWS::KinesisFirehose::DeliveryStream Verwenden Sie die <code>DeliveryStreamType</code> Eigenschaft, um den Streamtyp anzugeben, und die <code>KinesisStreamSourceConfiguration</code> 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 <code>Engine</code> den Wert <code>oracle-se</code> oder <code>oracle-se1</code> angegeben haben, können Sie auf <code>oracle-se2</code> aktualisieren, ohne dass die Datenbank-Instance ersetzt wird. AWS::S3::Bucket Verwenden Sie die Eigenschaft <code>AccelerateConfiguration</code>, 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Der Standardwert von umask wurde ab Version 1.4-22 geändert	14. September 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 CloudFormation -API-Referenz.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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 Ressource n 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	23. August 2017	AWS::S3::Bucket Verwenden Sie im Rule Eigenschaftstyp die <code>TagFilters</code> -Eigenschaft , um Tags anzugeben , die für die Identifizierung einer Untergruppe von Objekten für einen Amazon S3-Bucket verwendet werden. Verwenden Sie die <code>MetricsConfiguration</code> Eigenschaft, um eine Metrikkonfiguration für die CloudWatch Anforderungsmetriken aus einem Amazon S3 S3-Bucket anzugeben. AWS::IoT::TopicRule Verwenden Sie im Action Eigenschaftstyp die <code>DynamoDBv2Action</code> -Eigenschaft, um eine AWS IoT -Aktion zu beschreiben, die Daten in eine DynamoDB-Tabelle schreibt. Im Action-Eigenschaftstyp unterstützt die Eigenschaft <code>DynamoDBAction</code> jetzt die Eigenschaften <code>HashKeyType</code> und <code>RangeKeyType</code> . AWS::Lambda::Permission Verwenden Sie die <code>EventSourceToken</code> -Eigenschaft, um eindeutiges Token anzugeben, das von dem Prinzipal bereitgestellt werden muss, der die Funktion aufruft.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Pseudoparameter	23. August 2017	Verwenden Sie die <code>AWS::Partition</code> -Pseudo-Parameter, um die Partition zurückzugeben, in der sich eine Ressource befindet. Verwenden Sie den <code>AWS::URLSuffix</code> -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 <code>AWS::DAX::Cluster</code> -Ressource einen DAX-Cluster zur Verwendung mit Amazon DynamoDB. AWS::DAX::ParameterGroup Verwenden Sie die <code>AWS::DAX::ParameterGroup</code> -Ressource, um eine Parametergruppe zur Verwendung mit Amazon DynamoDB zu erstellen. AWS::DAX::SubnetGroup Verwenden Sie die <code>AWS::DAX::SubnetGroup</code> -Ressource, um eine Subnetzgruppe zur Verwendung mit DAX (DynamoDB Accelerator) zu erstellen.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	18. August 2017	AWS::ApiGateway::DocumentationPart und AWS::ApiGateway::DocumentationVersion -Ressourcen, um eine Dokumentation für Ihre API Gateway API zu erstellen. AWS::ApiGateway::GatewayResponse Verwenden Sie die <code>AWS::ApiGateway::GatewayResponse</code> -Ressource, um eine benutzerdefinierte Antwort für Ihre API Gateway API zu erstellen. AWS::ApiGateway::RequestValidator Verwenden Sie die <code>AWS::ApiGateway::RequestValidator</code> -Ressource, um Validierungsregeln für eingehende Anforderungen an Ihre API Gateway API einzurichten. AWS::EC2::NetworkInterfacePermission Verwenden Sie die <code>AWS::EC2::NetworkInterfacePermission</code> Ressource, um einem AWS Konto die Berechtigung für eine Netzwerkschnittstelle zu erteilen.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	18. August 2017	AWS::ApiGateway::Stage Verwenden Sie die <code>DocumentationVersion</code> -Eigenschaft, um einen versionierten Snapshot der API-Dokumentation anzugeben. AWS::AutoScaling::ScalingPolicy Verwenden Sie die <code>TargetTrackingConfiguration</code> -Eigenschaft, um eine Auto Scaling-Ziel-Tracking-Skalierungsrichtlinienkonfiguration anzugeben. AWS::CloudTrail::Trail Verwenden Sie die <code>EventSelectors</code> -Eigenschaft, um Amazon S3 Data Events zu unterstützen. AWS::CodeDeploy::DeploymentGroup Verwenden Sie die <code>LoadBalancerInfo</code> - und <code>DeploymentStyle</code> -Eigenschaften, um einen Elastic Load Balancing Load Balancer für eine In-Place-Bereitstellung anzugeben. Verwenden Sie die <code>AutoRollbackConfiguration</code> -Eigenschaft, um das automatische Zurücksetzen für die Bereitstellung zu konfigurieren. AWS::EC2::SpotFleet Verwenden Sie im SpotFleetRequestConfigData-Eigenschaftstyp die <code>ReplaceUnhealthyInstances</code> -Eigenschaft, um anzugeben, ob die Spot-Flotte fehlerhafte Instances ersetzen soll, und die <code>Type</code>-Eigenschaft, um den Anforderungstyp anzugeben.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		AWS::EC2::Subnet Verwenden Sie die <code>Ipv6CidrBlock</code> -Eigenschaften <code>AssignIpv6AddressOnCreation</code> und, um ein Subnetz mit einem IPv6 CIDR-Block zu erstellen. AWS::KinesisFirehose::DeliveryStream Verwenden Sie die <code>ExtendedS3DestinationConfiguration</code> -Eigenschaft, um in Amazon S3 ein Ziel zu konfigurieren. Verwenden Sie die <code>ProcessingConfiguration</code> -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 Der Standardwert <code>DeletionPolicy</code> ist jetzt <code>Snapshot</code> für <code>AWS::RDS::DBCluster</code> -Ressourcen und für <code>AWS::RDS::DBInstance</code> -Ressourcen, die nicht die Eigenschaft <code>DBClusterIdentifier</code> angeben. Weitere Informationen finden Sie unter DeletionPolicy Attribut. AWS::S3::Bucket Verwenden Sie im Rule-Eigenschaftstyp die <code>AbortIncompleteMultipartUpload</code> -Eigenschaft, um eine Lebenszyklusregel anzugeben, die unvollständige mehrteilige Uploads in ein Amazon S3-Bucket abbricht.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
		AWS::SQS::Queue Verwenden Sie die <code>KmsMasterKeyId</code> - und <code>KmsDataKeyReusePeriodSeconds</code> -Eigenschaften, eine serverseitige Verschlüsselung für Amazon SQS zu konfigurieren. Hinzufügen des <code>Arn</code>-Attributs zur intrinsischen Funktion <code>Fn::GetAtt</code> für die folgenden Ressourcen: • AWS::CloudTrail::Trail. Außerdem wurde <code>SnsTopicArn</code> 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Erstellen verschlüsselter Dateisysteme	14. August 2017	AWS::EFS::FileSystem Verwenden Sie die <code>Encrypted</code> -Eigenschaft zum Verschlüsseln eines Amazon EFS-Dateisystems während der Erstellung. Verwenden Sie die <code>KmsKeyId</code>-Eigenschaft, um optional einen benutzerdefinierten kundenverwalteten Schlüssel anzugeben, um das verschlüsselte Dateisystem zu schützen.	15.05.2010
Neue Ressourcen für den AWS Batch Support	8. August 2017	AWS::Batch::ComputeEnvironment Verwenden Sie die <code>AWS::Batch::ComputeEnvironment</code> Ressource, um Ihre AWS Batch Rechenumgebung zu definieren. AWS::Batch::JobDefinition Verwenden Sie die <code>AWS::Batch::JobDefinition</code> Ressource, um die Parameter für eine AWS Batch Jobdefinition anzugeben. AWS::Batch::JobQueue Verwenden Sie die <code>AWS::Batch::JobQueue</code> Ressource, um Ihre AWS Batch Job-Warteschlange zu definieren.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen für den Support von Amazon Managed Service für Apache Flink	28. Juli 2017	AWS::KinesisAnalytics::Application Verwenden Sie die Ressource <code>AWS::KinesisAnalytics::Application</code>, um eine Amazon-Managed-Service-für-Apache-Flink-Anwendung zu erstellen. AWS::KinesisAnalytics::ApplicationOutput Verwenden Sie die Ressource <code>AWS::KinesisAnalytics::ApplicationOutput</code>, um Ihrer Amazon-Managed-Service-für-Apache-Flink-Anwendung ein externes Ziel hinzuzufügen. AWS::KinesisAnalytics::ApplicationReferenceDataSource Verwenden Sie die Ressource <code>AWS::KinesisAnalytics::ApplicationReferenceDataSource</code>, um einer bestehenden Amazon-Managed-Service-für-Apache-Flink-Anwendung eine Referenzdatenquelle hinzuzufügen.	15.05.2010
Verwenden Sie StackSets, 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Anzeigen von Stack-Ereignissen durch Client-Anforderungstoken	14. Juli 2017	In der Konsole zeigen Stack-Operationen das Client-Anforderungstoken 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-Managementkonsole und StackEvent in 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen für den AWS Database Migration Service Support	12. Juli 2017	AWS::DMS::Certificate Verwenden Sie die <code>AWS::DMS::Certificate</code> Ressource, um ein SSL-Zertifikat zu erstellen, das Verbindungen zwischen AWS DMS Endpunkten und der Replikationsinstanz verschlüsselt. AWS::DMS::Endpoint Verwenden Sie die <code>AWS::DMS::Endpoint</code> -Ressource zum Erstellen eines AWS DMS -Endpunkts. AWS::DMS::EventSubscription Verwenden Sie die <code>AWS::DMS::EventSubscription</code> Ressource, um Benachrichtigungen über AWS DMS Ereignisse über den Amazon Simple Notification Service zu erhalten. AWS::DMS::ReplicationInstance Verwenden Sie die <code>AWS::DMS::ReplicationInstance</code> Ressource, um eine AWS DMS Replikationsinstanz zu erstellen. AWS::DMS::ReplicationSubnetGroup Verwenden Sie die <code>AWS::DMS::ReplicationSubnetGroup</code> Ressource, um eine AWS DMS Replikationssubnetzgruppe zu erstellen. AWS::DMS::ReplicationTask Verwenden Sie die <code>AWS::DMS::ReplicationTask</code> Ressource, um eine AWS DMS Replikationsaufgabe zu erstellen.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	5. Juli 2017	AWS::CloudWatch::Dashboard Verwenden Sie die <code>AWS::CloudWatch::Dashboard</code> -Ressource, um ein benutzerdefiniertes CloudWatch Dashboard für Ihre CloudWatch Konsole anzugeben. AWS::ApiGateway::DomainName Verwenden Sie die <code>AWS::ApiGateway::DomainName</code> -Ressource, um einen benutzerdefinierten URL-Anzeigenamen für Ihre API anzugeben, der API Amazon API Gateway bereitgestellt wird. AWS::EC2::EgressOnlyInternetGateway Verwenden Sie die <code>AWS::EC2::EgressOnlyInternetGateway</code> -Ressource, um ein Internet-Gateway nur für ausgehenden Datenverkehr für Ihre VPC zu erstellen. InstanceFleetConfig Verwenden Sie die <code>InstanceFleetConfig</code> -Ressource, um eine Spot-Instance-Flotte für einen Amazon-EMR-Cluster zu konfigurieren.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	5. Juli 2017	AWS::ApiGateway::RestApi Verwenden Sie die <code>BinaryMediaTypes</code> -Eigenschaft, um unterstützte binäre Medientypen anzugeben. AWS::ApplicationAutoScaling::ScalingPolicy Verwenden Sie die <code>TargetTrackingScalingPolicyConfiguration</code> -Eigenschaft, um eine Skalierungsrichtlinien-Konfiguration für die Ziel-Nachverfolgung anzugeben. AWS::CloudTrail::Trail Verwenden Sie die <code>TrailName</code> Eigenschaft, um einen benutzerdefinierten Namen für eine AWS CloudTrail Ressource anzugeben. Verwenden Sie die <code>Tags</code>-Eigenschaft, um Ressourcen-Tags anzugeben. AWS::CodeDeploy::DeploymentGroup Verwenden Sie die <code>AlarmConfiguration</code> -Eigenschaft, um Alarme für die Bereitstellungsgruppe zu konfigurieren. Verwenden Sie die <code>TriggerConfigurations</code> -Eigenschaft, um Benachrichtigungsauslöser für die Bereitstellungsgruppe zu konfigurieren. AWS::EMR::Cluster Verwenden Sie die <code>CoreInstanceFleet</code> Eigenschaft und die <code>MasterInstanceFleet</code> Eigenschaft im Eigenschaftstyp, um die JobFlowInstancesConfig Spot-Instance-Flotte für einen Amazon EMR-Cluster zu konfigurieren.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		AWS::DynamoDB::Table Verwenden Sie die <code>TimeToLiveSpecification</code> -Eigenschaft, um die TTL-Einstellungen (Time to Live) für eine Amazon DynamoDB-Tabelle anzugeben. Verwenden Sie die <code>Tags</code>-Eigenschaft, um Ressourcen-Tags für eine DynamoDB-Tabelle anzugeben. AWS::EC2::Instance Die <code>IamInstanceProfile</code> -Eigenschaft unterstützt jetzt <code>No interruption</code> -Aktualisierungen. AWS::EC2::Route Verwenden Sie die <code>EgressOnlyInternetGatewayId</code> -Eigenschaft, um ein Internet-Gateway nur für eingehenden Verkehr für eine EC2-Route anzugeben. AWS::Kinesis::Stream Verwenden Sie die <code>RetentionPeriodHours</code> -Eigenschaft, um anzugeben, wie viele Stunden lang die in Shards gespeicherten Datensätze zur Verfügung stehen. AWS::RDS::DBCluster Verwenden Sie die <code>ReplicationSourceIdentifier</code> -Eigenschaft, um ein DB-Cluster als Read Replica eines anderen DB-Clusters oder einer Amazon RDS MySQL-DB-Instance zu erstellen.	

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		AWS::Redshift::Cluster Verwenden Sie die <code>LoggingProperties</code> -Eigenschaft, um Audit-Protokolldateien zu erstellen und in Amazon S3 zu speichern.	
Neue Ressourcen	6. Juni 2017	AWS::EMR::SecurityConfiguration Verwenden Sie die <code>AWS::EMR::SecurityConfiguration</code> -Ressource zum Erstellen einer Sicherheitskonfiguration, die im Service gespeichert wird und angegeben werden kann, wenn ein Cluster erstellt wird.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	6. Juni 2017	AWS::AutoScaling::LifecycleHook Die Eigenschaften <code>NotificationTargetARN</code> und <code>RoleARN</code> sind jetzt optional. AWS::CloudWatch::Alarm Sie können jetzt die Eigenschaften <code>EvaluateLowSampleCountPercentile</code>, <code>ExtendedStatistic</code> und <code>TreatMissingData</code> verwenden, wenn Sie <code>AWS::CloudWatch::Alarm</code>-Ressourcen erstellen. AWS::EC2::SpotFleet CloudFormation unterstützt veränderbare Änderungen an den Eigenschaften der Spot-Flotte. Die folgenden Eigenschaften der <code>SpotFleetRequestConfigData</code>-Eigenschaft unterstützen <code>Replacement</code>-Aktualisierungen: • <code>AllocationStrategy</code> • <code>IamFleetRole</code> • <code>LaunchSpecifications</code> • <code>SpotPrice</code> • <code>TerminateInstancesWithExpiration</code> • <code>ValidFrom</code> • <code>ValidUntil</code> Die folgenden Eigenschaften der <code>SpotFleetRequestConfigData</code>-Eigenschaft unterstützen <code>No interruption</code>-Aktualisierungen: • <code>ExcessCapacityTerminationPolicy</code> • <code>TargetCapacity</code>	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		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.	

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		AWS::S3::Bucket Das Arn-Attribut wurde zur intrinsischen Fn::GetAtt-Funktion hinzugefügt.	

Änderungen	Veröffentlichungsdatum	Description	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 <code>AWS::WAFRegional::ByteMatchSet</code> -Ressource zum Identifizieren eines Teils einer Webanforderung, die Sie prüfen möchten. AWS::WAFRegional::IPSet Verwenden Sie die <code>AWS::WAFRegional::IPSet</code> -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 <code>AWS::WAFRegional::Rule</code> -Ressource, um eine Kombination von <code>IPSet</code>-, <code>ByteMatchSet</code> - und <code>SqliInjectionMatchSet</code> -Objekten anzugeben, welche die Webanforderungen zum Zulassen, Blockieren oder Zählen identifizieren. AWS::WAFRegional::SizeConstraintSet Verwenden Sie die <code>AWS::WAFRegional::SizeConstraintSet</code> -Ressource, um eine Größenbeschränkung zum Prüfen der Größe einer Webanforderung sowie die zu prüfenden Teile der Anforderung anzugeben.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		<u>AWS::WAFRegional::SqlInjectionMatchSet</u> Verwenden Sie die <code>AWS::WAFRegional::SqlInjectionMatchSet</code> -Ressource, um Anforderungen zuzulassen, zu blockieren oder zu zählen, die schädlichen SQL-Code in einem bestimmten Teil der Webanforderungen enthalten. <u>AWS::WAFRegional::WebACL</u> Verwenden Sie die <code>AWS::WAFRegional::WebACL</code> -Ressource zum Identifizieren der Webanforderungen, die Sie zulassen, blockieren oder zählen möchten. <u>AWS::WAFRegional::WebACLAssociation</u> Verwenden Sie die <code>AWS::WAFRegional::WebACLAssociation</code> -Ressource, um eine Web-Zugriffskontrollgruppe (ACL) mit einer Ressource zu verknüpfen. <u>AWS::WAFRegional::XssMatchSet</u> Verwenden die <code>AWS::WAFRegional::XssMatchSet</code> -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.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	28. April 2017	AWS::Cognito::IdentityPool Verwenden Sie die <code>AWS::Cognito::IdentityPool</code> -Ressource zum Erstellen eines Amazon Cognito-Identitäten-Pools. AWS::Cognito::IdentityPoolRoleAttachment Verwenden Sie die <code>AWS::Cognito::IdentityPoolRoleAttachment</code> -Ressource zum Verwalten der Rollenkonfiguration für einen Amazon Cognito Identitäten-Pool. AWS::Cognito::UserPool Verwenden Sie die <code>AWS::Cognito::UserPool</code> -Ressource zum Erstellen eines Amazon Cognito-Benutzerpools. AWS::Cognito::UserPoolClient Verwenden Sie <code>AWS::Cognito::UserPoolClient</code> -Ressource zum Erstellen eines Benutzerpool-Clients. AWS::Cognito::UserPoolGroup Verwenden Sie die <code>AWS::Cognito::UserPoolGroup</code> -Ressource zum Erstellen einer Benutzergruppe in einem Amazon Cognito-Benutzerpool. AWS::Cognito::UserPoolUser Verwenden Sie die <code>AWS::Cognito::UserPoolUser</code> -Ressource zum Erstellen eines Amazon Cognito-Benutzerpool-Benutzers.	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		<u>AWS::Cognito::UserPoolUserToGroupAttachment</u> Verwenden Sie die <code>AWS::Cognito::UserPoolUserToGroupAttachment</code>-Ressource, um einen Benutzer an eine Amazon Cognito-Benutzerpool-Gruppe anzuhängen.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	28. April 2017	SourceDetails Verwenden Sie die <code>MaximumExecutionFrequency</code> -Untereigenschaft der <code>AWS::Config::ConfigRule</code> -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 <code>No interruption</code> -Aktualisierungen für drei Eigenschaften: <code>VolumeType</code>, <code>Size</code> und <code>Iops</code>. AWS::EC2::SecurityGroup Verwenden Sie die <code>GroupName</code> -Eigenschaft, um einen Namen für Ihre Amazon EC2-Sicherheitsgruppe anzugeben. AWS::ECS::Service Es gibt drei neue Eigenschaften für <code>AWS::ECS::Service</code>: <code>PlacementConstraints</code>, <code>PlacementStrategies</code> und <code>ServiceName</code>. AWS::ECS::TaskDefinition Verwenden Sie die <code>PlacementConstraints</code> -Eigenschaft zum Definieren von Platzierungseinschränkungen für Aufgaben im Service. AWS::ElastiCache::ReplicationGroup Das <code>ConfigurationEndpoint.Address</code> -Attribut und das <code>ConfigurationEndpoint</code>	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		<code>int.Port</code> -Attribut wurden zur intrinsischen <code>Fn::GetAtt</code> -Funktion hinzugefügt. AWS::ElasticLoadBalancingV2::LoadBalancer Verwenden Sie die <code>IpAddressType</code> -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 <code>ManagedPolicyName</code> -Eigenschaft, um einen benutzerdefinierten Namen für Ihre verwaltete IAM-Richtlinie anzugeben. AWS::Lambda::Function Verwenden Sie die <code>Tags</code>-Eigenschaft zum Hinzufügen von Tags zu Ihrer Lambda-Funktion. AWS::OpsWorks::Instance Die folgenden Attribute wurden zur intrinsischen <code>Fn::GetAtt</code> -Funktion hinzugefügt: <code>AvailabilityZone</code> , <code>PrivateDnsName</code> , <code>PrivateIp</code> und <code>PublicDnsName</code> . AWS::OpsWorks::UserProfile Verwenden Sie die <code>SshUsername</code> -Eigenschaft, um den SSH-Namen eines Benutzers anzugeben. Das <code>SshUsername</code> -Attribut wurde zur intrinsischen <code>Fn::GetAtt</code> -Funktion hinzugefügt.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
		AWS::Redshift::Cluster Verwenden Sie die <code>IamRoles</code> 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 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 <code>AWS::SSM::Parameter</code> - Ressource zum Erstellen eines SSM-Parameters im Parameter Store.	15.05.2010
AWS::Include - Transformation	28. März 2017	Verwenden Sie die <code>AWS::Include</code> -Transformation, um wiederverwendbare, in einem Amazon S3-Bucket gespeicherte Codeausschnitte wiederzuverwenden. Weitere Informationen finden Sie unter AWS::Include-Transformation .	15.05.2010
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

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
Neue Ressource	28. März 2017	AWS::ApiGateway::UsagePlanKey Verwenden Sie die <code>AWS::ApiGateway::UsagePlanKey</code> -Ressource, um einen Nutzungsp lanschlüssel zu verknüpfen und festzulegen, auf welche Benutzer der Nutzungsplan angewendet wird.	15.05.201 0

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	28. März 2017	AWS::EC2::VPCPeeringConnection Verwenden Sie die <code>PeerOwnerId</code> -Eigenschaft und die <code>PeerRoleArn</code> -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 <code>InstanceProfileName</code> -Eigenschaft zum Konfigurieren eines Instance-Profils. AWS::Lambda::Function Verwenden Sie die <code>DeadLetterConfig</code> -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: <code>AlarmIdentifier</code> , <code>ChildHealthChecks</code> , <code>EnableSNI</code> , <code>HealthThreshold</code> , <code>InsufficientDataHealthStatus</code> , <code>Inverted</code> und <code>MeasureLatency</code> . AWS::SQS::Queue Verwenden Sie die <code>FifoQueue</code> Eigenschaften <code>ContentBasedDeduplication</code> und, um First-In-First-Out (FIFO) Amazon Simple Queue Service-Warteschlangen zu erstellen.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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 <code>AWS::StepFunctions::Activity</code> Ressource, um eine AWS Step Functions Aktivität zu erstellen. AWS::StepFunctions::StateMachine Verwenden Sie die <code>AWS::StepFunctions::StateMachine</code> -Ressource zum Erstellen eines Step Functions-Zustandsautomaten.	15.05.2010
Neue intrinsische Funktion	17. Januar 2017	Verwenden Sie die <code>Fn::Split</code> -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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	17. Januar 2017	AWS::AutoScaling::AutoScalingGroup Die <code>LoadBalancerNames</code> -Eigenschaft kann aktualisiert werden, ohne die Auto-Scaling-Gruppe zu ersetzen. AWS::ECS::TaskDefinition Die Eigenschaften <code>NetworkMode</code> und <code>MemoryReservation</code> wurden hinzugefügt. AWS::RDS::DBCluster CloudFormation unterstützt Aktualisierungen der <code>Tags</code> Eigenschaft. AWS::RDS::DBInstance Die <code>Timezone</code>-Eigenschaft wurde hinzugefügt. FirehoseAction Die <code>Separator</code> -Eigenschaft wurde hinzugefügt. AWS::OpsWorks::Instance Das <code>PublicIp</code>-Attribut für die intrinsische <code>Fn::GetAtt</code> -Funktion wurde hinzugefügt.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	01. Dezember 2016	AWS::CodeBuild::Project Verwenden Sie die <code>AWS::CodeBuild::Project</code> Ressource, um ein AWS CodeBuild Projekt zu erstellen, das definiert, wie Ihr Quellcode CodeBuild erstellt wird. AWS::SSM::Association Verwenden Sie die <code>AWS::SSM::Association</code> - Ressource, um ein Amazon EC2 Systems Manager-Dokument mit EC2-Instances zu verknüpfen. AWS::EC2::SubnetCidrBlock Verwenden Sie die <code>AWS::EC2::SubnetCidrBlock</code> Ressource, um einen einzelnen IPv6 CIDR-Block einem Amazon VPC-Subnetz zuzuordnen. AWS::EC2::VPCCidrBlock Verwenden Sie die <code>AWS::EC2::VPCCidrBlock</code> Ressource, um einen einzelnen von Amazon bereitgestellten IPv6 CIDR-Block einer Amazon VPC zuzuordnen.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Die Ressourcen für den Support wurden aktualisiert IPv6	01. Dezember 2016	AWS::EC2::Instance Die Eigenschaften <code>Ipv6AddressCount</code> und <code>Ipv6Addresses</code> wurden hinzugefügt. AWS::EC2::NetworkAclEntry Die <code>Ipv6CidrBlock</code> -Eigenschaft wurde hinzugefügt. AWS::EC2::NetworkInterface Die Eigenschaften <code>Ipv6AddressCount</code> und <code>Ipv6Addresses</code> wurden hinzugefügt. AWS::EC2::Route Die <code>DestinationIpv6CidrBlock</code> -Eigenschaft wurde hinzugefügt. AWS::EC2::SecurityGroupEgress Die <code>CidrIpv6</code>-Eigenschaft wurde hinzugefügt. AWS::EC2::SecurityGroupIngress Die <code>CidrIpv6</code>-Eigenschaft wurde hinzugefügt. AWS::EC2::SpotFleet Die Eigenschaften <code>Ipv6AddressCount</code> und <code>Ipv6Addresses</code> für die Startspezifikations-Netzwerkschnittstellen wurden hinzugefügt. AWS::EC2::Subnet Das <code>Ipv6CidrBlocks</code> -Attribut für die <code>Fn::GetAtt</code> -Funktion wurde hinzugefügt.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		AWS::EC2::VPC Das <code>Ipv6CidrBlocks</code> -Attribut für die <code>Fn::GetAtt</code> -Funktion wurde hinzugefügt. AWS::SSM::Document Die <code>DocumentType</code> -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 <code>AWS::OpsWorks::UserProfile</code> Ressource, um den SSH-Zugriff für Benutzer zu konfigurieren, die Zugriff auf Instanzen in einem OpsWorks Stack benötigen. AWS::OpsWorks::Volume Verwenden Sie die <code>AWS::OpsWorks::Volume</code> Ressource, um ein Amazon Elastic Block Store-Volumen bei einem OpsWorks Stack zu registrieren.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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

Änderungen	Veröffentlichungsdatum	Description	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 <code>AWS::SNS::Subscription</code> - 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 <code>Environment</code> Eigenschaft, um Schlüssel-Wert-Paare (Umgebungsvariablen) anzugeben, auf die Ihre AWS Lambda Funktion zugreifen kann. Verwenden Sie die <code>KmsKeyArn</code> Eigenschaft, um einen KMS-Schlüssel anzugeben, der zum Verschlüsseln und Entschlüsseln von Umgebungsvariablen AWS Lambda verwendet wird.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue CLI-Befehle	17. November 2016	Hochladen lokaler Artefakte in einen S3-Bucket Verwenden Sie den <code>package</code> 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 <code>deploy</code>-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 <code>HttpVersion</code> Eigenschaft, um die neueste HTTP-Version anzugeben, die Zuschauer für die Kommunikation mit Amazon verwenden können CloudFront. Verwenden Sie für die ForwardedValues<code>QueryStringCacheKeys</code> 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

Änderungen	Veröffentlichungsdatum	Description	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

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
Neue Ressourcen	06. Oktober 2016	AWS::ApiGateway::UsagePlan Verwenden Sie die <code>AWS::ApiGateway::UsagePlan</code> Ressource, um einen Nutzungsplan für das bereitgestellte Amazon API Gateway anzugeben APIs. AWS::CodeCommit::Repository Verwenden Sie die <code>AWS::CodeCommit::Repository</code> Ressource, um ein CodeCommit Repository zu erstellen, das von Amazon Web Services gehostet wird.	15.05.201 0

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	06. Oktober 2016	AWS::ApiGateway::Authorizer Verwenden Sie die <code>ProviderARNs</code> -Eigenschaft, um Amazon Cognito-Benutzerpools als Amazon API Gateway-API-Genehmiger zu verwenden. AWS::ApiGateway::Deployment Die <code>StageName</code> -Eigenschaft ist nicht mehr erforderlich. AWS::ElasticLoadBalancingV2::TargetGroup Verwenden Sie für die <code>GetAtt</code> Funktion das <code>LoadBalancerArns</code> Attribut, um die Amazon-Ressourcennamen (ARNs) der Load Balancer abzurufen, die den Verkehr an die Zielgruppe weiterleiten. AWS::RDS::DBInstance Verwenden Sie die Eigenschaften <code>Domain</code> und <code>DomainIAMRoleName</code>, um Windows-Authentifizierung zu verwenden, wenn Benutzer eine Verbindung zu der RDS DB-Instance herstellen. AWS::EC2::SecurityGroupEgress Verwenden Sie die <code>DestinationPrefixListId</code> 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

Änderungen	Veröffentlichungsdatum	Description	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 CloudFormation Dienstrolle .	15.05.2010
Neues Feature	19. September 2016	Sie können das Export-Ausgabefeld und die intrinsische <code>Fn::ImportValue</code> -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 Stack. 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 <code>Fn::Sub</code> -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 <code>AWS::KMS::Alias</code> , um für den AWS KMS key einen Alias zu erstellen.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	19. September 2016	AWS::EC2::SpotFleet Verwenden Sie für die <code>LaunchSpecifications</code> -Eigenschaft die <code>SpotPrice</code> -Eigenschaft, um einen Angebotspreis für einen bestimmten Instance-Typ anzugeben. AWS::ECS::Cluster Verwenden Sie die <code>ClusterName</code> -Eigenschaft, um einen Namen für einen Amazon Elastic Container Service-Cluster anzugeben. AWS::ECS::TaskDefinition Verwenden Sie die <code>TaskRoleArn</code> 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 <code>Family</code>-Eigenschaft für die Registrierung einer Aufgabendefinition an eine bestimmte Familie. AWS::Elasticsearch::Domain Verwenden Sie die <code>ElasticsearchVersion</code> -Eigenschaft, um anzugeben, welche Version von Elasticsearch verwendet werden soll.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	11. August 2016	Verwenden Sie die folgenden Elastic Load Balancing Application Load Balancer-Ressourcen, um den eingehenden Anwendungsdatenverkehr an mehrere Ziele zu verteilen, z. B. an EC2-Instances in mehreren Availability Zones: • 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

Änderungen	Veröffentlichungsdatum	Description	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 <code>ViewerCertificate</code> -Eigenschaft der Verteilungskonfiguration können Sie ein AWS Certificate Manager -Zertifikat angeben. Für die <code>Origin</code>-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

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
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	15.05.201 0

Änderungen	Veröffentlichungsdatum	Description	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 Attribut.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressource	25. April 2016	Verwenden Sie die Ressource AWS::EC2::Host , um einen vollständig dedizierten physischen Server für den Start von EC2-Instances zuzuweisen.	15.05.2010
Aktualisierte Ressourcen	25. April 2016	AWS::EC2::Instance Verwenden Sie die Eigenschaften <code>Affinity</code> und <code>HostId</code>, um Instances auf einem Dedicated Host von Amazon Elastic Compute Cloud zu starten. AWS::ECS::Service Verwenden Sie die <code>DeploymentConfiguration</code>-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 <code>MinSize</code> Eigenschaften <code>MaxSize</code> 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 <code>FunctionName</code> Eigenschaft, um einen Namen für Ihre AWS Lambda Funktion anzugeben. Sie können auch Python 2.7 verwenden, um eine Inlinefunktion anzugeben.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	18. April 2016	Amazon API Gateway Verwenden Sie die Amazon API Gateway-Ressourcen zum Veröffentlichen, Warten und Überwachen von APIs in jeder Größenordnung. Sie können APIs erstellen, die Clients für den Zugriff auf Ihre Back-End-Services aufrufen können, z. B. Anwendungen, die EC2-Instances ausführen oder Code, der auf 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 Simple Notification Service-Themen, und aktiviert diese.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		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::Alias Ressource, um Aliase für Ihre AWS Lambda Funktionen zu erstellen, und die AWS::Lambda::Version Ressource, um Versionen Ihrer Funktionen zu erstellen.	15.05.2010
Aktualisierte Ressourcen	31. März 2016	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::EMR::Cluster und AWS::EMR::InstanceGroupConfig Verwenden Sie die <code>EbsConfiguration</code> -Eigenschaft zum Konfigurieren von Amazon Elastic Block Store-Speicher-Volumes für Ihre Amazon EMR-Cluster oder Instance-Gruppen. AWS::Lambda::Function Verwenden Sie die <code>VpcConfig</code> -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

Änderungen	Veröffentlichungsdatum	Description	API-Version
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::Fleet Ressourcen AWS::GameLift::Alias AWS::GameLift::Build , und, um Multiplayer-Spieleserver in bereitzustellen AWS.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Ressourcen	26. Februar 2016	CloudFormation hat die folgenden Ressourcen hinzugefügt: AWS::ECR::Repository Erstellen Sie Amazon Elastic Container Registry-Repositories, 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 eine Internetverbindung 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierte Ressourcen	26. Februar 2016	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::CloudTrail::Trail Verwenden Sie die <code>IsMultiRegionTrail</code> 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 <code>IncludeGlobalResourceTypes</code> -Eigenschaft, um alle globalen Ressourcentypen aufzuzeichnen. AWS::RDS::DBCluster Verwenden Sie die Eigenschaften <code>KmsKeyId</code> und <code>StorageEncrypted</code> zum Verschlüsseln von Datenbank-Instances im Cluster.	15.05.2010
Beibehalten von Ressourcen	26. Februar 2016	Verwenden Sie für Stacks im <code>DELETE_FAILED</code> -Status den <code>RetainResources</code> -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

Änderungen	Veröffentlichungsdatum	Description	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 Beispielvorgänge für die Region Asien-Pazifik (Seoul) verfügbar.	7. Januar 2016	Die folgende Sammlung von CloudFormation Beispielvorgängen bezieht sich auf die Region ap-northeast-2: • Lösungsbeispiele • Anwendungs-Frameworks • Dienstleistungen Weitere Informationen finden Sie unter Arbeiten mit Vorlagen. CloudFormation	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Ressourcenaktualisierungen	28. Dezember 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::CloudFront::Distribution Verwenden Sie für die Distributionskonfiguration die <code>WebACLId</code> 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 <code>ReplicationConfiguration</code> -Eigenschaft, um anzugeben, welche Objekte repliziert werden und wo sie gespeichert werden. Verwenden Sie die Eigenschaften in der <code>NotificationConfiguration</code> -Eigenschaft, um Filter anzugeben, damit Amazon Simple Storage Service Benachrichtigungen für von Ihnen angegebene Objekte sendet.	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
Gruppieren und Sortieren von Parametern	3. Dezember 2015	Verwenden Sie den AWS::CloudFormation::Interf ace Metadatenschlüssel, 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

Änderungen	Veröffentlichungsdatum	Description	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 Amazon EC2 Systems Manager, um ein Dokument zu erstellen, das On-Instance-Konfigurationen angibt.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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

Änderungen	Veröffentlichungsdatum	Description	API-Version
		Amazon DynamoDB (DynamoDB)-Tabelle gespeichert sind. AWS::EC2::Instance Verwenden Sie die <code>SsmAssociations</code> -Eigenschaft, um ein Amazon EC2 Systems Manager-Dokument mit einer Instance zu verknüpfen. AWS::EC2::SpotFleet Verwenden Sie die <code>AllocationStrategy</code> -Eigenschaft, um festzulegen, wie Zielkapazität über Spot-Pools hinweg zugeordnet werden soll. Verwenden Sie die <code>ExcessCapacityTerminationPolicy</code> -Eigenschaft, um anzugeben, wie Instances beendet werden, wenn die Zielkapazität unter der Größe der Spot-Flotte liegt. AWS::Redshift::Cluster Verwenden Sie die <code>KmsKeyId</code>-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::Volume Ressource die <code>AutoEnableIO</code> Eigenschaft, um den I/O Betrieb automatisch wieder aufzunehmen, wenn die Daten eines Volumes inkonsistent werden.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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 auf EC2- oder lokale Instances zu erstellen und anzuwenden. AWS::DirectoryService::SimpleAD Verwenden Sie die Simple Active Directory-Ressource, um ein 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

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		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.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
Ressourcenaktualisierungen	1. Oktober 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::ElastiCache::ReplicationGroup Verwenden Sie die intrinsische <code>Fn::GetAtt</code> - Funktion, um eine Liste der schreibgeschützten Replikatadressen und -Ports abzurufen. AWS::OpsWorks::Stack Verwenden Sie die <code>AgentVersion</code> Eigenschaft, um einen bestimmten OpsWorks Agenten anzugeben. AWS::OpsWorks::App Verwenden Sie die <code>Environment</code> Eigenschaft, um Umgebungsvariablen für eine OpsWorks App anzugeben. AWS::S3::Bucket Für die NotificationConfiguration Eigenschaft 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
CloudFormation Designer	1. Oktober 2015	Verwenden Sie CloudFormation Designer, um Vorlagen über eine drag-and-drop Benutzeroberfläche zu erstellen und zu ändern.	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
Neue Ressource	24. August 2015	Verwenden Sie die AWS::EC2::VPCEndpoint Ressource , um eine private Verbindung zwischen Ihrer VPC und einem anderen AWS Service herzustellen.	15.05.201 0

Änderungen	Veröffentlichungsdatum	Description	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

Änderungen	Veröffentlichungsdatum	Description	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 <code>https://s3.amazonaws.com/templates/myTemplate.template?versionId=123ab1cdeKd0W5IH4GAcYbEngcpTJTDW</code> 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, müssen Sie jetzt auch die Berechtigung haben, die <code>ec2:DescribeAccountAttributes</code> -Aktion aufzurufen.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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 Ressourcen, AWS::ECS::Service, AWS::ECS::Cluster und AWS::ECS::TaskDefinition, um Docker-Container auf einem Cluster von EC2-Instances 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

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		Database Service (Amazon RDS)-Datenbanken zu unterstützen.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
Ressourcenaktualisierungen	11. Juni 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::EC2::Subnet Verwenden Sie die <code>MapPublicIpOnLaunch</code> -Eigenschaft, um automatisch öffentliche IP-Adressen zu Instances in einem Subnetz zuzuweisen. AWS::ElastiCache::CacheCluster Verwenden Sie die <code>SnapshotName</code> -Eigenschaft für die Wiederherstellung von Snapshot-Daten in einem neuen Redis-Cache-Cluster. AWS::IAM::User Verwenden Sie für die <code>LoginProfile</code> -Eigenschaft die <code>PasswordResetRequired</code> -Eigenschaft, damit Benutzer ein neues Passwort festlegen müssen, wenn sie sich bei der AWS-Managementkonsole anmelden. AWS::OpsWorks::Layer Verwenden Sie die <code>LifecycleEventConfiguration</code> Eigenschaft, um Lebenszyklusereignisse für eine OpsWorks Ebene zu konfigurieren. AWS::S3::Bucket Verwenden Sie für die <code>LifecycleConfiguration</code> -Eigenschaft die Eigenschaften <code>NoncurrentVersionExpirationInDays</code> und <code>NoncurrentVersionTransition</code>, um Lebenszyklusregeln für nicht aktuelle Objektversionen anzugeben.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue Parameter typen	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. • <code>AWS::EC2::AvailabilityZone::Name</code> • <code>List<AWS::EC2::AvailabilityZone ::Name></code> • <code>AWS::EC2::Instance::Id</code> • <code>List<AWS::EC2::Instance ::Id></code> • <code>AWS::EC2::Image::Id</code> • <code>List<AWS::EC2::Image::Id ></code> • <code>AWS::EC2::SecurityGroup::GroupName</code> • <code>List<AWS::EC2::SecurityGroup::Gr oupName ></code> • <code>AWS::EC2::Volume::Id</code> • <code>List<AWS::EC2::Volume::Id ></code> • <code>AWS::Route53::HostedZone::Id</code> • <code>List<AWS::Route53::HostedZone ::Id></code>	15.05.2010

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
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.	15.05.201 0

Änderungen	Veröffentlichungsdatum	Description	API-Version
Ressourcenaktualisierungen	16. April 2015	CloudFormation hat die folgenden Ressourcen aktualisiert: AWS::AutoScaling::AutoScalingGroup Verwenden Sie die <code>NotificationConfigurations</code> -Eigenschaft, um mehrere Benachrichtigungen anzugeben. AWS::AutoScaling::LaunchConfiguration Verwenden Sie die <code>PlacementTenancy</code> -Eigenschaft zum Angeben der Tenancy von Instances. Verwenden Sie die <code>ClassicLinkVPCSecurityGroups</code> Eigenschaften <code>ClassicLinkVPCId</code> und, um EC2-Classic-Instances mit einer ClassicLink -fähigen VPC zu verknüpfen. AWS::AutoScaling::ScalingPolicy Verwenden Sie die <code>MinAdjustmentStep</code> -Eigenschaft, um eine Mindestanzahl von Instances festzulegen, die während eines Skalierungsereignisses hinzugefügt oder entfernt werden. AWS::CloudFront::Distribution Für Betrachterzertifikate verwenden Sie die <code>MinimumProtocolVersion</code> -Eigenschaft, um eine Mindestprotokollversion anzugeben. Verwenden Sie für Cache-Verhalten die <code>CachedMethods</code> Eigenschaft, um anzugeben, für welche Methoden Amazon CloudFront (CloudFront) Antworten zwischenspeichert. Verwenden Sie für Ursprünge den, <code>OriginPath</code> um einen Pfad anzugeben, der	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
		zum Anfordern von Inhalten CloudFront verwendet wird. AWS::ElastiCache::CacheCluster Verwenden Sie für Memcached-Cache-Cluster die <code>PreferredAvailabilityZones</code> Eigenschaft, um Knoten in mehreren Availability Zones () AZs anzugeben. AWS::EC2::Volume Verwenden Sie die <code>KmsKeyId</code>-Eigenschaft, um einen vom Kunden verwalteten Schlüssel für verschlüsselte Volumes anzugeben. AWS::OpsWorks::Instance Verwenden Sie die <code>TimeBasedAutoScaling</code> -Eigenschaft für die automatische Skalierung der Instances basierend auf einem von Ihnen festgelegten Zeitplan. AWS::OpsWorks::Layer Verwenden Sie die <code>LoadBasedAutoScaling</code> -Eigenschaft, um lastbasierte Skalierungsrichtlinien anzugeben. Verwenden Sie für Volume-Konfigurationen die <code>Iops</code> Eigenschaften <code>VolumeType</code> und, um jeweils einen Volume-Typ und die Anzahl der I/O Operationen pro Sekunde anzugeben. AWS::RDS::DBInstance Verwenden Sie die <code>CharacterSetName</code> -Eigenschaft, um einen Zeichensatz für unterstützte Datenbank-Engines anzugeben.	

Änderungen	Veröffentlichungsdatum	Description	API-Version
		Verwenden Sie die <code>StorageEncrypted</code> -Eigenschaft, um anzugeben, ob Datenbank-Instances verschlüsselt werden, und <code>KmsKeyId</code>, um einen vom Kunden verwalteten Schlüssel für verschlüsselte Datenbank-Instances anzugeben. AWS::Route53::HealthCheck Verwenden Sie die <code>HealthCheckTags</code> -Eigenschaft, um Tags mit Zustandsprüfungen zu verknüpfen. AWS::Route53::HostedZone Verwenden Sie die <code>VPCs</code>-Eigenschaft zum Erstellen von privaten gehosteten Zonen. Verwenden Sie die <code>HostedZoneTags</code> -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::CustomResource Ressource können Sie die Lambda-Funktion Amazon Resource Names (ARNs) in der <code>ServiceToken</code> 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Elastic Load Balancing update	24. Dezember 2014	Sie können die <code>ConnectionSettings</code> -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 <code>WaitOnResourceSignals</code> 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 UpdatePolicy Attribut .	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

Änderungen	Veröffentlichungsdatum	Description	API-Version
CreationPolicy Attribut	6. November 2014	Mit dem CreationPolicy Attribut können Sie anweisen, CloudFormation zu warten, bis die Anwendungen auf EC2-Instances 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 CreationPolicy Attribut .	15.05.2010
Amazon hat Werte CloudFront weitergeleitet	29. September 2014	Für Cache-Verhaltensweisen können Sie Forward-Header an den Ursprung weiterleiten. Siehe Forwarded Values .	15.05.2010
AWS OpsWorks update	29. September 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. September 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. September 2014	Sie können jetzt Amazon SNS-Themenrichtlinien aktualisieren. Weitere Informationen finden Sie unter AWS::SNS::TopicPolicy .	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierung der RDS DB-Instance	5. September 2014	Sie können angeben, ob eine DB-Instance über das Internet zugänglich ist, indem Sie die Eigenschaft <code>PubliclyAccessible</code> in der Ressource AWS::RDS::DBInstance verwenden.	15.05.2010
UpdatePolicy Aktualisierung von Attributen	05. September 2014	Sie können eine Aktualisierungsrichtlinie für eine Auto-Scaling-Gruppe mit einer verknüpften geplanten Aktion angeben. Weitere Informationen finden Sie unter UpdatePolicy Attribut .	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. Für weitere Informationen siehe AWS::Logs::LogGroup oder AWS::Logs::MetricFilter .	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Aktualisierung der CloudFront Amazon-Vertriebskonfiguration	17. Juni 2014	Sie können zusätzliche Eigenschaften für die CloudFront Distributionskonfiguration 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
Aktualisierung der EC2-Instance	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

Änderungen	Veröffentlichungsdatum	Description	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 herstellt. VPCs Weitere Informationen finden Sie unter AWS::EC2::VPCPeeringConnection .	15.05.2010
Amazon EC2 Auto-Scaling-Gruppenaktualisierung	17. Juni 2014	Sie können eine vorhandene Cluster Placement-Gruppe angeben, in der Instances für eine Amazon EC2 Auto-Scaling-Gruppe gestartet werden. 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

Änderungen	Veröffentlichungsdatum	Description	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
Amazon EC2 Auto Scaling-Unterstützung	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

Änderungen	Veröffentlichungsdatum	Description	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::Stack oder 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

Änderungen	Veröffentlichungsdatum	Description	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. Für weitere Informationen siehe 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-Gruppe basierend auf 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Sekundäre DynamoDB-Indizes	27. Januar 2014	Sie können lokale und globale sekundäre Indizes für DynamoDB-Datenbanken erstellen. Durch die Nutzung von sekundären Indizes 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. Für weitere Informationen siehe 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

Änderungen	Veröffentlichungsdatum	Description	API-Version
AWS Elastic Beanstalk Ebenen der Umgebung	19. Dezember 2013	Sie können angeben, ob AWS Elastic Beanstalk Ressourcen zur Unterstützung eines Webserver oder zur Ausführung von Hintergrundverarbeitungsaufgaben bereitgestellt werden. Weitere Informationen finden Sie unter AWS::ElasticBeanstalk::Environment .	15.05.2010
Ressourcenamen	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

Änderungen	Veröffentlichungsdatum	Description	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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Zuweisen von benutzerdefinierten Ressourcentypen	8. November 2013	In Ihren Vorlagen können Sie Ihren eigenen Ressourcentyp für benutzerdefinierte CloudFormation-Ressourcen angeben (<code>AWS::CloudFormation::CustomResource</code>). Durch die Verwendung Ihres eigenen benutzerdefinierten Ressourcentypnamens können Sie schnell identifizieren, welcher Typ benutzerdefinierter Ressourcen in Ihrem Stack vorhanden ist. Sie können beispielsweise <code>"Type": "Custom:: <i>MyCustomResource</i>"</code> 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 <code>AWS::AccountId</code> 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

Änderungen	Veröffentlichungsdatum	Description	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-Managementkonsole Mithilfe von IAM-Rollen können Sie auch Aufrufe CloudFormation von EC2-Instances aus tätigen, 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 Eigenschaft <code>SourceDBInstanceIdentifier</code> in der Ressource AWS::RDS::DBInstance .	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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Zusätzliche VPC-Unterstützung	17. September 2013	CloudFormation fügt mehrere Verbesserungen zur Unterstützung von VPC- und VPN-Funktionen hinzu - Sie können eine öffentliche IP-Adresse und mehrere private IP-Adressen mit Amazon EC2-Netzwerkschnittstellen verknüpfen. 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

Änderungen	Veröffentlichungsdatum	Description	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
Fortlaufenden Bereitstellungen für Amazon EC2 Auto-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 Attribute. UpdatePolicy	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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-Gruppen	20. Februar 2013	Sie können jetzt EBS-optimierte Instances in Amazon EC2 Auto-Scaling-Gruppen für dedizierten Durchsatz an Amazon Elastic Block Store (Amazon EBS) in automatisch skalierten Instances bereitstellen. Die Implementierung ist ähnlich der zuvor veröffentlichten Unterstützung für optimierte Amazon EBS-EC2-Instances. Weitere Informationen finden Sie unter der neuen EbsOptimized -Eigenschaft in AWS::AutoScaling::LaunchConfiguration.	15.05.2010
Neue Dokumentation	21. Dezember 2012	AWS::EC2::Instance stellt jetzt eine BlockDeviceMappings -Eigenschaft bereit, mit der Sie Blockgerät-Zuweisungen für Ihre EC2-Instance festlegen können. Mit dieser Änderung wurden zwei neue Typen hinzugefügt: • BlockDeviceMapping • Blockgerät	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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-Managementkonsole hinzugefügt. Sie finden diese hier: • Einen Stack von der CloudFormation Konsole aus erstellen • 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 lops-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

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neue und aktualisierte Dokumentation	27. August 2012	Die Themen wurden neu organisiert, um spezifische Informationen zur Verwendung der AWS-Managementkonsole 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: Stapeloptionen 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 CloudFormation Vorlagen • Bootstrapping CloudFormation von Windows-Stacks Neues Thema: Verwendung regulärer Ausdrücke in Vorlagen. CloudFormation	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	API-Version
Neues Feature	25. April 2012	CloudFormation bietet jetzt volle Unterstützung für Virtual Private Cloud (VPC) -Sicherheit mit 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

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
		• 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 Ressource n hinzugefügt und entfernt werden, wenn der Stack aktualisiert wird.	15.05.201 0

Änderungen	Veröffentlichungsdatum	Description	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: • Eine EC2 Dedicated Instance in einer vorhandenen Amazon VPC starten Weitere Informationen finden Sie unter AWS::EC2::Instance. • Legen Sie das <code>SourceDestCheck</code> -Attribut einer EC2-Instance fest, die sich in 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 <code>VPCZoneIdentifier</code> -Eigenschaft Ihrer <code>AWS::AutoScaling::AutoScalingGroup</code> -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

Änderungen	Veröffentlichungsdatum	Description	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 Inhaltsverzeichnis. Die Schätzung der Kosten Ihres CloudFormation Stacks wurde in den Abschnitt Erste Schritte verschoben.	15.05.2010

Änderungen	Veröffentlichungsdatum	Description	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. • Deployment Applications on Amazon EC2 with AWS CloudFormation beschreibt, wie Sie CloudFormation Hilfsskripte verwenden, um Anwendungen mithilfe der in Ihrer Vorlage gespeicherten Metadaten bereitzustellen. • 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 <code>list-stacks</code> 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 Beschreiben und Auflisten Ihrer Stapel .	15.05.2010
Neue Features	26. Mai 2011	Mit den Befehlen <code>describe-stack-resources</code> und <code>get-template</code> können Sie jetzt Informationen aus gelöschten Stacks während 90 Tagen nach der Löschung abrufen. Weitere Informationen finden Sie unter Auflisten von Ressourcen und Abrufen einer Vorlage .	15.05.2010
Neuer Link	1. März 2011	CloudFormation Endpunkthinformationen 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

Änderungen	Veröffent lichungsda tum	Description	API- Versi on
Erstversion	25. Februar 2011	Die erste öffentliche Veröffentlichung von CloudForm ation.	15.05.201 0

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.